POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Bayesian spatio-temporal modeling of air pollution data in the Po valley via INLA method

Laurea Magistrale in Mathematical Engineering - Ingegneria Matematica

Author: Alessandro Dalla Bona

Advisor: Prof. Alessandra Guglielmi

Co-advisor: Matteo Gianella

Academic year: 2022-2023

1. Introduction

In recent years, air pollution caused environmental crises and health issues for individuals living in critical regions, causing an increase in concern worldwide. This master thesis fits into this context as we have analyzed air pollution data collected across the Po Valley in northern Italy, a highly polluted area due to its geography and the numerous sources of pollution. In particular, the observations have been gathered by the Regional Agencies for Environmental Protection (ARPA). The focus has been on the particulate matter with a diameter smaller than 10 μm, commonly known as PM10. To analyze the available data, we opted for a Bayesian approach using the *INLA* method and software (details in [6]) instead of the traditional MCMC technique.

Specifically, we used the *R-INLA* package to analyze two spatio-temporal datasets of daily PM10 observations in 2018 from Emilia-Romagna and Lombardia. Both datasets include information on monitoring stations, but the Lombardia dataset also comprises meteorological data from nearby sites. Ph.D. student Michela Frigeri merged the air pollution observations and the meteorological data to create an integrated dataset. Our models examine the influence of weather variables on PM10 concentrations and attempt to predict the concentrations' future evolution. *INLA* model showed faster run times than standard MCMC techniques when applied to large datasets. Nonetheless, we needed a powerful virtual machine provided by Politecnico di Milano through the computational infrastructure DataCLoud to overcome memory issues. In addition to the multicore virtual machine, we exploited the PARDISO library (see [9]) to effectively parallelize the computation and further reduce run time.

2. Modeling Spatio-Temporal Point-Referenced Processes

In this chapter, we present the theory necessary to analyze spatio-temporal data. Although [2] distinguishes three types of spatial data, we will only focus on the *point-referenced* ones, defined as follows: $Y(\mathbf{s})$ is a random vector at a location $\mathbf{s} \in \mathbb{R}^r$, where $\mathbf{s}$ varies continuously over the domain D, a fixed subset of $\mathbb{R}^r$.

2.1. Stochastic Processes

Point-referenced data are assumed to be realizations of a stochastic process. [2] defines it as an uncountably infinite collection of random variables defined on a continuous domain that can be space, time, or both. To characterize processes, we define the mean and the covariance function. In particular, stationary processes present a constant mean $\boldsymbol{\mu}$ and a covariance function defined by $Cov(Y(\mathbf{s}), Y(\mathbf{s}+\mathbf{h})) = C(\mathbf{h})$ for any $\mathbf{h} \in \mathbb{R}^r$. In this way, the correlation between two points in the domain is only a function of the distance vector. We say that a process is *intrinsically stationary* if the following holds:

$$Var(Y(\mathbf{s}+\mathbf{h}) - Y(\mathbf{s})) = 2\gamma(\mathbf{h}) \quad \forall \mathbf{s} \in D \quad \forall \mathbf{h} \in D. \tag{1}$$

In (1), we introduce an important function in the field of geostatistics, called *variogram* and usually identified as $\gamma(\mathbf{h})$. Moreover, if the variogram (or the covariance function) depends only on the length of the distance, we say that a stochastic process is *isotropic*. In other words, if

$$C(\mathbf{h}) \equiv C(\|\mathbf{h}\|) \equiv C(d). \tag{2}$$

In our work, we assumed *homogeneous* processes, i.e. isotropic and intrinsically stationary, which is a common choice in the literature (see, for example, [3]).

2.2. Gaussian Processes

The most popular homogeneous process is the Gaussian Process, henceforth denoted by GP. GPs are fully identified by their mean, $\mu(\mathbf{s})$, and valid covariance functions, $C(\|\mathbf{h}\|) = Cov(Y(\mathbf{s}), Y(\mathbf{s}+\mathbf{h}))$. The formal definition of a Gaussian process, as in Section 2.7 of [7], is the following: let $\{Y(\mathbf{s}), \mathbf{s} \in D\}$ be a stochastic process, with mean $\mu(\mathbf{s})$ and covariance function $C(\mathbf{h})$. If for any n and any finite collection of n locations $\{\mathbf{s}_1, ..., \mathbf{s}_n\} \subset D$, the n-variate random variable $\mathbf{Y} = (Y(\mathbf{s_1}), ..., Y(\mathbf{s_n}))$ is normally distributed, with mean μ and covariance matrix Σ, then the process is Gaussian. Since the GP assumption is usually made for the error process or spatial residuals in a complex latent Gaussian model, the mean is commonly assumed as $\mu(\mathbf{s}) = 0$ for all $\mathbf{s} \in D$. From now on, we will use the notation $\{w(\mathbf{s}), \mathbf{s} \in D\}$ to denote the GP as a zero-mean spatial random field, as it is commonly represented in literature. There are many possible choices for the Covariance function $C(h)$, but the most common one is to assume a *Matérn-like* structure, which is widely used for its versatility. Henceforth, we will keep the notation of [3], that expresses the covariance function as follows:

$$\begin{aligned} Cov(w(\mathbf{s}_i), w(\mathbf{s}_j)) &= \sigma_w^2 C(h), \\ h &= \|\mathbf{s}_i - \mathbf{s}_j\| > 0, \\ C(h) &= \frac{1}{\Gamma(\nu)2^{\nu-1}}(kh)^\nu K_\nu(kh), \\ \nu &> 0,\ k > 0, \end{aligned} \tag{3}$$

where $K_\nu()$ denotes the modified Bessel function of second kind of order $\nu > 0$ and $\Gamma()$ is the Gamma function. Parameter $\nu > 0$ encodes the degree of smoothness of the process, and it is usually kept fixed, while parameter $k > 0$ is a scaling parameter. This quantity is empirically related to the *range* of the process, which identifies the distance beyond which two points are uncorrelated. The effective range is computed as

$$\rho = \frac{\sqrt{8\nu}}{k}, \tag{4}$$

with ρ corresponding to the distance where the correlation is approximately equal to 0.1. We discussed purely spatial GPs so far, but we still need to extend the discussion to the spatio-temporal case. A spatio-temporal process is denoted as $Y(\mathbf{s}, t) \equiv \{y(\mathbf{s}, t),\ (\mathbf{s}, t) \in D \subseteq \mathbb{R}^2 \times \mathbb{R}^+\}$. We will assume stationary and isotropic processes, whose definitions are immediate extensions of those given above. Most importantly, we will further simplify the model, assuming *separability*. As defined is [7], an isotropic process $Y(\mathbf{s}, t)$ is said to be *separable* if

$$C(\|\mathbf{h}\|, l) = C_s(\|\mathbf{h}\|)C_t(l),$$

with two independent covariance functions for the spatial and temporal parts.

2.3. Bayesian Hierarchical Modeling

In this section, we briefly present key ideas of hierarchical Bayesian modeling for spatio-temporal data. We specify the conditional law $f(\mathbf{y}|\boldsymbol{\theta})$ for the observed data $\mathbf{y} = (y_1, ..., y_d)$ given a vector of unknown parameters $\boldsymbol{\theta} = (\theta_1, ..., \theta_k)$, and a *prior* distribution $\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})$ for $\boldsymbol{\theta}$. The main goal is to make inference on the parameters $\boldsymbol{\theta}$, as well as to assess the suitability of *likelihood* $f(\cdot|\cdot)$. The inference on parameters $\boldsymbol{\theta}$ is based on the posterior distribution, computed with the Bayes Theorem. The posterior is an update of the prior distribution having observed the data, modeled through the likelihood. Since, in many cases, the required integrations for computing posterior distributions are not analytically tractable, numerical methods are necessary, such as Markov Chain Monte Carlo (MCMC) methods, which are the most common. These techniques allow the simulation of samples from the true posterior distribution, which is the limiting distribution of a purposely built Markov Chain (see [2]). In this work, on the contrary, we use another method called Integrated Nested Laplace Approximation (INLA). The core idea of this technique is to avoid building Markov Chains, approximating the posterior distributions with an alternative numerical method that is supposed to be much faster while at the same time granting accuracy and reliability in the estimation process. We will discuss the details of the algorithm in Section 3.

Following the notation in [3], if we denote by $y(\mathbf{s}_i, t)$, the observation of the response variable (PM10 concentrations) at time t and at measurement station $\mathbf{s}_i$, we can now define the model as:

$$y(\mathbf{s}_i,t) = z(\mathbf{s}_i,t)\boldsymbol{\beta} + \xi(\mathbf{s}_i,t) + \varepsilon(\mathbf{s}_i,t), \qquad i = 1,...,d \quad t = 1,...,T,$$

We denote with $z(\mathbf{s}_i,t) = (z_1(\mathbf{s}_i,t),...,z_p(\mathbf{s}_i,t))'$ the vector of p covariates $\boldsymbol{\beta}$ as the related unknown coefficient vector. On the other hand, $\varepsilon(\mathbf{s}_i,t) \overset{iid}{\sim} \mathcal{N}(0,\sigma_\varepsilon^2)$ is the measurement error, defined by a zero-mean Gaussian distribution (white noise) with unknown variance σ_ε^2.

Finally, $\xi(\mathbf{s}_i,t)$ is the realization of the spatio-temporal Gaussian process. This latter component is assumed to be separable, as defined in Section 2.2. Therefore, we choose the covariance of the spatial and temporal parts separately. We assume a GP with a Matérn covariance structure for the spatial part and an *autoregressive* process for the temporal correlation. In [2], we can find more details about these processes, but to briefly present them, an autoregressive process of order p (AR(p)) is a stochastic process in which the value of the current observation at time t is modeled as a linear combination of the p previous observations, plus an error term. The structure of ξ, in the case of an AR(1), can be then written as

$$\xi(\mathbf{s}_i,t) = a\xi(\mathbf{s}_i,t-1) + \omega(\mathbf{s}_i,t), \quad t = 2,...,T$$
$$\xi(\mathbf{s}_i,1) \overset{iid}{\sim} \mathcal{N}\left(0, \frac{\sigma_\omega^2}{1-a^2}\right),$$

where $|a| < 1$; $\omega(\cdot,\cdot)$ is assumed to be a zero-mean Gaussian field, with a Matérn covariance structure for the spatial correlation (as defined in (3)).

If we collect all the observations at time t, $\mathbf{y}_t = (y(\mathbf{s}_1,t),...,y(\mathbf{s}_d,t))'$, we can write model in vectorial form:

$$\begin{aligned} \mathbf{y}_t &= \mathbf{z}_t\boldsymbol{\beta} + \boldsymbol{\xi}_t + \boldsymbol{\varepsilon}_t, & \boldsymbol{\varepsilon}_t &\overset{iid}{\sim} \mathcal{N}(\mathbf{0}, \sigma_\varepsilon^2 I_d) \\ \boldsymbol{\xi}_t &= a\boldsymbol{\xi}_{t-1} + \boldsymbol{\omega}_t, & \boldsymbol{\omega}_t &\sim \mathcal{N}(\mathbf{0}, \Sigma = \sigma_\omega^2\tilde{\Sigma}), \\ & t = 1,...,T. \end{aligned}$$

where $\mathbf{z}_t$ is the $d \times p$ matrix of the covariates, $\boldsymbol{\xi}_t$ is the zero-mean spatio-temporal Gaussian field with $\boldsymbol{\xi}_1$ distributed as a stationary AR(1) process $\mathcal{N}(\mathbf{0}, \Sigma/(1-a^2))$. Finally, $\tilde{\Sigma}$ is the $d \times d$ correlation matrix with elements $\tilde{\Sigma_{ij}} = C(\|\mathbf{s}_i - \mathbf{s}_j\|)$, given by the function (3), with parameters k and ν.

2.4. From modeling to computation

At this point, we need to move from the theoretical framework into the implemented algorithm, which allows us to compute the marginal posterior distributions of the model's parameters. A critical problem is represented by the spatio-temporal Gaussian field, which must be estimated for any station in the regions. The idea behind the algorithm implemented in INLA is to find an approximation of the true unknown GP that represents the spatial residual, changing daily with autoregressive dynamics. For this purpose, they developed the so-called Stochastic Partial Differential Equations approach (SPDE), which allows us to approximate the underlying GP with a Gaussian Markov Random Field (GMRF). All the details of the GMRF can be found in [5], but for our purpose, it is enough to know that they are finite-dimensional Gaussian Processes possessing Markov-like properties. In particular, the field's value at a certain point depends only on values at some of its neighbors, leading to a sparse precision matrix.

To provide insights into the SPDE approach, we can say that the underlying infinite-dimensional

GP, the spatial residual, is approximated at a finite collection of points that covers the domain (mesh) by finding the weak solution of a particular SPDE. In fact, if we denote with $\mathcal{W}(\cdot)$ a Gaussian white noise process on D, SPDEs of the form

$$\mathcal{L}u(\cdot) = \mathcal{W}(\cdot) \tag{5}$$

can be used to define random fields $u(\cdot)$ where the choice of operator $\mathcal{L}$ determines the structure of the covariance matrix of $u(\cdot)$. In particular, it has been proven that the choice of

$$\tau(k^2 - \Delta)^{\alpha/2} u = \mathcal{W},$$

in (5), where Δ is the Laplacian operator, the stationary solution $u(\cdot)$ has a Matérn covariance function of the form (3), with

$$\begin{aligned} \nu &= \alpha - d/2 \\ \sigma_\omega^2 &= \frac{\Gamma(\nu)}{\Gamma(\alpha)(4\pi)^{d/2} k^{2\nu} \tau^2}. \end{aligned}$$

As a result, we will have a GMRF that represents our infinite-dimensional GP. Therefore, we can make posterior inference on the model's parameters, in particular, k and σ_ω, since we can analytically find the precision matrix Q of the GMRF. Finally, we briefly extend this approach to a spatio-temporal GF, $\{u(\mathbf{s}, t),\ \mathbf{s} \in D, t \in \mathbb{R}\}$. In this case, depending on the structure of the temporal correlation, the operator $\mathcal{L}$ is different. For example, as specified in [4], for a Matérn field with AR(1) temporal dynamics, the SPDE becomes:

$$\left(a + \frac{\partial}{\partial t}\right)\left(k^2 - \Delta_s\right)^{\alpha/2} u = \mathcal{W},$$

where Δ_s is the Laplacian operator concerning the spatial coordinates.

2.5. Prior Specifications

The Bayesian approach uses prior distributions of the parameters to incorporate external knowledge or expertise into the model. In our case, we needed to define the prior distributions for the parameters in our regression model. The *R-INLA* package provides various options for priors, but we opted to use the default settings for most parameters. A particular prior that we selected from the available options has been the Penalized Complexity Prior, commonly used in the *INLA* framework. Further details of this method for prior definition are given in [8]. The particular characteristic of these prior is that they can be defined through probability statements. Specifically, the software allows us to define the prior of a particular parameter through two numbers: a threshold for the value and the probability of the parameter being bigger or smaller than such value.

3. Integrated Nested Laplace Approximation (INLA)

In this section, we briefly present the algorithm implemented in the *R-INLA* package that computes marginal posterior distributions of the model parameters. We refer to [6] for the algorithm's details. The basic idea behind *INLA* is to approximate the posterior distribution of the model parameters using a Laplace approximation. This involves approximating the posterior distribution with a Gaussian distribution centered around the mode of the posterior distribution itself. This approximation is then used to calculate marginal likelihoods and other quantities of interest. In particular, the program works by integrating out the latent variables in the model using a series of nested approximations. The first step is to approximate the posterior distribution of the latent variables using a Gaussian distribution. This is done by applying a Laplace approximation (or a simplified version) to the joint distribution of the observed and latent variables. Next, the posterior distribution of the model parameters are approximated by integrating out the latent variables using a numerical integration technique, which relies on finding the "best" values of the parameters $\boldsymbol{\theta}$ by exploring them.

We point out that during this thesis project work and data analysis, we started using the plain version of the package `inla`. We later moved on to the specific version created for spatio-temporal datasets, called `inlabru`. This version (see [1]) is nothing but a wrapper of the original package `inla`. However, `inlabru` simplifies the analysis by providing functions specifically defined for spatio-temporal data.

4. Case study: PM10 in the Po valley

We are now ready to present the posterior analysis of spatio-temporal data with the *INLA* software. As we already mentioned, the data were collected separately in Emilia-Romagna (49 stations) and Lombardia (64 stations); in each case, we built regression models for the response variable, the PM10 concentration. We considered daily PM10 observations from the year 2018, along with associated information for each monitoring station. We included only the variables **Altitude** and **Traffic** in our regression models. The former denotes the altitude in meters above sea level; the latter is a categorical variable that takes a value of 1 if the station is located in a trafficked area. In Lombardia, we had access to meteorological data from a separate network of stations, distinct from the air pollution monitoring stations. To combine the two datasets, we had to match each PM10 station with the nearest meteorological station. Additionally, the meteorological data were collected at 10-minute intervals, so we calculated daily averages to align the temporal scale of both datasets. For this region, we also included this information in the regression models. Figure 1 shows trajectories of daily PM10 in 2018 for all the stations.

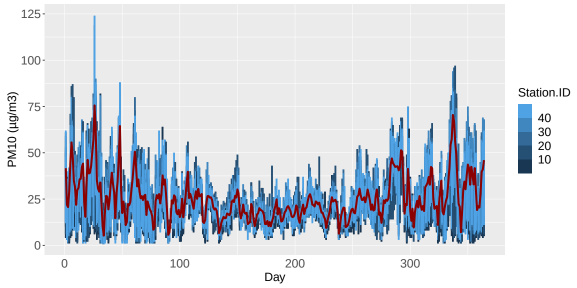


Figure 1: PM10 values in 2018 in Emilia-Romagna. Each station is depicted in a different shade of blue, and the superimposed dark-red line represents the daily average.

The common method to build the model and analyze the data can be summarized in the following steps:

- from the measurement stations build the triangulation of the domain (mesh). In this step, we can choose the spatial resolution and the coordinate system (UTM in our case);
- define all the components using the notation of *INLA*. In particular, we need to specify the spde model with Matérn-like spatial correlation and autoregressive process for the temporal part;
- fit the response variable using the function `bru`, specifying the components and other requests for the algorithm;
- try to predict future trends of PM10 concentrations, using the function `predict`, implemented again by the *R-INLA* package.

The following sections are an overview of some regression models, together with some interesting posterior inference results. We would like to highlight that although we made predictions for future data in all of our models, for clarity we have chosen to present the results in the dedicated Section 4.5.

4.1. Model 1 Emilia-Romagna

Our initial regression model has been constructed to analyze the data gathered from 49 stations situated in the Emilia-Romagna region, considering only the first 90 days of the year 2018. In particular, we analyzed the scaled log-transformed PM10 concentrations, which are assumed to have zero mean Gaussian distributions. In Figure 2 we plotted the triangulation of the domain. At each vertex, the software computes the GMRF approximating the GP.

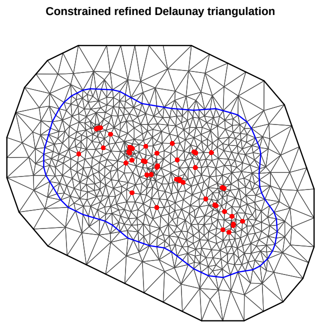


Figure 2: Triangulation of the domain containing Emilia-Romagna, with 856 vertices. In red the starting coordinates of the observation sites, and in blue the *non-convex hull* around the region's border.

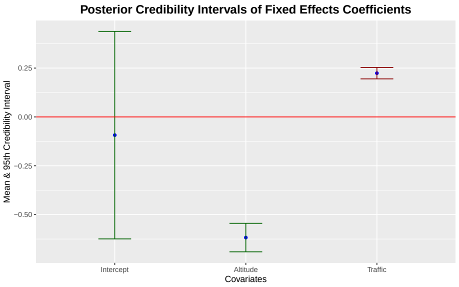


Figure 3: Posterior 95% credibility interval for the regression parameters.

The measurement equation of the model is:

$$y(\mathbf{s}_i,t) = \beta_0 + \beta_1 z_1(\mathbf{s}_i) + \beta_2 z_2(\mathbf{s}_i) + \xi(\mathbf{s}_i,t) + \varepsilon(\mathbf{s}_i,t),$$
$$\varepsilon(\mathbf{s}_i,t) \overset{iid}{\sim} \mathcal{N}(0,\sigma_\varepsilon^2), \qquad i = 1, ..., 49, \quad t = 1, ..., 90,$$

where $z_1(\mathbf{s}_i)$ and $z_2(\mathbf{s}_i)$ are **Altitude** and **Traffic** regression variables, respectively. The spatio-temporal field denoted by $\xi(\cdot)$ is assumed to have a Matérn spatial covariance structure and autoregressive dynamics of order 3 for the temporal part. This process has then five unknown parameters, namely, $(a1, a2, a3)'$ for the AR(3) and $(\sigma_\omega, k)'$ for the Matérn field. However, in the following paragraphs, we will not estimate directly k, but instead the effective range ρ; the two quantities are empirically related by (4).
From the posterior analysis, we can affirm that the regression variables considered are significant. As depicted in Figure 3, the related coefficients have 95% posterior credibility intervals that do not contain zero. In particular, the **Altitude** has a negative impact on the PM10 concentration, while the **Traffic** positively influences the air pollution (as we expected).
Moreover, we estimated an effective range of about 180 km.

4.2. Model 2 Emilia-Romagna

The second model has for response variable a different transformation of PM10 concentrations. In particular, we created a new variable, **Diff**, which stores the sequential difference in log-transformed PM10 concentrations. We computed this variable by taking the log transformation of daily PM10 concentrations, calculating the difference between each day and the previous day for each station, and detrending the variable by subtracting the mean. The differencing was performed to make the time series stationary and remove the offset. The model is the same as the one presented in Section 4.1, although we considered 120 days for posterior distributions estimation. The results obtained reflect the transformation of the dataset. Firstly, the regression variables lost their significance, indicating that the changes in daily PM10 concentrations are independent of the monitoring stations' characteristics. Secondly, all three autoregressive parameters had negative values, as shown in Figure 4. This result highlights the oscillatory behavior of the **Diff** variable, around zero.

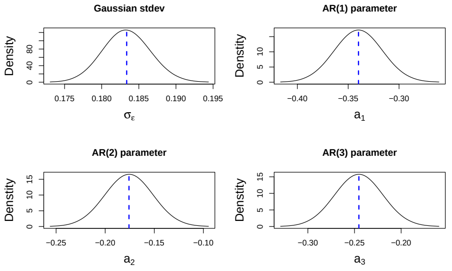


Figure 4: Marginal posterior distributions of parameters σ_ε, a_1, a_2 and a_3.

4.3. Model 1 Lombardia

As we already mentioned, in the case of Lombardia, we integrated the available meteorological information. It should be noted that all of the continuous variables have undergone a process of scaling and centering. After some preliminary considerations, we selected just a few variables to integrate as regressors in our model, as we can see from the model equation (6)

$$\{y(\mathbf{s}_i,t) = \beta_0 + \sum_{k=1}^{2} \beta_k z_k(\mathbf{s_i}) + \tag{6}$$
$$\sum_{l=3}^{6} \beta_l z_l(\mathbf{s_i},t) + \xi(\mathbf{s}_i,t) + \varepsilon(\mathbf{s}_i,t)\},$$
$$\varepsilon(\mathbf{s}_i,t) \overset{iid}{\sim} \mathcal{N}(0,\sigma_\varepsilon^2),$$
$$i = 1, ...,64, \; t = 1, ..., 90.$$

The response variable is the scaled log-transformed PM10; the notation is the same as in Section 4.1. The vector of regressors is composed by:

- constant level, identified by the **Intercept**,
- $z_1(\mathbf{s}_i)$, denoting the **Altitude** of the station,
- $z_2(\mathbf{s}_i)$, denoting whether the station is of type **Traffic** or not,

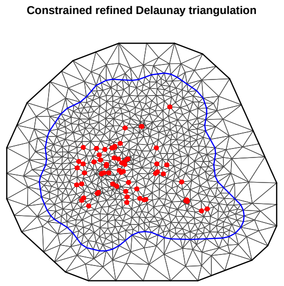


Figure 5: Triangulation of the domain containing Lombardia, with 806 vertices. In red the starting coordinates of the observation sites, and in blue the *non-convex hull* around the border.

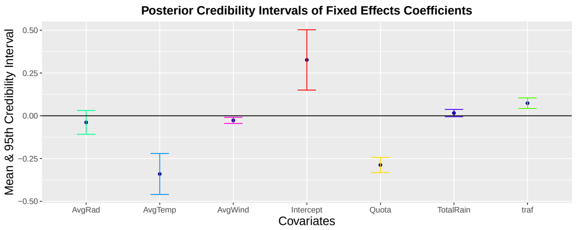


Figure 6: 95% credibility intervals of regression parameters.

- $z_3(\mathbf{s}_i, t)$, denoting the daily **AvgRad**,
- $z_4(\mathbf{s}_i, t)$, denoting the daily **AvgTemp**,
- $z_5(\mathbf{s}_i, t)$, denoting the daily **AvgWind**,
- $z_6(\mathbf{s}_i, t)$, denoting the daily **TotalRain**.

Finally, we point out that in this case, the stations are 64 and that we consider 90 days of observations for the marginal posterior distributions estimation. Figure 5 shows the mesh used to approximate the GMRF.

As we can see in Figure 6, the variables **Altitude** and **Traffic** remain relevant, with a negative and a positive impact on the concentration of pollution, respectively. In addition, as we expected, it emerged that some of the meteorological variables included in the model are relevant. Both the **AvgTemp** and **AvgWind** variables have negative coefficients. Finally, we observe an unusual behavior of the intercept coefficient, which is estimated to be significantly positive, even though the data have been standardized and the logPM10 values have a sample mean equal to 0. To solve this issue we tried to remove the intercept from the model. However, the software automatically included a coefficient for the constant level, leading to even poorer results. We hypothesize that adding more covariates and components resulted in identifiability problems, despite the other posterior estimates appearing reasonable. The other posterior distributions present similar behavior concerning the case of Emilia-Romagna, although we estimated a higher range of about 240 km.

4.4. Model 2 Lombardia

The last presented model is another regression model in which the detrended daily difference in logPM10 is the response variable, as in Section 4.2. This model shares the same regression variables as the one presented in Section 4.3, except **AvgRad**, which was found to be insignificant a posteriori. In addition, we have decided to reduce the spatial resolution by constructing a coarser mesh. This aims at fitting a longer time series, specifically one that spans 180 days.

When analyzing the posterior distributions, it is imminent to notice the difference concerning the model presented in Section 4.3 due to the different transformations of the response variable. In particular, we can see how regression variables that express general characteristics of the stations, namely **Altitude** and **Traffic**, turn out to be irrelevant. Daily changes in the concentration of PM10 do not depend on these variables. On the other hand, our statistical analysis supports the intuitive notion that weather conditions play a role in the fluctuation of air pollution levels. Specifically, the coefficients for **AvgWind** and **TotalRain** are consistently negative. This is not surprising: it is common sense that through rain and wind, the air gets cleaner.

Moreover, we got similar results for the autoregression parameters seen in Section 4.2 and a similar range, estimated to be around 190 km.

4.5. Prediction of future data

As previously stated, another objective of this study was to forecast future changes in PM10 concentrations. Our approach was to predict approximately 10% of the days considered for the marginal posterior distribution estimation.

From the beginning, the outcomes have been unsatisfactory. As illustrated in Figure 7, while the model fits the response variable accurately, it fails in forecasting future data.

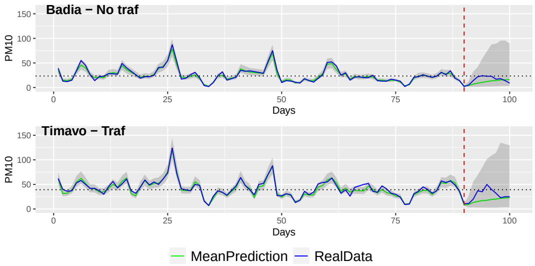


Figure 7: Time series of the true value of PM10 concentration (blue line) compared to the mean over 1000 samples of predicted value (green line). The grey area represents the posterior mean's 95% credibility interval. The red dashed line represents the border between past and future data (the 90th day). The horizontal black dashed line is the overall mean of the pollution observed at the stations.

By including meteorological information in Lombardia, we hoped to improve the forecasting, but it did not happen. Figure 8 displays prediction results for two stations in Lombardia, confirming that the model fits well past data but cannot predict the future evolution of PM10.

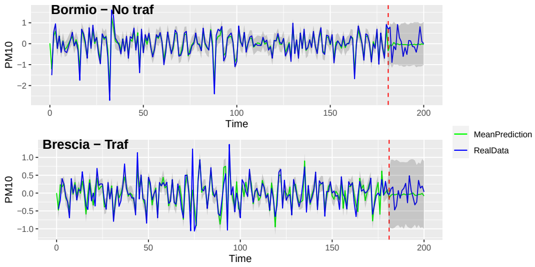


Figure 8: Time series of consecutive differences of logPM10 concentration (blue line) compared to the mean over 1000 samples of predicted value (green line). The grey area represents the posterior mean's 95% credibility interval. The red dashed line represents the border between past and "future" data (the 180th day).

5. Conclusions

Learning and applying a completely new Bayesian inference method has been quite challenging. Specifically, we had difficulties comprehending the step-by-step algorithm used by *INLA* to approximate the posterior distributions. Nonetheless, we have been able to construct and analyze complex models and to get interesting results, particularly regarding the regression aspect. Finally, we could not find better outcomes in future data prediction, even when we introduced meteorological information in the case of Lombardia. However, we want to point out that this deficit of the model in the prediction task does not necessarily mean that the problem lies in the *INLA* software or its underlying method. I suppose that the complex nature of the air pollution phenomenon and my inability to build better models is responsible for the poor results obtained in the future prediction task. In conclusion, our suggestion to other analysts is to use the *INLA* program in constructing Bayesian models for inference, especially if there's a chance to compare the outcomes with those acquired through traditional MCMC methods, to find the best trade-off between computational costs and accuracy.

References

[1] F. E. Bachl, F. Lindgren, D. L. Borchers, and J. B. Illian. inlabru: an R package for Bayesian spatial modelling from ecological survey data. *Methods in Ecology and Evolution*, 10(6):760–766, 2019.

[2] S. Banerjee, B. P. Carlin, and A. E. Gelfand. *Hierarchical Modeling and Analysis for Spatial Data.* CRC Press, 2015.

[3] M. Cameletti, F. Lindgren, D. Simpson, and H. Rue. Spatio-temporal modeling of particulate matter concentration through the spde approach. *AStA Advances in Statistical Analysis*, 97(2):109–131, 2013.

[4] F. Lindgren, D. Bolin, and H. Rue. The spde approach for gaussian and non-gaussian fields: 10 years and still running. *Spatial Statistics*, 50:100599, 2022.

[5] H. Rue and L. Held. *Gaussian Markov Random Fields: Theory and Applications.* Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 2005.

[6] 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.

[7] S. Sahu. *Bayesian modeling of spatio-temporal data with R.* CRC, 2022.

[8] 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.

[9] J. van Niekerk, H. Bakka, H. Rue, and O. Schenk. New frontiers in Bayesian modeling using the inla package in R. (1), 2021.

POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Bayesian spatio-temporal modeling of air pollution data in the Po valley via INLA method

Tesi di Laurea Magistrale in
Mathematical Engineering - Ingegneria Matematica

Author: **Alessandro Dalla Bona**

Student ID: 10600333
Advisor: Prof. Alessandra Guglielmi
Co-advisors: Matteo Gianella
Academic Year: 2022-23

Abstract

The focus of this thesis is on air pollution, specifically regarding the concentration of particulate matter. Data consist of time series for PM10 concentrations collected in 2018 from various monitoring stations located across Lombardia and Emilia-Romagna, two Italian regions in the Po Valley. We employed a Bayesian hierarchical approach to analyze the data and used the *INLA* method and its corresponding R package. Due to the availability of meteorological information only for Lombardia, we carried out separate analyses for the two regions. The Regional Associations for Environmental Protection (ARPA) provided all the data used in this study. We begin by discussing the theoretical background of spatio-temporal analysis, including stochastic processes in geostatistics and Bayesian hierarchical modeling. We then provide an overview of the Stochastic Partial Differential Equations (SPDE) approach, used by the software *INLA* for spatio-temporal models. Finally, we describe the algorithm implemented in *INLA* package for posterior Bayesian inference. Finally, we present the results of our analysis using various additive regression models to explain the PM10 concentration as the response variable. We identified significant regressors and observed that meteorological conditions play a role in air pollution in Lombardia. However, despite our attempts, we have been unable to predict the future evolution of PM10 concentration in monitoring stations.

Key words: air pollution; meteorological data; Bayesian hierarchical inference; spatio-temporal regression models; INLA; SPDE approach.

Abstract in lingua italiana

Il focus di questa tesi riguarda la problematica dell'inquinamento atmosferico, segnatamente la concentrazione di particolato. La ricerca è stata condotta utilizzando un dataset costituito da serie temporali di osservazioni delle concentrazioni di PM10 raccolte nel 2018 in varie stazioni di monitoraggio situate in Lombardia ed Emilia-Romagna, due regioni italiane nella valle del Po. Per analizzare i dati, abbiamo utilizzato un approccio gerarchico bayesiano e il metodo *INLA* con il relativo pacchetto R. Sono state presentate due analisi separate a causa della disponibilità di informazioni meteorologiche solo per la Lombardia. Tutti i dati utilizzati in questo studio sono stati forniti dalle Associazioni Regionali per la Protezione dell'Ambiente (ARPA). Iniziamo discutendo il background teorico dell'analisi spazio-temporale, in particolare i processi stocastici nell'ambito della geostatistica e la modellizzazione gerarchica Bayesiana. Forniamo quindi una panoramica dell'approccio ad Equazioni Differenziali Stocastiche a Derivate Parziali (SPDE), utilizzato nel software *INLA* per i modelli spazio-temporali. Infine, descriviamo l'algoritmo implementato nel pacchetto *INLA* per l'inferenza bayesiana a posteriori. Infine, presentiamo i risultati della nostra analisi utilizzando vari modelli di regressione additivi per spiegare la concentrazione di PM10 come variabile di risposta. Abbiamo identificato regressori significativi e osservato che le variabili meteorologiche hanno un impatto sull'inquinamento atmosferico in Lombardia. ioneTuttavia, non siamo stati in grado di prevedere l'evoluz futura della concentrazione di PM10 nelle stazioni di monitoraggio nonostante i nostri sforzi.

Parole chiave: inquinamento atmosferico; dati meteorologici; inferenza Bayesiana; modelli gerarchici spazio-temporali; INLA; approccio SPDE.

Parole chiave:

Contents

Introduction

Nowadays, the importance of information encoded in all sorts of data is well known. Among the different types of data, geographically referenced and temporally correlated ones play a significant role in acquiring insight into the world around us. To exploit such empirical information, we need to model and analyze the underlying stochastic random processes that can appropriately interpret those data. In this thesis, we study georeferenced time series of air pollution data, collected by a network of monitoring stations managed by Regional Agencies for Environmental Protection (ARPA) across the Po Valley. In particular, we concentrate on a type of Particulate Matter, PM10. We consider the Bayesian statistical approach, which offers point estimates and credibility intervals for parameters and predictions. We investigate a specific Bayesian method and its associated software tool called *INLA*, which uses an alternative technique for Bayesian inference. *INLA* approximates the marginal posterior distributions of model parameters using a combination of numerical integration and Laplace approximations, enabling efficient and accurate inference for complex models that involve latent variables or complex dependence structures. Our exploration of the *INLA* method aims at finding a viable and efficient alternative to standard Markov Chain Monte Carlo (MCMC) techniques. MCMC is widely used in Bayesian inference, but it can be computationally demanding and time-consuming, particularly for complex models or large datasets. Therefore, try to understand whether the *INLA* method can offer a valid alternative, achieving comparable accuracy and precision in inference while being computationally more efficient.

To use this specific package, which works within the R environment, we needed to understand the mathematical theory and the associated techniques for approximating Bayesian posteriors. The numerical approximation methods involved are rather complex from the mathematical point of view, and operating the software was not straightforward or immediate. It took some time for us to produce the necessary code to interface with the *R-INLA* package and perform posterior analysis of our spatio-temporal datasets. In the end, thanks to the patience and availability of the authors, in particular, professors Finn Lindgren and Håvard Rue, we have become more familiar with the philosophy of the software and its coding implementation. Our sincere gratitude goes to professors Lindgren and Rue for their constant help.

Our focus is on developing spatio-temporal models that can fit geo-referenced time series datasets and possibly predict the future evolution of the PM10 concentrations. This goal required an extensive study of the theoretical foundations of these models, including a detailed examination of geostatistics-based spatial processes and their temporal extensions. Additionally, the methods implemented in the package employ an innovative approach to handling spatio-temporal data based on Stochastic Partial Differential Equations. Delving into this field has proven to be a challenging task, requiring considerable effort and commitment.

After becoming proficient enough in the *R-INLA* package, we finally used it to analyze spatio-temporal data, which consist of two distinct datasets, collecting daily PM10 observations in 2018. In addition to the pollutant measurements, both datasets, one for Emilia-Romagna and the other for Lombardia include information on the monitoring stations. The Lombardia dataset also contains meteorological data integrated by Ph.D. student Michela Frigeri. The air pollution monitoring stations and the meteorological sites are located in different places, and this problem has required merging information from both sources to create an integrated dataset. Specifically, any monitoring station in Lombardia has been associated with meteo information from the closest meteorological site in the same region. While this data preprocessing strategy may not be the most accurate, it was the most straightforward approach we could adopt. Despite the limitations of the merged dataset, we were able to build models that could investigate the influence of weather variables on air pollutants.

By applying the *INLA* method, we could gain insight into its strengths and weaknesses. One notable drawback we have encountered is the difficulty in interpreting the numerical nature of the Bayesian inference algorithm. Specifically, it has been challenging to visualize the posterior step-by-step estimation process. Nonetheless, the approximation provided by the *R-INLA* program has been accurate and fast, especially compared to the reputedly time-consuming standard MCMC techniques. However, we have faced significant computational challenges as we have analyzed larger datasets with more complex models. Specifically, the technique requires many matrix operations that need a lot of memory. Large covariance matrices are necessary for accurate spatio-temporal models, which slow down the process. By imposing high levels of spatial resolution and temporal extension, insufficient memory space causes the program to crash. Fortunately, we solved this issue using a computational infrastructure, DataCloud, provided by Politecnico di Milano, which gave us access to a powerful virtual machine. In general, when using *R-INLA*, memory issues can pose significant obstacles for those who do not have access to such hardware.

In the end, the VM allows us to run complex models analyzing big datasets in a fast and efficient way. We found out that the regression models implemented with *R-INLA* have been able to fit the response variable accurately.

Our analysis of the modeled regressors yielded intriguing findings that corroborated expert opinions and existing literature on spatio-temporal models for air pollution data. Notably, we were delighted to uncover evidence supporting the notion that meteorological factors play a role in air pollution levels in Lombardia. However, our efforts to forecast PM10 concentration in both regions have been unsuccessful. We could not build a model that can accurately predict the evolution of PM10 concentration. We had hoped that integrating meteorological information would have led to better outcomes in Lombardia, which ultimately did not happen. However, we attribute our inability to make accurate predictions to the complexity of the task and our limitations in building better models, rather than to any shortcomings in the software.

1 | Modeling Spatio-Temporal Point-Referenced Processes

In this chapter, we focus on the theory behind a proper quantitative analysis of spatio-temporal data, with special attention on concepts needed for INLA software (see Section 1.3).

As far as spatial data are concerned, Section 1.1 of Banerjee et al. (2015) discriminates three types of spatial data, each of them leading to a specific model formulation:

- *point-referenced data*, where $Y(\mathbf{s})$ is a random vector at a location $\mathbf{s} \in \mathbb{R}^r$, where $\mathbf{s}$ varies continuously over the domain D, a fixed subset of $\mathbb{R}^r$.
- *areal data*, where the domain D is partitioned into a finite number of areal units with well-defined boundaries
- *point pattern data*, where D itself is random; $Y(\mathbf{s})$ is equal to 1 for all locations $\mathbf{s} \in \mathbb{R}^r$, indicating the occurence of the event.

Given the nature of our specific problem, we will focus on point-referenced data.

In Section 1.1 we address the modeling of spatial and spatio-temporal data through stochastic processes, while in Section 1.2 we present the Bayesian inference approach for hierarchical models. In Section 1.3, we quickly outline the main ideas written in Lindgren et al. (2011) and Lindgren et al. (2022), to will move from a theoretical framework to its numerical implementation, building a path to the analysis of real data.

1.1. Stochastic Processes

In this section, basic notions in point-referenced models are discussed. A model for such data needs the assumption of an underlying stochastic process, that is, as defined in Chapter 3 of Banerjee et al. (2015), or similarly in Chapter 2 of Sahu (2022), an uncountably infinite collection of random variables defined on a continuous domain, that can be space, time or both. We will present the main features of such a collection of random variables through the characteristics of a spatial stochastic process $\{Y(\mathbf{s}), \mathbf{s} \in D\}$, where in general $D \subseteq \mathbb{R}^r$ ($r = 2$ in most cases, included ours). We assume that the stochastic processes have finite mean $\mu(\mathbf{s})$ and variance $Var(\mathbf{s})$ for all $\mathbf{s}$ in D. We proceed now to define a fundamental property of $\{Y(\mathbf{s}), \mathbf{s} \in D\}$, the stationarity, which can be of different types, depending on the characteristics of the process.

Following the definitions of Banerjee et al. (2015), the process $\{Y(\mathbf{s}), \mathbf{s} \in D\}$ is said to be *strictly stationary* if, for any given $n \geq 1$, any set of n sites $\{\mathbf{s}_1, ..., \mathbf{s}_n\}$ and any $\mathbf{h} \in \mathbb{R}^r$, the joint distribution of $(Y(\mathbf{s}_1), ..., Y(\mathbf{s}_n))$ is the same as that of $(Y(\mathbf{s}_1+\mathbf{h}), ..., Y(\mathbf{s}_n+\mathbf{h}))$. This requirement is not necessarily met in real cases, so we introduce a less restrictive condition, *weak stationarity.* From Banerjee et al. (2015), a spatial process, with covariance given by

$$Cov(Y(\mathbf{s}), Y(\mathbf{s}+\mathbf{h})) = \mathbb{E}[(Y(\mathbf{s}) - \mu)(Y(\mathbf{s}+\mathbf{h}) - \mu)] \tag{1.1}$$

is called *weakly stationary* if

$$\mu(\mathbf{s}) = \mu \tag{1.2}$$
$$Cov(Y(\mathbf{s}), Y(\mathbf{s}+\mathbf{h})) = C(\mathbf{h}) \quad \forall \mathbf{h} \in \mathbb{R}^r. \tag{1.3}$$

The function $C(\mathbf{h})$ is called *covariogram*, or just covariance function. The second request implies that the correlation between random variables at distinct points in the domain depends only on the distance, not on their specific locations. In many real applications, weak stationarity, also known as second-order stationarity, is assumed without any loss of accuracy. In some cases, an even weaker notion may be needed. Indeed, if (1.2) holds, and if

$$Var(Y(\mathbf{s}+\mathbf{h}) - Y(\mathbf{s})) = 2\gamma(\mathbf{h}) \quad \forall \mathbf{s} \in D \quad \forall \mathbf{h} \in D, \tag{1.4}$$

then the process is said to be *intrinsically stationary.* With this property, we introduce an important function, the *variogram*, $2\gamma(\mathbf{h})$ which is extensively used in classical geostatistical analysis to gain insight into the spatial behavior of the random field, usually in the form of the *semivariogram*, $\gamma(\mathbf{h})$.

Finally, we briefly discuss the property of *isotropy.* An *intrinsically stationary* process is called *isotropic* if its *semivariogram* depends only on the length of the separation vector $\mathbf{h}$. Since it can be shown, see for example Banerjee et al. (2015), that the *covariogram* and the *variogram* are strictly related (see equation (1.5) below), we can intuitively interpret isotropy in the perspective of correlation: the correlation between values at different locations depends only on the length of distance, not on the direction of the separation vector $\mathbf{h}$. We point out that the isotropy is a property of the *variogram*, but it can be extended to the *covariogram* $C(\mathbf{h})$. It can be shown that, for intrinsic stationary processes,

$$\gamma(\mathbf{h}) = C(\mathbf{0}) - C(\mathbf{h}) \tag{1.5}$$

(see Section 2.4 of Sahu (2022)). Therefore, in practical modeling, it is enough to specify a valid covariance function $C(\mathbf{h})$, and the *variogram* automatically follows. By valid covariance function, we mean a positive quadratic form. In real scenarios, isotropy may not be fully adequate, but still, it is assumed very often, and we will do the same. The crucial role of isotropy is that it allows us to specify the covariance function as a monodimensional function that depends only on the parameter $\|\mathbf{h}\|$ that we will call from now on, d, as in *distance.*

$$C(\mathbf{h}) \equiv C(\|\mathbf{h}\|) \equiv C(d). \tag{1.6}$$

Moreover, if a process is *intrinsically stationary* and *isotropic*, it is called *homogeneous.*

We now present another quantity related to the covariance, which is strictly related to the spatial interpretation of the stochastic process. Usually, the correlation function, which measures the co-dependence of two points in the field, decreases monotonically with the distance. The minimum value ρ at which $C(\rho) = 0$ is called *range* and is the distance beyond which the field values are uncorrelated. We specify that, usually, the covariance function is strictly positive in the domain $h > 0$, therefore it does not allow a finite value for the *range*. In such cases, we define the *effective* range as the distance beyond which correlation is smaller than a chosen threshold.

In Section 1.1.1, we discuss a very important *homogeneous* stochastic process, which is widely used because of its convenience and versatility, the Gaussian Process.

1.1.1. Gaussian Processes

Gaussian Process are fully identified by their mean, $\mu(\mathbf{s})$, and valid covariance functions, $C(\|\mathbf{h}\|) = Cov(Y(\mathbf{s}), Y(\mathbf{s}+\mathbf{h}))$. The formal definition of a Gaussian process, as in Section 2.7 of Sahu (2022), is the following: let $\{Y(\mathbf{s}), \mathbf{s} \in D\}$ be a stochastic process, with mean $\mu(\mathbf{s})$ and covariance function $C(\mathbf{h})$. If for any n and any finite collection of n locations $\{\mathbf{s}_1, ..., \mathbf{s}_n\} \subset D$, the n-variate random variable $\mathbf{Y} = (Y(\mathbf{s_1}), ..., Y(\mathbf{s_n}))$ is normally distributed, with mean μ and covariance matrix Σ given by:

$$\boldsymbol{\mu} = \begin{pmatrix} \mu(\mathbf{s_1}) \\ \mu(\mathbf{s_2}) \\ \vdots \\ \mu(\mathbf{s_n}) \end{pmatrix}, \Sigma = \begin{pmatrix} C(0) & C(d_{12}) & \cdots & C(d_{1n}) \\ C(d_{21}) & C(0) & \cdots & C(d_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ C(d_{n1}) & C(d_{n2}) & \cdots & C(0) \end{pmatrix}, \text{ where } d_{ij} = \|\mathbf{s}_i - \mathbf{s}_j\|,$$

the process is Gaussian.

As a consequence of the definition, we can immediately express the joint density of $\mathbf{Y}$ for any finite value of n.

At this point, to fully define the structure of a Gaussian Process (GP) and therefore use it in practical modeling, we need to specify the mean and the covariance function. As discussed in Sahu (2022), since the GP assumption is usually made for the error process, or more in general, for spatial residuals in a complex latent Gaussian model, the mean is commonly assumed as $\mu(\mathbf{s}) = 0$ for all $\mathbf{s} \in D$. From now on, we will use the notation $\{w(\mathbf{s}), \mathbf{s} \in D\}$ to denote the GP as a zero-mean spatial random field, as it is commonly represented in literature.

Matérn Covariance Function

Regarding the covariance, one common choice is to assume a *Matérn-like* covariance structure. Henceforth, we will keep the notation of Cameletti et al. (2013), that expresses the function as follows:

$$\begin{aligned} Cov(w(\mathbf{s}_i), w(\mathbf{s}_j)) &= \sigma_w^2 C(h), \qquad h = \|\mathbf{s}_i - \mathbf{s}_j\| > 0, \\ C(h) &= \frac{1}{\Gamma(\nu)2^{\nu-1}}(kh)^\nu K_\nu(kh), \qquad \nu > 0,\ k > 0, \end{aligned} \tag{1.7}$$

where $K_\nu()$ denotes the modified Bessel function of second kind of order $\nu > 0$ and $\Gamma()$ is the Gamma function. This form comes from the assumption that in spatial (and spatio-temporal) modeling, the variance is equal for all locations, namely

$$Var(w(\mathbf{s})) = \sigma_w^2, \qquad \forall \mathbf{s} \in D \tag{1.8}$$

and that two points distant h have correlation function $C(h)$.

As stated in Cameletti et al. (2013), parameter $\nu > 0$ is responsible for the level of smoothness of the process, and it is commonly kept constant. The choice of smoothness parameter ν, as discussed in Section 2.6 of Sahu (2022), gives rise to particular cases of the Matérn family of covariance functions, for example:

- if $\nu = 0.5$, the function reduces to an exponential correlation, decaying with the distance as e^{-kh};
- if $\nu \to \infty$, the covariance function is called Gaussian since the correlation decays with the distance as $e^{-k^2h^2}$.

On the other hand, parameter $k > 0$ in the *Matérn* case acts as a scaling factor and it is linked to the *effective range* through the empirical relation

$$\rho = \frac{\sqrt{8\nu}}{k}, \tag{1.9}$$

with ρ corresponding to the distance where the correlation is approximately equal to 0.1. (see Cameletti et al. (2013) and Section 2 of Lindgren et al. (2011) for more details).

At this point, we have introduced one of the building blocks of our spatio-temporal model, the (purely spatial) Gaussian Process, with zero mean and a Matérn-like covariance, widely used in the literature for its versatility. This particular structure may seem restrictive and peculiar, but its massive diffusion in the literature shows the contrary. The assumption of Gaussianity allows the use of well-known formulas and distributional properties, which ease computational tasks. Moreover, through transformations and mixtures of GPs, we can build a wide range of models.

1.1.2. Spatio-temporal processes

At this point, to examine geostatistical data that is observed at multiple time points, it is necessary to expand our analysis to include spatio-temporal processes. The concept of time can be approached using two alternative methods: viewing it as a *continuous* variable or as a *discrete* variable. As discussed in Chapter 11 of Banerjee et al. (2015), under the assumption of continuity, namely $t \in \mathbb{R}^+$, there is an ideal measurement of the response variable at each moment. In the case of discrete time, the observation will be some kind of average over the specified period (hourly, daily, monthly, etc). In our discussion, the time variable, for convenience, is assumed to be discrete, and in the case study that we will present in Chapter 3, it will be an indicator of the day of the observation. Nonetheless, with an abuse of notation, we will still formally identify the time domain as $t \in \mathbb{R}^+$, as it is commonly done in literature (see, for example, Cameletti et al. (2013).

We can now introduce the formal extension of a random process to spatio-temporal processes. As in Banerjee et al. (2015), a spatio-temporal process is denoted as $Y(\mathbf{s}, t) \equiv$

$\{y(\mathbf{s},t),\ (\mathbf{s},t) \in D \subseteq \mathbb{R}^2 \times \mathbb{R}^+\}$, with a spatio-temporal covariance function of the form

$$Cov(y(\mathbf{s},t), y(\mathbf{s}',t')) = \sigma^2 C((\mathbf{s},t),(\mathbf{s}',t')), \quad (\mathbf{s},t),(\mathbf{s}',t') \in \mathbb{R}^2 \times \mathbb{R}^+. \tag{1.10}$$

The properties we presented in Section 1.1 can be easily extended to the spatio-temporal case. In particular, if the mean is constant and the covariance depends on the locations, both in space and in time, only through the distance vector $\mathbf{h} = (\mathbf{s} - \mathbf{s}')$ and the temporal lag $l = (t - t')$, the process is *weakly stationary*. Moreover, the process is *isotropic* if, according to the more convenient parametrization,

$$C((\mathbf{s},t),(\mathbf{s}',t')) = C(\mathbf{h},l) = C(\|\mathbf{h}\|,l), \quad \forall \mathbf{h} \in \mathbb{R}^2, \quad \forall l \in \mathbb{R}^+. \tag{1.11}$$

We can further simplify the spatio-temporal nature of the process by imposing separability. More specifically, as in Section 2.8 of Sahu (2022), an isotropic process $Y(\mathbf{s},t)$ is said to be *separable* if

$$C(\|\mathbf{h}\|,l) = C_s(\|\mathbf{h}\|)C_t(l). \tag{1.12}$$

In Section 1.2.1, we will specify a particular choice for the covariance structure that is very popular in the literature and that we also assumed in the real-case analysis performed in Chapter 3.

Now, with the necessary theoretical background, we can proceed to present the Bayesian approach to make inference on the parameters of the model, starting with the observation of spatio-temporal data. In particular, we model the supposed underlying process that generates a collection of measured data $Y(\mathbf{s}_i,t)$, with $i \in \{1,...,d\}$, with d number of collection stations in the spatial domain, and $t \in (0,...T)$. We aim at inference on the model unknowns, such as regression coefficients and covariance parameters.

1.2. Bayesian Hierarchical Modeling

In this section, we briefly revise the notions presented in Chapter 5 of Banerjee et al. (2015), discussing the Bayesian inference approach. As we will see later, the concepts of Section 1.1 will be integrated as a part of a hierarchical model to make inference on the parameters.

The key feature of a Bayesian approach is its ability to combine complex data models with external expertise by modeling observed data and any unknown parameter as a random variable. We specify the conditional law $f(\mathbf{y}|\boldsymbol{\theta})$ for the observed data $\mathbf{y} = (y_1, ..., y_d)$ given a vector of unknown parameters $\boldsymbol{\theta} = (\theta_1, ..., \theta_k)$, and a *prior* distribution $\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})$ for $\boldsymbol{\theta}$, where $\boldsymbol{\lambda}$ is a vector of hyperparameters. The prior distribution represents the *belief* that we have on the true value of the parameter, expressed with an uncertainty encoded in the probability distribution that we specify. The main goal is to make inference on the parameters $\boldsymbol{\theta}$, and possibly on the hyperparameters $\boldsymbol{\lambda}$, as well as to assess the suitability of the chosen distributional model $f(\cdot|\cdot)$, called *likelihood.*

In the case when $\boldsymbol{\lambda}$ is known, the inference on $\boldsymbol{\theta}$ is based on the *posterior* distribution,

$$p(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\lambda}) = \frac{p(\mathbf{y}, \boldsymbol{\theta}|\boldsymbol{\lambda})}{p(\mathbf{y}|\boldsymbol{\lambda})} = \frac{p(\mathbf{y}, \boldsymbol{\theta}|\boldsymbol{\lambda})}{\int_\Theta p(\mathbf{y}, \boldsymbol{\theta}|\boldsymbol{\lambda}) d\boldsymbol{\theta}} = \frac{f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})}{\int_\Theta f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{\lambda}) d\boldsymbol{\theta}} \tag{1.13}$$

where Θ is a k-dimensional parametric space. The *posterior* distribution $p(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\lambda})$ computed as in (1.13), which is a form of the Bayes Theorem, is ideally an update of the distribution of the parameters (*prior* π) having observed the data modeled through the *likelihood* f. In practical cases, though, the true value of the hyperparameters $\boldsymbol{\lambda}$ is unknown, so we need to provide a distribution, called *hyperprior* $h(\boldsymbol{\lambda})$ which represents another level in the hierarchy of the model. Bayes Theorem, in the form of (1.13), becomes:

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}, \boldsymbol{\theta})}{p(\mathbf{y})} = \frac{\int_\Lambda f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})h(\boldsymbol{\lambda})\, d\lambda}{\iint_{\Lambda,\Theta} f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})h(\boldsymbol{\lambda})\, d\lambda\, d\theta} \tag{1.14}$$

where Λ is the parametric space of the hyperparameter vector $\boldsymbol{\lambda}$. In this perspective, it is clear why this approach is called *hierarchical*, in the sense that we can identify multiple levels of nested probabilistic distributions, from the likelihood specification to the priors and hyperprior densities. Since the primary interest of the Bayesian investigation lies in $\boldsymbol{\theta}$, it is a popular choice to avoid integrating over $\boldsymbol{\lambda}$, which can be computationally demanding, and to replace the unknown vector of hyperparameters with an estimate $\hat{\boldsymbol{\lambda}}$ obtained by maximizing the marginal distribution $p(\mathbf{y}|\boldsymbol{\theta}) = \int_\Theta f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})d\boldsymbol{\theta}$, viewed as a function of $\boldsymbol{\lambda}$. In this commonly used approach, called *empirical Bayes analysis*, inference can then be carried out computing the *estimated* posterior distribution $p(\boldsymbol{\theta}|\mathbf{y}, \hat{\boldsymbol{\lambda}})$, obtained by plugging $\hat{\boldsymbol{\lambda}}$ into (1.13)

As expressed in Banerjee et al. (2015), adopting the Bayesian inference paradigm instead of the classical frequentist approach yields some important advantages, such as including external knowledge and expertise into the model through the definition of prior distributions for the parameters.

Despite its benefits, the Bayesian method presents certain drawbacks, the primary one being the difficulty in computation that arises when implementing this approach. In most real-life cases, computing posterior distributions through (1.13) or (1.14) requires integrations not tractable in closed form. Thus, they must be approximated numerically using Markov Chain Monte Carlo (MCMC) integration methods, such as the Metropolis-Hastings algorithm (Metropolis et al. (1953); Hastings (1970)) and the Gibbs Sampler (Geman & Geman (1984); Gelfand & Smith (1990)). The strategy underlying these methods is to build a Markov Chain that eventually converges to the (possibly unknown) posterior distribution of the parameters and then to sample simulated values from this distribution. This set of values is then used in a Monte Carlo approach to compute interesting quantities, such as the posterior mean and variance (*point estimation*) or credibility intervals (*interval estimations*). In this work, we use alternative software, INLA, developed accordingly to the research in Lindgren et al. (2011) and Rue et al. (2009). The core idea of this technique is to avoid building Markov Chains and to approximate the posterior distributions with numerical methods. In this way, the approximation should be faster while at the same time granting accuracy and reliability in the estimation process. We will discuss the details of the algorithm in Chapter 2.

In Section 1.2.1, we will present the general hierarchical model for spatio-temporal data, finally justifying the theoretical premises outlined in Section 1.1.

1.2.1. The Spatio-temporal model in regression context

In this subsection, we discuss the theoretical model for general point-referenced data in a regression context and in a hierarchical way. As we will see in detail, the central hypothesis is that geostatistical data, observed repeatedly in time from monitoring stations across a spatial region, can be seen as the realizations of a Gaussian random process (see Section 1.1), identified as $Y(\cdot,\cdot)$. It can be helpful to imagine the realization as the concentration of PM10 at a specific station and on a given day (see Chapter 3).

Following the notation in Cameletti et al. (2013), let $y(\mathbf{s}_i, t)$ represent the realization of the process $Y(\cdot,\cdot)$ at the location $\mathbf{s}_i$, with $i = 1, ...d$ identifying the d measurement stations, and at time $t = 1, ..., T$. The model assumption for the distribution of the realizations is the following

$$y(\mathbf{s}_i, t) = z(\mathbf{s}_i, t)\boldsymbol{\beta} + \xi(\mathbf{s}_i, t) + \varepsilon(\mathbf{s}_i, t), \quad i = 1, ..., d \quad t = 1, ..., T. \tag{1.15}$$

In (1.15) we denote with $z(\mathbf{s}_i, t) = (z_1(\mathbf{s}_i, t), ..., z_p(\mathbf{s}_i, t))'$ the vector of p covariates for each site $\mathbf{s}_i$ at time t and $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)'$ as the related unknown coefficient vector. On the other hand, $\varepsilon(\mathbf{s}_i, t) \overset{iid}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2)$ is the measurement error, defined by a zero-mean Gaussian distribution (white noise), both spatially and temporally uncorrelated, with unknown variance σ_ε^2. The realization of the spatio-temporal GP is denoted as $\xi(\mathbf{s}_i, t)$. It is considered the most significant aspect of the model as it represents the unknown field of interest. Thus, the following paragraphs focus on the characteristics of this spatio-temporal stochastic process.

For the sake of clarity, we need to specify here a particular model choice, which may seem restrictive, but it is popular and flexible. In particular, we present the general structure we used in Chapter 3. The upcoming section provides the formal definition of the *Auto-Regressive* process, commonly used to model temporal correlation.

Auto-Regressive processes

An Auto-Regressive process of order p (AR(p)) is a stochastic process in which the value of the current observation at time t is modeled as a linear combination of the p previous observations, plus an error term. Mathematically, an AR(p) process can be defined as:

$$Y_t = c + a_1 Y_{t-1} + a_2 Y_{t-2} + ... + a_3 Y_{t-p} + \varepsilon_t, \tag{1.16}$$

where Y_t is the current observation, $\{a_1, a_2, ..., a_p\}$ are the autoregressive coefficients representing the impact of the previous p observations on the current value, ε_t is the error term assumed to be independently and identically distributed, and c is a constant term.

The order p determines the number of autoregressive coefficients to be estimated. The larger the value of p, the more complex the model and the longer the process memory. The autocorrelation function of an AR(p) process decays exponentially as the lag increases, with a rate determined by the magnitude of the autoregressive coefficients.

At this point we can model the spatio-temporal process $\xi(\cdot, \cdot)$. First of all, we assume the process to be separable, thus the covariances of the spatial and temporal parts are defined separately. In particular, we define the evolution in time with Auto-Regressive dynamics, with a spatially structured error term, $\omega(\cdot, \cdot)$. For the sake of simplicity, we consider an AR(1) model with parameter a. We want to emphasize that extending this to higher-order processes does not alter the nature of the problem but makes it more complex and computationally demanding.

The structure of ξ can be then written as

$$\xi(\mathbf{s}_i, t) = a\xi(\mathbf{s}_i, t-1) + \omega(\mathbf{s}_i, t), \quad t = 2, ..., T \tag{1.17}$$

$$\xi(\mathbf{s}_i, 1) \overset{iid}{\sim} \mathcal{N}\left(0, \frac{\sigma_\omega^2}{1-a^2}\right), \tag{1.18}$$

where $|a| < 1$; $\omega(\cdot, \cdot)$ is assumed to be a zero-mean Gaussian field, with a Matérn covariance structure for the spatial correlation:

$$Cov(\omega(\mathbf{s}_i, t), \omega(\mathbf{s}_j, t')) = \begin{cases} 0 & \text{if } t \neq t' \\ \sigma_\omega^2 C(h) & \text{if } t = t' \end{cases} \tag{1.19}$$

where $C(h)$ depends only on the length of the distance of the two stations $h = \|\mathbf{s}_i - \mathbf{s}_j\|$, and it is of the form of (1.7). From (1.19), it is clear that in each station $\mathbf{s}_i$, $Var(\omega(\mathbf{s}_i, t)) = \sigma_\omega^2$, and that, following the definitions given in Section 1.1, the process is weakly stationary and isotropic.

If we collect all the observations at time t, $\mathbf{y}_t = (y(\mathbf{s}_1, t), ..., y(\mathbf{s}_d, t))'$, we can rewrite (1.15) and (1.17) as follows:

$$\mathbf{y}_t = \mathbf{z}_t \boldsymbol{\beta} + \boldsymbol{\xi}_t + \boldsymbol{\varepsilon}_t, \qquad \boldsymbol{\varepsilon}_t \overset{iid}{\sim} \mathcal{N}(\mathbf{0}, \sigma_\varepsilon^2 I_d) \tag{1.20}$$

$$\boldsymbol{\xi}_t = a \boldsymbol{\xi}_{t-1} + \boldsymbol{\omega}_t, \qquad \boldsymbol{\omega}_t \sim \mathcal{N}(\mathbf{0}, \Sigma = \sigma_\omega^2 \tilde{\Sigma}), \quad t = 1, ..., T. \tag{1.21}$$

where $\mathbf{z}_t$ is the $d \times p$ matrix of the covariates, $\boldsymbol{\xi}_t$ is the zero-mean spatio-temporal Gaussian field with $\boldsymbol{\xi}_1$ distributed as a stationary AR(1) process $\mathcal{N}(\mathbf{0}, \Sigma/(1-a^2))$. Finally, $\tilde{\Sigma}$ is the $d \times d$ correlation matrix with elements $\tilde{\Sigma_{ij}} = C(\|\mathbf{s}_i - \mathbf{s}_j\|)$, given by the function (1.7), with parameters k and ν. However, we assume ν to be fixed, and therefore the parameters of the models can be written as $\boldsymbol{\theta} = (\boldsymbol{\beta}, \sigma_\varepsilon^2, a, \sigma_\omega^2, k)$.

Our goal now is to estimate parameters by computing their marginal posterior distributions and to summarize them with usual quantities, like mean, standard deviation, and quantiles. The posterior distribution, as we showed in 1.2, is computed using Bayes theorem, combining the likelihood and the prior densities of the parameters (see Banerjee et al. (2015) for more details), respectively $\mathcal{L}(\mathbf{y}|\boldsymbol{\xi}, \boldsymbol{\theta})$ and $\pi(\boldsymbol{\theta})$, where $\mathbf{y} = \{\mathbf{y}_t\}$ and $\boldsymbol{\xi} = \{\boldsymbol{\xi}_t\}$, with $t = 1, ..., T$, as in Cameletti et al. (2013). In our model we assume independent priors for $\boldsymbol{\beta}$, σ_ε^2, a, σ_ω^2 and k, that we will specify in Section 1.3.4. On the other hand, we point out that the process $\boldsymbol{\xi}$ is conditionally dependent on a, σ_ω^2 and k, as in (1.21).

As we already mentioned, the computation of the posterior distribution in (1.13) can be very complicated since the integration of the marginal in the divisor is often not in closed form. Therefore it must be computed in other ways. Nonetheless, it is important to notice that the posterior density, least for the marginal in the divisor, has the distributional form of the dividend, namely *posterior* $\propto$ *likelihood* $\times$ *prior*. In our case, the kernel of the posterior is given by

$$\pi(\boldsymbol{\theta}, \boldsymbol{\xi}|\mathbf{y}) \propto \mathcal{L}(\mathbf{y}|\boldsymbol{\xi}, \boldsymbol{\theta}) \pi(\boldsymbol{\xi}|\boldsymbol{\theta}) \pi(\boldsymbol{\theta}) \tag{1.22}$$

where $\pi(\cdot)$ is used to denote the density function of $\boldsymbol{\xi} = \{\boldsymbol{\xi}_t\}$ and $\boldsymbol{\theta}$, and $\mathcal{L}(\cdot)$ for the likelihood of the data. Considering that the observation vectors $\mathbf{y}_t$ are independent conditionally on $\boldsymbol{\xi}$ and that we assumed independent priors, we can rewrite (1.22) as

$$\pi(\boldsymbol{\theta}, \boldsymbol{\xi}|\mathbf{y}) \propto \left(\prod_{t=1}^{T} \mathcal{L}(\mathbf{y}_t|\boldsymbol{\xi}_t, \boldsymbol{\theta})\right) \times \left(\pi(\boldsymbol{\xi}_1|\boldsymbol{\theta}) \prod_{t=2}^{T} \pi(\boldsymbol{\xi}_t|\boldsymbol{\xi}_{t-1}, \boldsymbol{\theta})\right) \times \pi(\boldsymbol{\theta}) \tag{1.23}$$

By plugging in the distributions defined in (1.20) and (1.21), together with the prior distributions, we would get the specific posterior kernels. In our case, the idea is to approximate the posterior distributions with the algorithm presented in Rue et al. (2009), which represents an alternative to standard MCMC techniques.

1.3. From modeling to computation

In this section, we present the main definitions and results that lead us from the purely theoretical model proposed in Section 1.2 to the numerical approximation that will yield estimations of the parameters of interest. The problem we have to face is how to make posterior inference on the parameters of the Matérn Gaussian Field from the available data, which are realizations of the process. Initially, we focus on the case where our data are realizations of the stochastic process $\{Y(\mathbf{s}), \mathbf{s} \in D \subseteq \mathbb{R}^2\}$, which is a Gaussian Process, isotropic and weak stationary, with a Matérn-like covariance function, that depends on the parameters k and ν. The blueprint of the approach is to approximate this GP, which has a continuous domain D, with finite elements representation. In particular, we look for a finite-dimensional Gaussian Field with a sparse precision matrix, fundamental to speed up computations. This field is called a Gaussian Markov Random Field. We briefly discuss its properties in Section 1.3.1, referring to Section 3 of Cameletti et al. (2013) and to Lindgren et al. (2011) for more technical details.

1.3.1. Introduction to Gaussian Markov Random Fields (GMRFs)

A GMRF is a finite-dimensional spatial process used to model the spatial correlation of georeferenced data. Suppose we have n observation points, then we refer to a GMRF as $\mathbf{y} = (y_1, ..., y_n)'$, such that $\mathbf{y} \sim N(\boldsymbol{\mu}, \mathbf{Q}^{-1})$. The choice to emphasize the precision matrix $\mathbf{Q}$, which is the inverse of the covariance matrix, will become clearer later.

The density of the $n-$dimensional GMRF is:

$$\pi(\mathbf{y}) = (2\pi)^{-n/2}|\mathbf{Q}|^{1/2} exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})'\mathbf{Q}(\mathbf{y}-\boldsymbol{\mu})\right) \tag{1.24}$$

For a detailed discussion, we refer to Rue & Held (2005), but for our purpose, the two important properties that define a GMRF are the following:

- the law of $\mathbf{y}$ is specified through the collection of full conditional distributions $\pi(y_i|\mathbf{y}_{-i})$, denoting the distribution of a component given all the others $(\mathbf{y}_{-i})$;
- $\pi(\mathbf{y})$ has the Markovian property, namely that full conditional distribution depends only on a few neighboring components,

  $$\pi(y_i|\mathbf{y}_{-i}) = \pi(y_i|\mathbf{y}_{\delta_i}), \tag{1.25}$$

  where δ_i is the set of neighbors of unit i

The formal presence of a neighboring structure is fundamental in our context. In this way, the value of the GMRF depends on surrounding locations, reflecting the same spatial property of the underlying GP $\xi(\cdot|\cdot)$. Moreover, the Markovian property is the property that justifies the use of a GMRF in computations. Indeed, as recalled in Cameletti et al. (2013), this property is the equivalent of assuming that y_i is independent from $\mathbf{y}_{-\{i,\delta_i\}}$, conditionally on $\mathbf{y}_{\delta_i}$. In the notation of Rue & Held (2005), we have that

$$y_i \perp \mathbf{y}_{-\{i,\delta_i\}} \mid \mathbf{y}_{i,\delta_i} \tag{1.26}$$

for $i = 1, ..., n$. This conditional independence property is related to the precision matrix $\mathbf{Q}$, since

$$y_i \perp y_j \mid \mathbf{y}_{-\{i,j\}} \iff \mathbf{Q}_{ij} = 0 \tag{1.27}$$

for a generic couple $i \neq j$. This is the key point: the precision matrix will have non-zero elements in positions related to the neighboring structure, leading to a possible sparse matrix. We can say that

$$\mathbf{Q}_{ij} \neq 0 \quad \text{if } j \in \{i, \delta_i\} \tag{1.28}$$

The sparsity of a GMRF justifies its use for approximating a GP since linear algebra computations can be done much faster. For example, if we consider the factorization of a dense $n \times n$ matrix usually costs $\mathcal{O}(n^3)$, dropping to $\mathcal{O}(n^{3/2})$, $\mathcal{O}(n)$ and to $\mathcal{O}(n^2)$ for respectively spatial, temporal and spatio-temporal GMRF with a sparse precision matrix.

We also point out why we chose to express the GMRF with the precision matrix and not, as usual, with the covariance matrix. Indeed, the sparsity implied by the Markovian property regards only $\mathbf{Q}$, not the covariance matrix Σ. A simple but effective example that can help us understand this last argument is the autoregressive process (of order 1, for the seek of simplicity); keeping the notation of Cameletti et al. (2013), the process $\mathbf{y} = (y_1, ..., y_n)$ can be written as:

$$y_t = ay_{t-1} + \varepsilon_t \qquad \varepsilon_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \tag{1.29}$$

$$y_1 \sim \mathcal{N}\left(0, \frac{\sigma^2}{1-a^2}\right) \tag{1.30}$$

with $t = 2, ..., n$ and $|a| < 1$. This process fulfills the Markovian property, in fact, in terms of full conditional distributions,

$$\pi(y_t|\mathbf{y}_{-t}) = \pi(y_t|y_{t-1}, y_{t+1}) \tag{1.31}$$

For this process, the precision matrix $\mathbf{Q}$ is sparse, precisely tridiagonal:

$$Q = \begin{pmatrix} \sigma^2 & -a/\sigma^2 & & & \\ -a/\sigma^2 & (1+a^2)/\sigma^2 & & & \\ & & \cdots & & \\ & & & (1+a^2)/\sigma^2 & -a/\sigma^2 \\ & & & -a/\sigma^2 & \sigma^2 \end{pmatrix}$$

with zero entries where not specified, namely outside the three central diagonals. On the other hand, the covariance matrix is not sparse, justifying the choice to express a GRMF through its precision matrix.

At this point, the theoretical model for spatio-temporal Gaussian processes is complete. However, to apply it in a practical framework with the software INLA we still have to introduce the numerical approach and results, through which the approximation of the posterior distributions of $\boldsymbol{\theta}$ is computed: the Stochastic Partial Differential Equation (SPDE) method. In the next chapter, we present and resume the main concepts formulated in Lindgren et al. (2011) and revised in Lindgren et al. (2022).

1.3.2. The Stochastic Partial Differential Equation approach

After defining the notion and properties of GMRFs, we need to understand how they can approximate GPs with the Matérn covariance function. For this purpose, we discuss the results of Lindgren et al. (2011) and Lindgren et al. (2022). In particular, to clarify the link between this approach and our hierarchical model (1.21), we point out that our goal is to explain how to approximate the spatial residual $\boldsymbol{\omega_t}$, which is a Gaussian process, with a proper GMRF that reflects the properties of the Matérn covariance structure of the infinite-dimensional, theoretical process. The next step will be to approximate the whole spatio-temporal process $\boldsymbol{\xi_t}$ with the same strategy.

The core idea of this method is to build a finite-dimensional Hilbert space of functions and project the infinite dimension Matérn field, $\boldsymbol{\omega_t}$ in our case, onto this space. In particular, Lindgren et al. (2011) identified a finite collection of local piecewise linear basis functions to span the finite-dimensional space. With this choice, the authors proved that the projection of the Matèrn field (assumed to have Markovian properties) carries Markovian properties to the weights combining the basis functions.

Following the notation of Lindgren et al. (2022), we call $u(\cdot)$ a stochastic process over a continuous domain $D \subseteq \mathbb{R}^d$ and we assume that $\mathbb{E}(\{u(\mathbf{s})\}) = 0$ and that $Cov(u(\mathbf{s}), u(\mathbf{s}')) = \rho_M(\mathbf{s}, \mathbf{s}')$ is generic covariance function.

If we denote with $\mathcal{W}(\cdot)$ a Gaussian white noise process on D, SPDEs of the form

$$\mathcal{L}u(\cdot) = \mathcal{W}(\cdot) \tag{1.32}$$

can be used to define random fields $u(\cdot)$ where the choice of operator $\mathcal{L}$ determines the structure of the covariance matrix of $u(\cdot)$ (see Lindgren et al. (2022)). In particular, Whittle (1954) and Whittle (1963) proved that the choice of the operator

$$\tau(k^2 - \Delta)^{\alpha/2} u = \mathcal{W}, \tag{1.33}$$

in (1.32), where Δ is the Laplacian operator, implies that the stationary solution $u(\cdot)$ has a Matérn covariance function of the form (1.7), with

$$\nu = \alpha - d/2 \tag{1.34}$$

$$\sigma^2 = \frac{\Gamma(\nu)}{\Gamma(\alpha)(4\pi)^{d/2} k^{2\nu} \tau^2} \tag{1.35}$$

Before exploring the details of the solution to this equation, we need to introduce some general notation. In particular, we call H the function space where u belongs, essentially a Sobolev space, namely $u \in H$, endowed with the inner product $< \cdot, \cdot >$. Then $\mathcal{L}$ is an operator such that $\mathcal{L} : H \to H$.

At this point, the problem is to build a finite-dimensional representation of the SPDE solution. For this purpose, Lindgren et al. (2011) proposed to use an orthonormal basis of piecewise linear functions with local support on a spatial triangulation. The idea is to divide the domain D into nonintersecting triangles that share at most one edge, resulting in a set of nodes, $j = 1, ..., N$, and to define the basis for the projection through a set of piecewise linear functions, $\{\psi_j(\mathbf{s}), j = 1, ..., N\}$, such that $\psi_j(\mathbf{s})$ is 1 at vertex j and 0 on all other vertices. The finite-dimensional representation of the solution is found by solving the weak form of (1.33), using the basis functions as test functions. Specifically, Lindgren et al. (2011) proposed to use as test functions:

- ψ_j for $\alpha = 2$
- $\mathcal{L}\psi_j$ for $\alpha = 1$

The equation, for $\alpha = 2$, becomes then

$$< \psi_j, \mathcal{L}u >=< \psi_j, \mathcal{W} > \quad \text{for } j = 1, ..., N. \tag{1.36}$$

To solve this equation explicitly, i.e. to compute the precision matrix $\mathbf{Q}$ of the GMRF that approximates the continuous GP $u(\cdot)$, we have to explicit the basis expansion and plug it into the weak formulation of the SPDE. In particular, we assume that $u(\cdot)$ can be approximated by

$$\tilde{u}(\mathbf{s}) = \sum_{j=1}^{N} u_j \psi_j(\mathbf{s}), \tag{1.37}$$

where the coefficients $\{u_j, j = 1, ..., N\}$ will represent the GMRF, approximating the original GP. By substituting (1.37) and the proper operator $\mathcal{L}$ in (1.36), we obtain the following weak formulation:

$$\left[< \psi_i, \sum_{j=1}^{N}(k^2 - \Delta)\psi_j(\cdot)u_j >\right]_{i=1,...,N} = [< \psi_i, \mathcal{W} >]_{i=1,...,N} \tag{1.38}$$

$$\left[\sum_{j=1}^{N} < \psi_i, (k^2 - \Delta)\psi_j > u_j\right]_{i=1,...,n} = [< \psi_i, \mathcal{W} >]_{i=1,...,N} \tag{1.39}$$

$$(k^2 C + G)\mathbf{u} = [< \psi_i, \mathcal{W} >]_{i=1,...,N} \, , \tag{1.40}$$

where the matrices C and G are such that

$$C_{ij} =< \psi_i, \psi_j > \tag{1.41}$$

$$G_{ij} =< \nabla\psi_i, \nabla\psi_j > . \tag{1.42}$$

At this point, since on the left side of (1.36) we have that

$$[Cov(<\psi_i, \mathcal{W}>, <\psi_j, \mathcal{W}>)]_{i,j=1}^{N} = C \tag{1.43}$$

we can analytically find the precision matrix Q of the GMRF:

$$(k^2 C + G)\mathbf{u} \sim \mathcal{N}(0, C) \tag{1.44}$$
$$Q = (k^2 C + G)C^{-1}(k^2 C + G). \tag{1.45}$$

In the next figure we can see an example, from Cameletti et al. (2013) of the approximation of a GP with a GMRF through the SPDE approach.

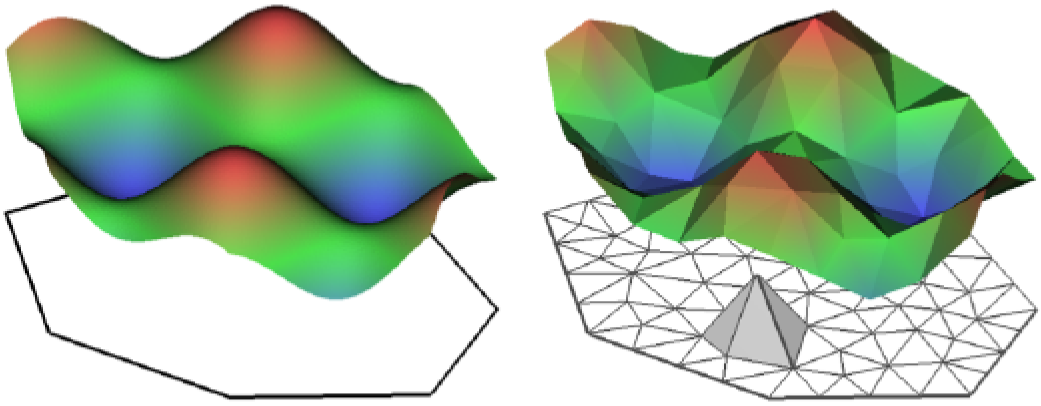


Figure 1.1: Left panel: example of a spatial random field (left) given by $u(\mathbf{s})$, where $\mathbf{s} = \{s_1, s_2\}$. Right panel: corresponding finite element representation of the spatial random field $\tilde{u}(\mathbf{s})$ according to (1.37)

Finally, we briefly extend this approach to a spatio-temporal GP, $\{u(\mathbf{s}, t),\ \mathbf{s} \in D, t \in \mathbb{R}\}$. In this case, depending on the structure of the temporal correlation, the operator $\mathcal{L}$ is different. For example, as specified in Lindgren et al. (2022), for a Matérn field with AR(1) temporal dynamics, the SPDE becomes:

$$\left(a + \frac{\partial}{\partial t}\right)\left(k^2 - \Delta_s\right)^{\alpha/2} u = \mathcal{W}, \tag{1.46}$$

where Δ_s is the Laplacian operator concerning the spatial coordinates.

1.3.3. Rewriting the model

In this subsection, we revise the model described by the equation (1.15), considering the approximation of the GP with a GMRF having precision matrix Q parameterized by k and ν.

The first step to building the GMRF is the partition of the domain. Intuitively, we start from the locations of the observations, the d stations $\{s_1, ..., s_d\}$, that are the starting node of the triangulation. From these points, we can build a *mesh* with the desired number of nodes, N, as we will see later in Chapter 3. Figure 1.2 represents a simple mesh generated from the monitoring stations across Emilia-Romagna with an `INLA` specific function.

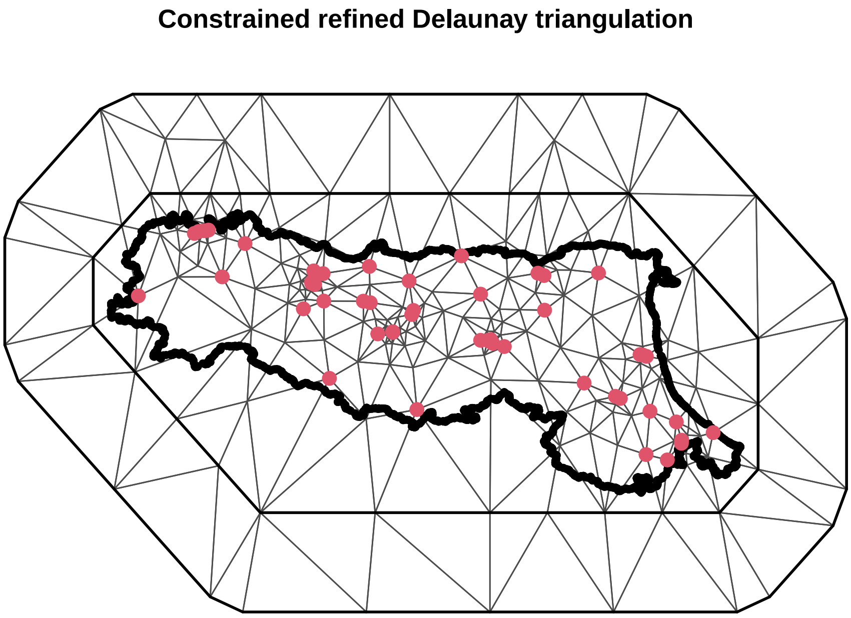


Figure 1.2: A mesh defined using the software INLA, built starting from the measurement stations (red points) in Emilia-Romagna

Then, we can represent the random field $\xi(\cdot, \cdot)$ with the GMRF defined on the mesh. For $t = 1, ..., T$, we can now represent the Matérn field $\boldsymbol{\omega}_t$ in (1.21) through the GMRF $\tilde{\boldsymbol{\omega}}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q_s}^{-1})$, where $\mathbf{Q_s}$ is the precision matrix found through the SPDE approach, discussed in 1.3.2. The precision matrix does not change in time, thanks to the assumption of separability of the spatio-temporal model, and has dimension $N \times N$, where N is the number of mesh nodes. The model can be rewritten as

$$\tilde{\boldsymbol{\xi}}_t = a\tilde{\boldsymbol{\xi}}_{t-1} + \tilde{\boldsymbol{\omega}}_t, \quad \tilde{\boldsymbol{\omega}}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q_s}^{-1}) \quad t = 1, ..., T \tag{1.47}$$

$$\tilde{\boldsymbol{\xi}}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{Q_s}^{-1}/(1-a^2)). \tag{1.48}$$

The new GMRF $\tilde{\boldsymbol{\xi}} = (\tilde{\boldsymbol{\xi}}'_1, ..., \tilde{\boldsymbol{\xi}}'_T)$ has a $TN - dimensional$ multinormal distribution, with 0 mean and $\mathbf{Q}$ precision matrix, with

$$\mathbf{Q} = \mathbf{Q}_T \otimes \mathbf{Q}_S, \tag{1.49}$$

where $\mathbf{Q}_T$ the T-dimensional precision matrix of the temporal dynamics (see for example the explicit form for the AR(1) in Section 1.3.1). The product $\otimes$ in 1.49 is the Kroneker product that combines the sparse precision matrices of the spatial and temporal fields, of dimensions N and T, respectively, into a $TN \times TN$ dimensional one.

Finally we can rewrite (1.20) incorporating the GMRF, in the following way:

$$\mathbf{y}_t = \mathbf{z}_t\boldsymbol{\beta} + \mathbf{B}\tilde{\boldsymbol{\xi}}_t + \boldsymbol{\varepsilon}_t, \qquad \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2_\varepsilon I_d) \tag{1.50}$$

where $\mathbf{B}$ is the $d \times N$-dimensional matrix that selects the raw of the GMRF relative to the d observation stations. It has only one unit element per row that picks the index of the $i - th$ station from the N nodes of the mesh.

As we introduced before, in Section 1.2.1, to complete the Bayesian model, we need to specify the priors assigned to the parameters. Although the choices can be very different, we present in the next section a specific choice that we used for the analysis carried out in Chapter 3.

1.3.4. Prior Specifications

As explained in Section 1.2, we can incorporate external information into the model by defining prior distributions for the parameters.

Instead of assigning marginal priors to the variances parameters σ^2_ε and σ^2_ω, we consider the precisions $\tau_\omega = \frac{1}{\sigma^2_\omega}$ and $\tau_\varepsilon = \frac{1}{\sigma^2_\varepsilon}$. Moreover, regarding the parameter k in the Matern covariance function (1.7), the prior distribution is directly given to the transformation (1.9) to highlight the effective range. In this perspective, the parameter vector becomes $\boldsymbol{\theta} = (\boldsymbol{\beta}, \tau_\varepsilon, a, \tau_\omega, \rho)$.

In this context, we propose a possible selection of priors for the parameters assumed in the case study discussed in Chapter 3

- $\beta_i \overset{iid}{\sim} \mathcal{N}(0, 10^{-5})$;
- $\tau_\varepsilon \sim gamma(1, 5 \cdot 10^{-5})$;
- $logit(a) \sim \mathcal{N}(0, 0.15)$.

We specify here that the prior on the AR(1) parameter is assigned to the logit transformation to force its value to be in $[0, 1]$.

At this point, we must introduce a new category of priors, widely used in the package, to cover the remaining parameters. We refer to Chapter 5 of Gomez-Rubio (2020) for further details about this class of densities.

Penalized-Complexity Priors

Simpson et al. (2017) proposed a novel approach for assigning priors to parameters of a specific class of models, namely additive models with different components. The underlying idea is that any form of complex and informative prior distribution is a flexible extension of a simpler base model. In this perspective, one should stick to the base model until enough counterarguments are found. The suggested prior densities are called Penalized Complexity priors (PC) since they penalize the departure from the so-called base model, the simplest possible model for the specific parameter. Occam's Razor is the underlying principle: lacking counter-evidence, a simpler model ought to be chosen. The application of this principle, carried out in Simpson et al. (2017), leads to the introduction of a penalization of the distance from the base model, measured with the Kullback-Leiber distance. Of course, depending on the parameter, the PC prior is different, as the base model is. The result is a class of distributions, each suited for a specific parameter and implemented in the INLA software. Furthermore, it is possible to define these densities by making probability statements about the parameters in a specific manner. For example, we can assign PC priors to the parameters of the Matérn field in our model, in particular to the standard deviation σ_ω and the range ρ. For the standard deviation the hyperparameters are σ_0 and α, so that

$$\mathbb{P}(\sigma > \sigma_0) = \alpha \tag{1.51}$$

while for the range parameter ρ, the PC prior definition is done through the hyperameters

ρ_0 and p_ρ:, with the following request:

$$\mathbb{P}(\rho < \rho_0) = p_\rho. \tag{1.52}$$

As described in Section 3.2 of Simpson et al. (2017), in the case of our model, the particular PC prior assigned to the precision of a Gaussian random effect τ_ω for the GP with Matérn covariance function is a type 2 Gumbel distribution, with density

$$\pi(\tau_\omega) = \frac{\lambda}{2}\tau_\omega^{-3/2}\text{exp}\left(-\lambda\tau_\omega^{-1/2}\right), \quad \tau_\omega > 0. \tag{1.53}$$

Parameter λ is set through the probability statement (1.51).

Now, we can proceed with fitting the data to obtain posterior inference on the parameters $\boldsymbol{\theta}$. In Chapter 2, we will delve into the details of the algorithm implemented in the *R-INLA* package, which was initially introduced in Rue et al. (2009). This algorithm exploits the sparsity of the precision matrix to construct rapid numerical approximations of the posterior distributions without resorting to conventional MCMC methods.

2 | Integrated Nested Laplace Approximation (INLA)

In this chapter, we will present the strategy implemented in the R-INLA package to make posterior inference on model parameters in a Bayesian framework. The name indicates the method implemented in the model, which performs an Integrated Nested Laplace Approximation of the posterior distributions. We hope to provide a general comprehension of the technique without delving too much into the details presented in Rue et al. (2009).

2.1. Introduction

The strategy aims at performing inference in latent Gaussian models, namely a subclass of additive regression models. Following the notation of Rue et al. (2009), in these models, the response variable y_i belongs to an exponential family, with mean μ_i related to an additive predictor η_i through a link function $\eta_i = g(\mu_i)$. This class of models is flexible and widely adopted because of the wide range of choices that can be made for the link function $g(\cdot)$.

The structured additive predictor η_i is a sum of different effects:

$$\eta_i = \alpha + \sum_{j=1}^{n_f} f^{(j)}(u_{ji}) + \sum_{k=1}^{n_\beta} \beta_k z_{ki} + \varepsilon_i, \quad i = 1, ..., n_d, \tag{2.1}$$

where n_f and n_β respectively represent the number of modeled functions f and the number of covariates taken into consideration, each associated with a coefficient β_k. In this most general formulation, $\{f^{(j)}(\cdot)\}$ are generic functions of the covariates $\mathbf{u}$; in particular we could have n_f different functions, each one applied for a specific covariate u_{ji}. We can expand (2.1) as:

$$\eta_i = \alpha + f^{(1)}(u_{1i}) + f^{(2)}(u_{2i}) + ... + f^{(n_f)}(u_{n_f i}) + \sum_{k=1}^{n_\beta} \beta_k z_{ki} + \varepsilon_i, \quad i = 1, ..., n_d.$$

On the other hand, the $\{\beta_k\}$s are the unknown coefficients representing the linear effects of covariates $\mathbf{z}$, structured in the same way as the nonlinear effects modeled by f. Finally, ε_i is the uncorrelated error term.

We will focus on latent Gaussian models, namely on models in which we assign a Gaussian prior to α, $\{f^{(j)}(\cdot)\}$, $\{\beta_k\}$ and ε_i. To clarify the connection with our spatio-temporal model, presented in Chapter 1 we have that $\mathbf{u} = u(\mathbf{s}, t)$ and f represents the stochastic process $\boldsymbol{\xi}$ in (1.21). We now introduce the notation necessary to proceed with the discussion, following Rue et al. (2009). In general, we denote with $\pi(\cdot|\cdot)$ a conditional density, and let $\mathbf{x}$ be the collection of the Gaussian variables $\{\eta_i\}$, $\{f^{(j)}(\cdot)\}$ and $\{\beta_k\}$. We assume the density $\pi(\mathbf{x}|\boldsymbol{\theta}_1)$ to be a zero-mean Gaussian with precision matrix $\mathbf{Q}(\boldsymbol{\theta}_1)$. The law of the observational variables $\mathbf{y} = \{y_i,\ i \in \mathcal{I}\}$ can be denoted with $\mathcal{L}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_2)$, assuming y_i conditionally independent given $\mathbf{x}$ and $\boldsymbol{\theta}_2$. If we denote $\boldsymbol{\theta} = (\boldsymbol{\theta}_1', \boldsymbol{\theta}_2')$, we can express the kernel of the posterior distribution as:

$$\begin{aligned}\pi(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y}) &\propto \pi(\boldsymbol{\theta})\pi(\mathbf{x}|\boldsymbol{\theta}) \prod_{i\in\mathcal{I}} \mathcal{L}(y_i|x_i, \boldsymbol{\theta}) \\ &\propto \pi(\boldsymbol{\theta})|\mathbf{Q}(\boldsymbol{\theta})|^{1/2}\text{exp}\left[-\frac{1}{2}\mathbf{x}'\mathbf{Q}(\boldsymbol{\theta})\mathbf{x} + \sum_{i\in\mathcal{I}} log\{\mathcal{L}(y_i|x_i, \boldsymbol{\theta})\}\right].\end{aligned}$$

Our aim is now to approximate the posterior marginals of $\pi(x_i|\mathbf{y})$, $\pi(\boldsymbol{\theta}|\mathbf{y})$ and $\pi(\theta_j|\mathbf{y})$, where θ_j represent the generic hyperparameter for the latent Gaussian field (for example k in the case of a Matérn-like stochastic process).

The general strategy for inference is to approximate the posteriors exploiting the integrals:

$$\begin{aligned}\pi(x_i|\mathbf{y}) &= \int \pi(x_i|\boldsymbol{\theta}, \mathbf{y})\pi(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta} \\ \pi(\theta_j|\mathbf{y}) &= \int \pi(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}_{-j},\end{aligned}$$

where x_i is the generic Gaussian variable and $d\boldsymbol{\theta}_{-j}$ means that we integrate over all components of $\boldsymbol{\theta}$, with the exception of θ_j. The next step is to build nested approximations

$$\tilde{\pi}(x_i|\mathbf{y}) = \int \tilde{\pi}(x_i|\boldsymbol{\theta}, \mathbf{y})\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta} \tag{2.2}$$

$$\tilde{\pi}(\theta_j|\mathbf{y}) = \int \tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}_{-j}, \tag{2.3}$$

where $\tilde{\pi}(\cdot|\cdot)$ represents an approximated conditional density of the arguments. At this point, it is clear that, to compute $\tilde{\pi}(x_i|\mathbf{y})$ we need to approximate $\pi(\boldsymbol{\theta}|\mathbf{y})$ and $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$ and to integrate out, with a numerical method, $\boldsymbol{\theta}$.

2.2. The Inference strategy

As mentioned before, we refer to Rue et al. (2009) for further details; nonetheless, in this section, we present the overall strategy behind the Integrated Nested Laplace Approximation. This method is mainly composed of three steps that will eventually yield the targeted approximation $\tilde{\pi}(x_i|\mathbf{y})$. The first step is to find a proper approximation of $\pi(\boldsymbol{\theta}|\mathbf{y})$. Then, with the help of the previous computation, the (Laplace) approximation of $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$ is computed. Finally, with numerical integration, we compute $\tilde{\pi}(x_i|\mathbf{y})$, in the sense of (2.2).

We start by studying a possible approximation of the posterior densities of the hyperparameters $\boldsymbol{\theta}$, namely by defining:

$$\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y}) \propto \left. \frac{\pi(\mathbf{x}, \boldsymbol{\theta}, \mathbf{y})}{\tilde{\pi}_G(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})} \right|_{\mathbf{x}=\mathbf{x}^*(\boldsymbol{\theta})}, \tag{2.4}$$

where $\tilde{\pi}_G(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$ and $\mathbf{x}^*(\boldsymbol{\theta})$ are the Gaussian approximation and the mode of the full conditional distribution of $\mathbf{x}$, respectively.

As discussed in Section 2.2 of Rue et al. (2009), the Gaussian approximation $\tilde{\pi}_G(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$ in (2.4), is computed by matching the modal configuration and the curvature at the mode $\boldsymbol{\mu}$. The latter is computed iteratively using a Newton-Raphson method, described in detail in Section 2.2 of Rue et al. (2009). In particular, the goal is to approximate the densities of the form

$$\pi(\mathbf{x}) \propto \exp\{-\frac{1}{2}\mathbf{x}'\mathbf{Q}\mathbf{x} + \sum_{i \in \mathcal{I}} g_i(x_i)\}, \tag{2.5}$$

where $g_i(x_i)$ is $\log\{\pi(y_i|x_i, \boldsymbol{\theta})\}$ in our case.

In (2.5) can also see the link with Section 1.3, through the precision matrix $\mathbf{Q}$ of the latent GMRF, which is built following the SPDE approach. Recalling our specific model, this step of the algorithm is responsible for computing the posterior marginal of the spatio-temporal latent field, namely a GMRF with autoregressive dynamics ($\tilde{\boldsymbol{\xi}}$ in our notation, which is one of the components of $\mathbf{x}$).

At this point we have computed the Gaussian approximation $\tilde{\pi}_G(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$ centered around the estimated mode $\boldsymbol{\mu}$, to be used in (2.4) to get the kernel of the distribution of $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$. We clarify now that this strategy computes the approximation of the density of $\boldsymbol{\theta}$ given $\mathbf{y}$ up to a normalization constant, which is still unknown (we have a proportionality sign in (2.4)). Nonetheless, it is sufficient for our scope since we need that density just for the numerical integration of (2.2). In particular, we do not need to represent parametrically $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$, but we just need to explore it and select *good* values $\boldsymbol{\theta_k}$ for the numerical integration

$$\tilde{\pi}(x_i|\mathbf{y}) = \sum_k \tilde{\pi}(x_i|\boldsymbol{\theta_k}, \mathbf{y})\tilde{\pi}(\boldsymbol{\theta_k}|\mathbf{y})\Delta_k, \tag{2.6}$$

where Δ_k are area weights for each explored value $\boldsymbol{\theta_k}$.

In Sections 2.2.1 and 2.2.2, we will briefly outline the two steps that are needed to finally perform the numerical integration in (2.6). The steps are :

- explore $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$ to get good values $\boldsymbol{\theta_k}$,
- compute the approximating distribution $\tilde{\pi}(x_i|\boldsymbol{\theta}, \mathbf{y})$, with different possible methods.

As a result, we can compute the weighted sum in (2.6).

2.2.1. Exploring $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$

As explained in Section 3.1 of Rue et al. (2009), in which the authors present the strategy to explore $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$, we need the distribution defined in (2.4), to integrate out the uncertainty of $\boldsymbol{\theta}$ in the numerical integration (2.6), by selecting good evaluation points $\boldsymbol{\theta_k}$.

The procedure consists of three steps, summed up in the following paragraphs.

- **Step 1**: we locate the mode of $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$, identified as $\boldsymbol{\theta}^*$ with a quasi-Newton method that computes an approximation to the second derivative of $\log\{\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})\}$;
- **Step 2**: compute the negative Hessian matrix $\mathbf{H} > 0$, at the modal configuration $\boldsymbol{\theta}^*$. Then, compute the standardized variable $\mathbf{z}$, with the eigendecomposition of $\mathbf{H}^{-1} = \boldsymbol{\Sigma} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathbf{T}}$, and define $\boldsymbol{\theta}$ in function of $\mathbf{z}$ as:

  $$\boldsymbol{\theta}(\mathbf{z}) = \boldsymbol{\theta}^* + \mathbf{V}\boldsymbol{\Lambda}^{1/2}\mathbf{z}. \tag{2.7}$$

  The new coordinate system, defined through this reparametrization, is centered at the mode and has orthogonal axes given by the directions of $\mathbf{z}$.
- **Step 3**: this is the actual exploration part, in which, through the $\mathbf{z}$ variable, we explore $\log\{\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})\}$ to locate the values $\boldsymbol{\theta_k}$ that represent the bulk of the probability mass. Now, for the sake of simplicity, assume we have $\boldsymbol{\theta} = (\theta_1, \theta_2)'$, and therefore $\mathbf{z} = (z_1, z_2)$, as in the left panel of Figure 2.1. The idea is to start from the mode, i.e. $\mathbf{z} = \mathbf{0}$ and move first in the direction of z_1 with step length δ_z, and two save

values of θs as long as :

$$log\{\tilde{\pi}(\boldsymbol{\theta}(\mathbf{0})|\mathbf{y})\} - log\{\tilde{\pi}(\boldsymbol{\theta}(\mathbf{z})|\mathbf{y})\} < \delta_{\pi}, \tag{2.8}$$

for some δ_{π} that can be tuned (Rue et al. (2009) propose in Section 3.1 $\delta_z = 1$ and $\delta_{\pi} = 2.5$). The same can be done by exploring direction z_2; those values are the filled dots in the right panel of 2.1, while the empty ones are found by exploring other combinations of the two coordinates in the same way as in (2.8). This way, we get a set of interesting $\boldsymbol{\theta_k}$ to be used in (2.6), with all equal weights Δ_ks.

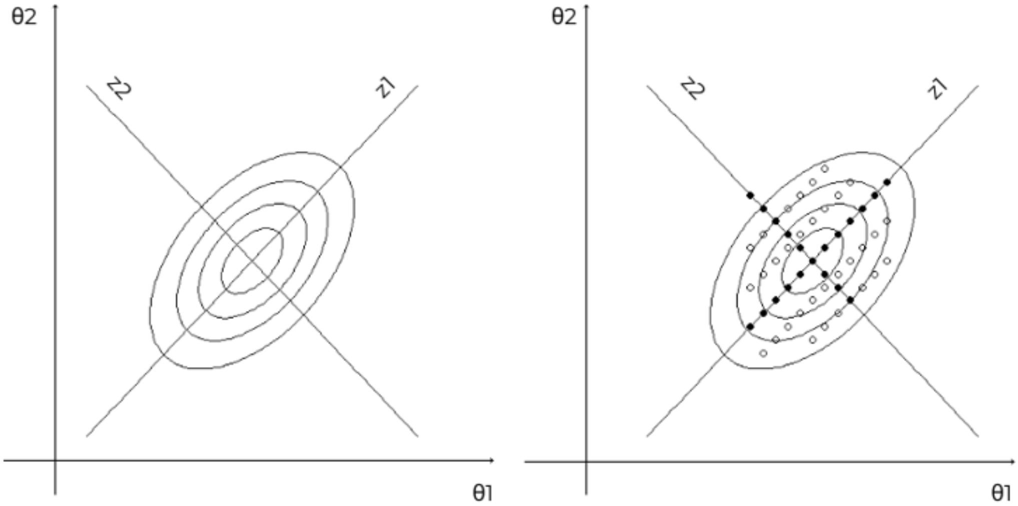


Figure 2.1: Left panel: display of the new coordinates system, centered at the mode, computed through (2.7). Right panel: values of $\boldsymbol{\theta}$ that have been explored and picked, following (2.8).

At this point, one may wonder if, beyond finding the evaluation points used for the numerical integration in (2.6), one could compute the marginal distributions $\tilde{\pi}(\theta_j|\mathbf{y})$ for the different components of $\boldsymbol{\theta}$. The answer is affirmative. For this purpose, the authors of Rue et al. (2009) suggest a numerical interpolation of $\log\{\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})\}$ based on the values $\boldsymbol{\theta_k}$ found through the three steps discussed before. Then the posterior marginals for the components are computed by numerical integration from the interpolant. Of course, we can produce a more accurate approximation by imposing a denser algorithm configuration (namely, smaller δ_z).

Finally, in Section 2.2.2, we will see the methods to approximate $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$, necessary for the final goal: find $\tilde{\pi}(x_i|\mathbf{y})$ through (2.6).

2.2.2. Approximating $\pi(x_i|\boldsymbol{\theta}, \mathrm{y})$

As already mentioned before, in order now to achieve our goal to compute posterior marginals of x_i, we need to approximate $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$. We briefly discuss three different methods proposed in Rue et al. (2009), referring for further details to Section 3.2. We also recall, that the approximating distribution $\tilde{\pi}(\cdot|\cdot)$ is conditioned on the selected values $\boldsymbol{\theta_k}$ found in Section 2.2.1.

As mentioned before, we discuss three different ways to compute $\tilde{\pi}(x_i|\boldsymbol{\theta_k}, \mathbf{y})$, following the Sections 3.2.1, 3.2.2, 3.2.3 of Rue et al. (2009).

- **Gaussian approximations**: the simplest approximation is to use directly $\tilde{\pi}_G(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$, which has already been computed for the exploration of $\pi(\theta_i|\mathbf{y})$ (see Section 2.2, and in particular definition (2.4)). This approach is problematic since it is not accurate enough: the Gaussian approximation is too simple and symmetric.

- **Lalpace approximations**: the common improvement of the Gaussian approximation is the Laplace approximation:

$$\tilde{\pi}_{LA}(x_i|\boldsymbol{\theta}, \mathbf{y}) \propto \left. \frac{\pi(\mathbf{x}, \mathbf{y}), \boldsymbol{\theta}}{\tilde{\pi}_{GG}(\mathbf{x}_{-i}|x_i, \boldsymbol{\theta}, \mathbf{y})} \right|_{\mathbf{x}_{-i}=\mathbf{x}^*{}_{-i}(x_i,\boldsymbol{\theta})},$$

  where $\tilde{\pi}_{GG}(\mathbf{x}_{-i}|x_i, \boldsymbol{\theta}, \mathbf{y})$ is the Gaussian approximation of $\pi(\mathbf{x}_{-i}|x_i, \boldsymbol{\theta}, \mathbf{y})$ which is different than $\tilde{\pi}_G$, but computed in the same way, centered around the mode $\mathbf{x}^*{}_{-i}(x_i, \boldsymbol{\theta})$. This expression is far too expensive because computing $\tilde{\pi}_{GG}$ requires n factorizations of the precision matrix, one for each x_i. Two tricks are implemented to reduce the complexity of this approximation, and they consist in avoiding the computation of $\tilde{\pi}_{GG}$ and of its modal configuration $\mathbf{x}^*{}_i$. Instead, the mode is approximated using the already computed $\tilde{\pi}_G$, i.e.:

$$\mathbf{x}^*{}_{-i}(x_i, \boldsymbol{\theta}) \approx \mathbb{E}_{\tilde{\pi}_G}(\mathbf{x}_{-i}|x_i),$$

  where the right end side comes from $\tilde{\pi}_G$. The second idea is to consider only those x_j that are close to x_i since they should be the ones affecting the marginals of x_i. With these two tricks, computing $\tilde{\pi}_{LA}$ is much less expensive (see Section 3.2.2 of Rue et al. (2009) for further specifications).

- **Simplified Laplace approximations**: further improvement of the Laplace approximation. The key idea is to do a series expansion of $\tilde{\pi}_{LA}$ centered around its mean, which leads to the density $\tilde{\pi}_{SLA}$; the improvements are purely computational, and the authors suggest that they compensate for the slight decrease in the accuracy.

> The details that can be studied in Section 3.2.3 of Rue et al. (2009) are complicated and beyond our scope.

The last step of the algorithm is to calculate the marginal likelihood of the model using a method called the Akaike Information Criterion (AIC) (see Akaike (1974)). It compares the current data fitting and the fit of a simpler model that assumes independence between the observed variables.

2.3. Packages and issues

Hitherto, we discussed the key ideas of the strategy for approximating the posteriors of $\mathbf{x}$, which is a vector that represents all the parameters of a general additive latent Gaussian model. As previously stated, the algorithm is implemented by the *R-INLA* package. In particular, the software deals with all the computations presented in the previous paragraphs, which remain in some way under the hood. The package provides the help functions needed to build and run the model. Upon a correct setup, the user will get the model posterior results once the software has completed the data fitting. In Chapter 3, we will show some of the code tokens needed for the Bayesian analysis of a dataset. This way, we provide some indications about the practical usage of the package.

Before proceeding, we specify that to analyze the data, we started using the plain version of the package `inla`. We later moved on to the specific version created for spatio-temporal datasets, called `inlabru`. This version (see Bachl et al. (2019)) is nothing but a wrapper of the original package `inla`. However, `inlabru` simplifies the analysis by providing specific functions for spatio-temporal data. The code presented in Chapter 3 works with this package.

By analyzing the air pollution dataset (see Chapter 3, we could appreciate the strengths and weaknesses of this inference method. In particular, from the beginning, we could notice a higher computational speed, especially compared to standard MCMC techniques. However, the problem with memory became insurmountable when attempting to incorporate additional data by considering longer time and more detailed meshes for computing the GRMF (as described in Section 1.3.1).

The entire analysis and approximation of the posterior are carried out through matrix inversions, specifically of the precision matrix for computing Gaussian approximations of the densities. Of course, when refining the mesh with more points, where we approximate the Gaussian spatial residual, and when considering months or even years of daily data in the autoregressive model, the precision matrices become huge. Consequently, the computation of the matrix inverse requires a longer time and more space in memory. Initially, our analysis was made on a Personal Computer equipped with 8.00 GB of RAM, which quickly reached its capacity and caused the program to crash. To overcome this challenge, we sought assistance from the interdepartmental lab of Politecnico di Milano, *DataCloud*. From then on, the computations have been executed thanks to the availability of the DataCloud computational infrastructure built with funds by Politecnico di Milano. Further information about DataCloud is available on the website `https://datacloud.polimi.it/`. In particular, we acquired access to a virtual machine (VM) with significantly bigger memory and more *cpus* to parallelize computations and expedite the overall process. More specifically, the VM we used possessed the following characteristics:

- vCPU: 24
- RAM: 64 Gib
- HDD: 500 GiB
- OS: ubuntu 22.04

With the help of the Virtual Machine, we could rapidly analyze more data. In addition to this powerful computational resource, we installed the `pardiso` library, which speeds up the inference by efficiently parallelizing the computations. The *R-INLA* package grants access to the library upon verifying the presence of a license provided by the `pardiso` developers. Further details and methods related to this approach are available in van Niekerk et al. (2021).

In the next Chapter, we present posterior inference for two air pollution datasets collected by ARPA Emilia-Romagna and ARPA Lombardia.

3 | Case study: PM10 in the Po valley

This chapter contains the main results of the Bayesian analysis of air pollution data collected in several monitoring stations managed by the Regional Agencies of Environmental Protection (ARPA). The original dataset includes observations from four regions in the Po Valley, namely Emilia-Romagna, Lombardia, Piemonte, and Veneto, recorded from 2014 to 2020; we focused just on the first two regions, Emilia-Romagna and Lombardia, and the year 2018. Among the many pollutants measured, we focus on daily PM10 concentrations (expressed in $\mu g/m^3$). For a detailed data description and exploration, we refer to Lonati & Riva (2021) and to Frigeri (2022).

The acronym PM10 stands for *Particulate Matter* with an approximated diameter of less than 10 μm. It indicates different pollutant elements, including dust, smoke, and droplets diluted in the atmosphere. PM10 (among others particulate matter, like PM2.5) can be very dangerous to the health of humans and other living beings since it can penetrate the respiratory system and cause harm to the whole organism (see for example Manisalidis et al. (2020)). From this perspective, it is important to study and monitor air pollution in specific regions, to assess risk potentials and help policymakers to take the necessary measures.

We divided the chapter into two main sections, one for the Emilia-Romagna dataset and the other for the Lombardia dataset. The main reason for this division is that in the case of Lombardia, in addition to air pollution observations, we have also integrated meteorological data, as we will see in Section 3.2.

We will briefly describe each regional dataset before discussing the models that have been developed to fit the data. In the case of the first implemented model, we also show some code, in particular the commands that allow performing data analysis in the `INLA` environment, in particular using the wrapper package `inlabru`, introduced in Section 2.3. Then we discuss the posterior inferences and future predictions of "new" data. Finally, at the end of the chapter, we draw some conclusions. In particular, we will comment on the

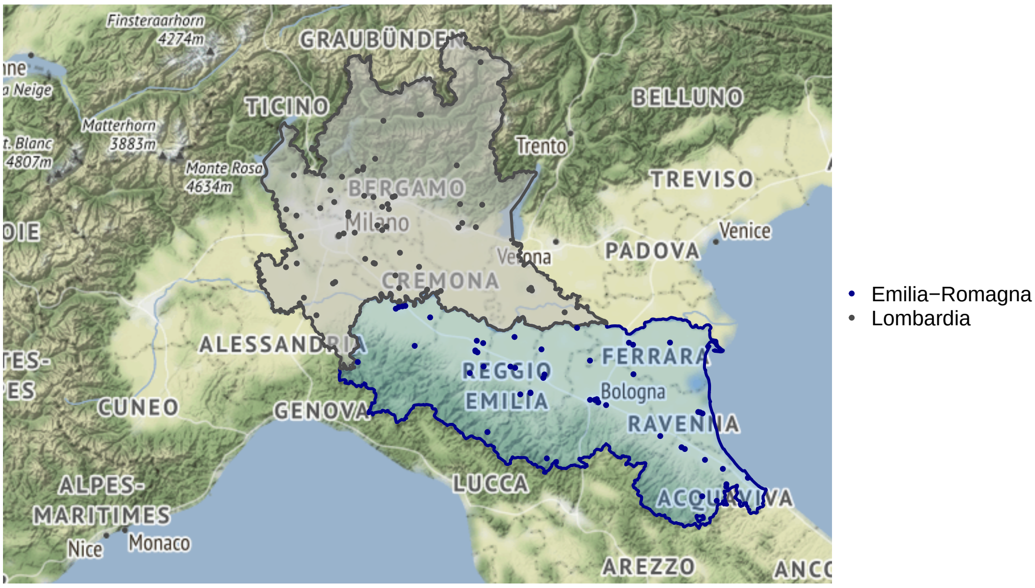


Figure 3.1: Maps of Emilia-Romagna (highlighted in blue) and Lombardia (in gray). Colored points denote the sites of the monitoring stations.

specific methods and tools used in our work, namely the SPDE approach and the `INLA` method.

3.1. Posterior Analysis of the Emilia-Romagna dataset

This section focuses on the analysis of data collected in Emilia-Lombardia, a region situated in the southeastern part of the Po Valley, as we can see in Figure 3.1.

Figure 3.1 also shows the locations of the 49 measurement stations that collected the data presented in Section 3.1.1.

3.1.1. Exploratory Data Analysis

We proceed now to analyze the dataset, referring to Lonati & Riva (2021) and Frigeri (2022) for complete exploration. Observations are collected in 49 stations across Emilia-Romagna, each identified with a name and a unique ID. The measurements consist of daily concentrations of PM10, collected each day of the year 2018. Besides the level of pollution, further information on the location of the stations is encoded in the dataset, namely:

- **Altitude** above sea level: measured in meters
- **Area**: A tag describing the area in which the stations are located; this categorical

variable can assume three levels: *Rural*, *Suburban*, *Urban*.

- **Type**: the station locations can be of type *Background*, *Industrial*, or *Traffic*; this categorical variable relates the station with the human activity present around it.
- **Zoning**: this variable groups stations in different areas, defined accordingly to geographical characteristics. The categories are specific for each region, and for Emilia-Romagna, they are *Agglomerato*, *Appennino*, *Pianura Est* and *Pianura Ovest*, identifying a different geographical zone of the region.
- **Geographical Coordinates**: the station location, measured originally in the lat-long system; for purposes that will be clear later, we have transformed these coordinates and expressed them in the UTM system.

The descriptive variables above, which contain some characteristics of the stations, can be used when building a regression model to fit the data. On the other hand, the coordinates are fundamental to constructing a spatio-temporal model, as in Section 1.2.1.

As mentioned before, we briefly explore the data, using mainly graphical tools, to get an idea of the temporal evolution of the pollution concentration and the differences that emerge for the various stations.

Figure3.2 shows trajectories in time of daily PM10 in 2018, for all the stations. Different shades of blue identify the evolution of PM10 collected in different stations, while the global day-by-day mean is depicted in dark red. General characteristics of the data emerge from Figure 3.2. For example, the temporal trend of the concentration of pollutants tends to be higher in mean and variability in the winter season (the two sides of the plot); it tends to decrease in the summer period (central area). If we look at all available data from 2014 to 2020, the oscillatory behavior of PM10 concentration would be even more recognizable (see Figure 1.2 of Frigeri (2022)). One explanation for this behavior could be that, during the colder seasons, the use of energy (mainly gas) to heat houses and other facilities increases, leading to more pollutants emission. Moreover, if we look at the mean, depicted in dark red in Figure 3.2, we can identify a typical behavior for the evolution of the pollutant in the air: the vast range of values it covers, characterized by steep peaks and valleys, especially pronounced during colder seasons.

In Figure 3.3, we plotted the same observations aggregated by the categorical variables **Type** and **Area** to highlight the variations among values collected at different Stations. In particular, in the first row, we grouped the observations by **Type**, dividing stations among *Background*, *Industrial*, and *Traffic*; in the second row, we grouped by the level of the categorical variable **Area**, separating among *Rural*, *Suburban* and *Urban*. Figure

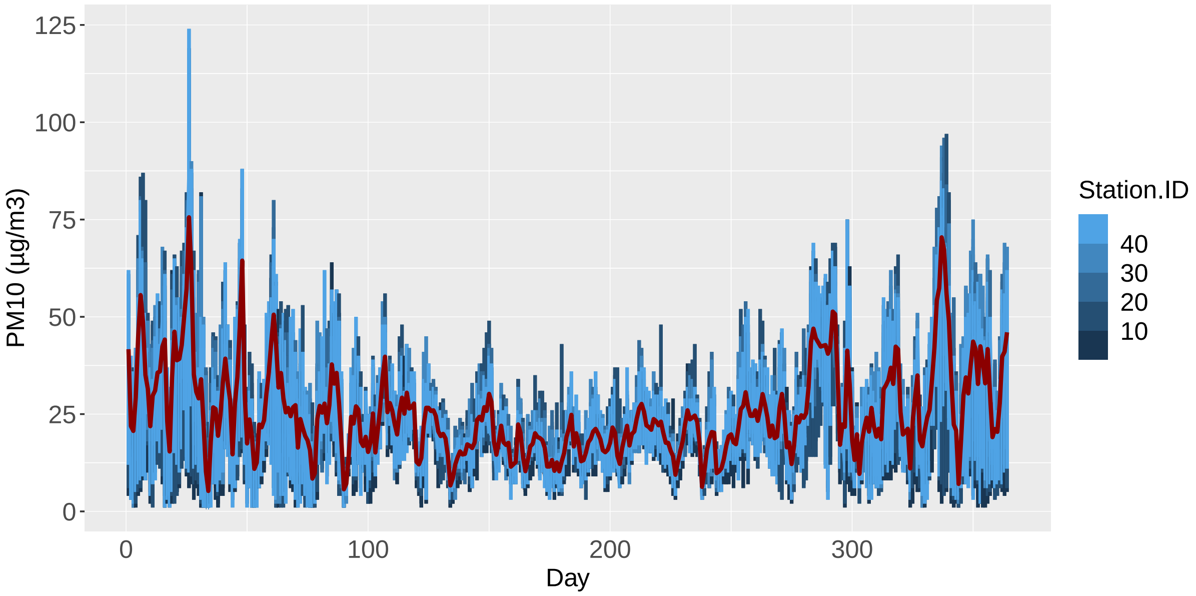


Figure 3.2: PM10 values in 2018 in Emilia-Romagna. Each station is depicted in a different shade of blue, and the superimposed dark-red line represents the daily average.

3.3 highlights different aspects of the data; in particular, we can see from the horizontal dark-red line, which represents the group mean of PM10 concentrations, that stations that are located in Industrial and Traffic sites tend to observe higher levels of pollution in the air. On the other hand, stations in rural areas experience a lower level of PM10 concerning urban and suburban areas.

This preliminary data analysis suggested we include in the regression model covariates representing this categorical variable since they can be used to explain the response variable, as we will see later in Sections 3.1.2 and 3.1.3.

The focus so far has been on the daily concentration of PM10; however, in the practical analysis we carry out in Section 3.1.2 and 3.1.3, we transformed the data by applying the logarithm. The log-transformation is a commonly used practice in air pollution and atmospheric studies (see Cameletti et al. (2013)) to make them more tractable. Moreover, it allows us to transform strictly positive data into observations with a more symmetric distribution, reducing variability and avoiding heavy distributional tails. These consequences of the transformation can be crucial in terms of computational efficiency and accuracy when performing posterior inference.

In the left panel of Figure 3.4 we plotted the daily series of the logarithm of PM10, to show the differences with respect to 3.2. The first thing we notice is the reduction in the range of the possible values: in particular, we moved from a range of roughly 125 μm

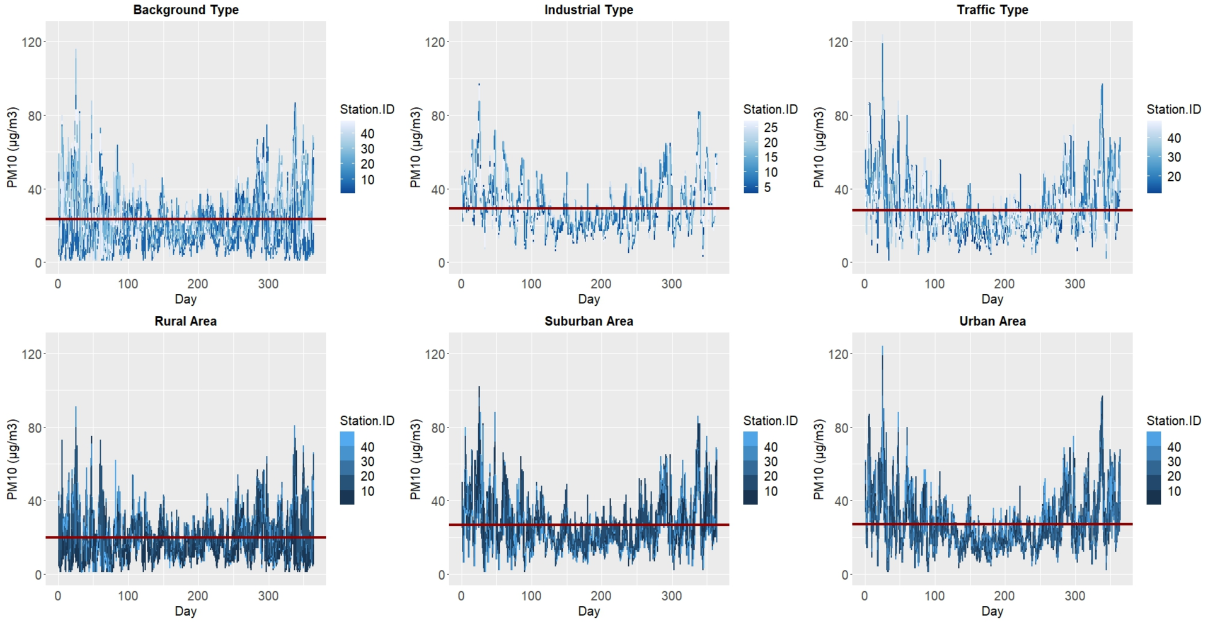


Figure 3.3: PM_{10} values grouped by different categories. First row: data divided by **Type**. Second row: data divided by **Area**. For each panel, the dark-red line represents the group mean.

to a range of 5 μm. In the second panel, we plotted the same data, after being scaled, following the classical formula

$$x_s = \frac{x - \overline{x}}{s}, \tag{3.1}$$

where x is the observation (in our case logarithm of the PM10 concentration), $\overline{x}$ the sample mean, s the sample standard deviation, and x_s the scaled value.

3.1.2. Bayesian inference with Model 1 in Emilia-Romagna

In this section, we present the first analysis of air pollution data in Emilia-Romagna using the R-package INLA. We will show how to build the regression model and how to run it with the help of some code tokens needed to use `inlabru`. The full R-script can be found in appendix A.1. Then we will analyze some results obtained, focusing on the posterior densities estimations and the prediction of future data. For the other models, presented in Sections 3.1.3, 3.2.2 and 3.2.3, we will not report specific code tokens, but the scripts can all be found in the Appendix, respectively A.2, A.3, A.4.

The first step in building the regression model is to choose the specific response variable, which in this case is the scaled log-transformation of PM10, observed at 49 different stations., Then, to implement the model in the `inlabru` approach, we need to specify

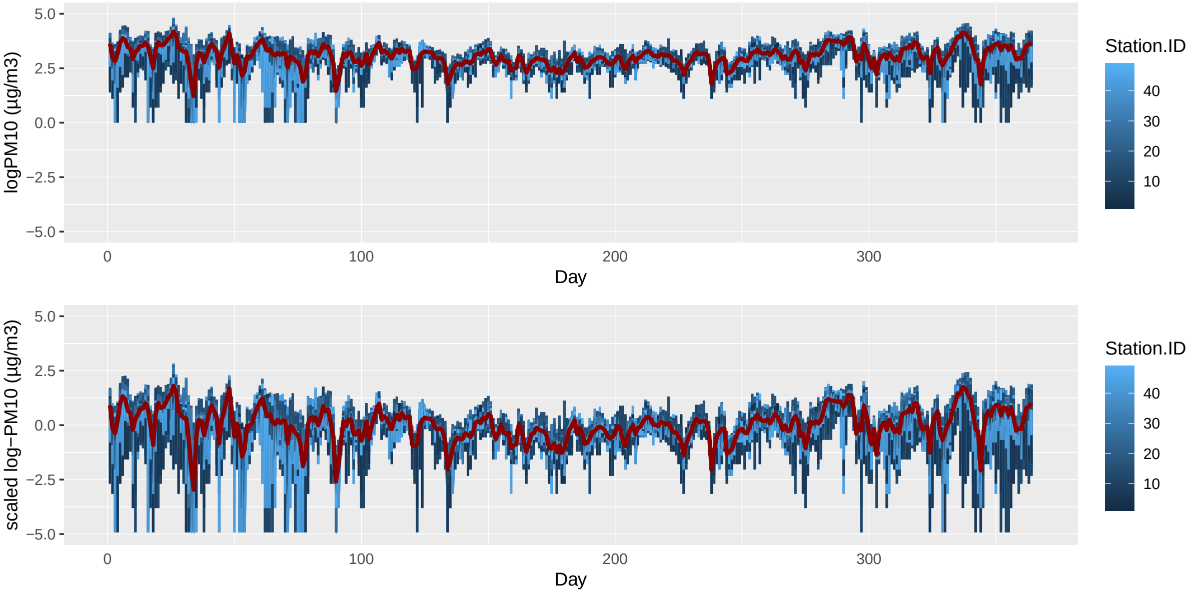


Figure 3.4: Top panel: time series of log-transformed PM10 concentration. Bottom panel: scaled log-transformed PM10 observations, centered at zero and with unit variance. Each station is identified by a different shade of blue, and the superimposed dark-red lines are the respective daily means.

the different additive components, formulate the likelihood structure, and finally fit the model.

Before presenting the necessary code, we recall that:

- the coordinate system is UTM; in this case, distances computed are expressed in meters.
- After preliminary analysis, we kept the relevant covariates for the regression, i.e. those with coefficients significatively different than zero a posteriori. We present in this section the resulting model of various attempts. In particular, we selected **Altitude** and **Type**. The first regressor has been standardized and centered, while the second has been used to define a dummy variable that associates 1 with a station of **Type** *Traffic* and 0 for other types.
- The response variable the spatio-temporal residuals in the SPDE approach (see Section 1.3) are assumed to have a zero-mean Gaussian distribution the spatio-temporal residuals in the SPDE approach (see Section 1.3).
- For the parameters' priors, we did not find any reason to change the default ones defined by INLA for computational reasons, except for the parameters of the SPDE model, namely the GMRF, as we will see later. Default priors and hyperparameters

can be found in Section 1.3.4, while the complete report can be read in the package's documentation, available at `https://www.r-inla.org/documentation`.

Following the notation of (1.15), the regression model can be then written as follows:

$$y(\mathbf{s}_i, t) = \beta_0 + \beta_1 z_1(\mathbf{s}_i) + \beta_2 z_2(\mathbf{s}_i) + \xi(\mathbf{s}_i, t) + \varepsilon(\mathbf{s}_i, t), \tag{3.2}$$

$$\varepsilon(\mathbf{s}_i, t) \overset{iid}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2), \qquad i = 1, ..., 49, \quad t = 1, ..., 90, \tag{3.3}$$

where the response variable is the scaled log-transformed concentration of PM10 at station i, which is located at $\mathbf{s}_i$, expressed in *UTMX* and *UTMY* coordinates. The random linear regression coefficients $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)'$ multiply the intercept, $z_1(\mathbf{s}_i)$, (scaled Altitude of the station i), and $z_2(\mathbf{s}_i)$ (dummy variable identifying those stations that are of type *Traffic*), respectively. We assume the spatio-temporal field denoted by $\xi(\cdot)$ has a Matérn spatial covariance structure and autoregressive dynamics of order 3 for the temporal part. This process has then five unknown parameters, namely, $(a1, a2, a3)'$ for the AR(3) and $(\sigma_\omega, k)'$ for the Matérn field. However, in the following paragraphs, we will not estimate directly k, but instead the effective range ρ; the two quantities are empirically related by (1.9).

We are now ready to discuss how to run such a model in the `INLA` framework using the package `inlabru`. Among the various components of the additive regression model, we start with the spatio-temporal residual. As presented in Section 1.3, the goal is to approximate the assumed underlying Gaussian process with a discrete field, the GMRF, following the SPDE approach. In this perspective, the first thing to do is to define a mesh, i.e. a set of points that maps the region of interests, to estimate the GMRF on those points. The standard procedure is to start from the coordinates of the stations and create the mesh with a particular domain triangulation, called *Delunay* triangulation (for further details, see Besag (1974)). This task is achieved with the help of an `INLA` function. Once the mesh is defined, we need an `spde` object, as we can see in the next lines of code:

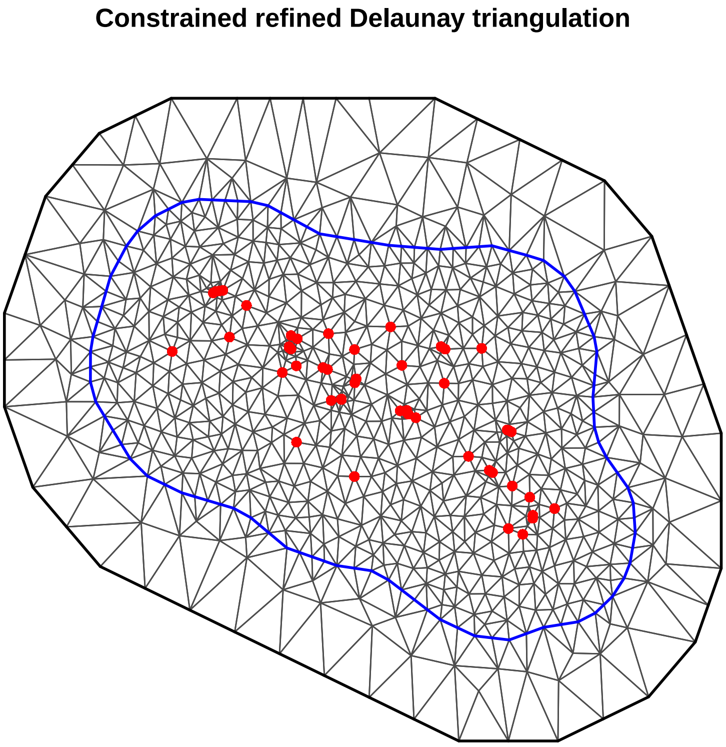
```
coo <- cbind(emilia$UTMX, emilia$UTMY)
bnd <- inla.nonconvex.hull(st_coordinates(map)[,1:2])

mesh <- inla.mesh.2d(loc = coo, boundary = bnd,
                     max.edge = c(16000,50000),cutoff = 100)

spde <- inla.spde2.pcmatern(mesh = mesh, alpha = 2,
                            prior.range = c(10000,0.01),
                            prior.sigma = c(3,0.01))
```

The mesh building function takes arguments through which we choose the desired characteristics of the triangulation, such as starting coordinates and boundaries. We use parameters like `maxedge` and `cutoff` to control the geometric features of the resulting triangulation. The denser the mesh, the more accurate the estimations and the higher the computational cost. Finally, when defining a `spde` object, we can assign parameters to the PC prior using `prior.range` and `prior.sigma`, in the sense of the statements (1.52) and (1.51), respectively.

The code above produces the mesh presented in Figure 3.5.



Constrained refined Delaunay triangulation

Figure 3.5: Triangulation of the domain containing Emilia-Romagna, with 856 vertices. In red the starting coordinates of the observation sites, and in blue the *non-convex hull* around the region's border.

At this point, we need to define all the components in the additive model in (3.2) and specify the likelihood form we assume for the data. There are specific functions implemented in the package that helps us:

```
cmp <- logPM10 ~ Intercept(1) +
        SPDE(coordinates, model = spde,
             group = time,ngroup = 100,
             control.group = list(model = "ar", order = 3)) +
        Altitude + Traffic

lik <- inlabru::like(formula = logPM10 ~ Intercept + SPDE +
            Altitude + Traffic,
            family = "gaussian",
            data = data)
```

The first command creates a textitformula that specifies the model components and uses compatible syntax for the `inlabru` internal structure. The `SPDE` component contains the characteristics of the spatio-temporal field, which is an autoregressive process of order 3 with the spatial structure defined by the `spde` object. We set `n.group = 100` to consider 90 days and additional 10 days for prediction. The second command defines the likelihood of the log-transformed response variable as Gaussian.

Finally, we are ready to run the whole model using the command `bru`, as we can see below:

```
    fit = bru(cmp, lik,
          options = list(
            verbose = TRUE,
            inla.mode = "experimental",
            control.inla = list(strategy = "simplified.laplace", int.strategy = "eb"),
            num.threads = "12:2"
          ))
```

This function, which calls the original `inla` function, is the main command of the package. It sets up the environment and analyzes the data based on the built and fed model. It produces a `bru` object that contains all algorithm results, including posterior marginals of defined parameters. The function offers many options that can be found in the package documentation. We highlight two options: the strategy is set to `simplified.laplace` through `control.inla` and `num.threads` enables the efficient use of a multi-core virtual machine. The first number of `num.threads` should be double the number of parameters, and the second should be 1 or 2

When the computations started by the `bru` command are over, we can analyze the results, as we will see in the next section.

Posterior Analysis

We now focus on the marginal posterior densities of the parameters $\boldsymbol{\beta}$, σ_ε, $a1$, $a2$, $a3$, σ_ω and ρ. Those densities can be easily extracted from the `bru` object with the help of specific functions, as we can see from the code below:

```
sig.eps.marg <- inla.tmarginal(function(x) (1/sqrt(x)),
                                    fit$marginals.hyperpar$‘Precision for the
     Gaussian observations‘)

a1.temporal.marg <- inla.tmarginal(function(x) x,
                                        fit$marginals.hyperpar$‘Group PACF1
    for SPDE‘)
a2.temporal.marg <- inla.tmarginal(function(x) x,
                                        fit$marginals.hyperpar$‘Group PACF2
    for SPDE‘)
a3.temporal.marg <- inla.tmarginal(function(x) x,
                                        fit$marginals.hyperpar$‘Group PACF3
    for SPDE‘)

field.residuals <- inla.spde.result(fit, name = "SPDE", spde = spde, do.
    tranf = T )
spde.sig.w.marg <- sqrt(field.residuals$marginals.variance.nominal[[1]])
spde.range.marg <- field.residuals$marginals.range.nominal[[1]]
fixed.marg <- fit$marginals.fixed
```

The function `inla.spde.result` is specifically built to extract information about the spatio-temporal GMRF estimated during the model's run. Moreover, there is a particular element in the `bru` object that collects the marginals of the linear effects, called *fixed effects*, namely the coefficients $\boldsymbol{\beta}$, related to the defined covariate.

Figure 3.6 shows marginal posterior densities of the regression coefficients $\boldsymbol{\beta}$ corresponding to the fixed effects, namely the *Intercept*, *Altitude* and the *Traffic*. As we expected a priori, the intercept's coefficient is centered at 0, which reflects the scaling pre-processing of the log-transformed concentration of PM10. Moreover, we can see a negative impact of the altitude on the expected level of pollution, while stations in traffic areas tend to have a higher concentration of PM10. This plot not only tells us that the position of the monitoring station influences the observed pollution but also that the covariates included in the model are significant. This aspect can also be seen in Figure 3.7, where we plotted the credibility intervals centered around the posterior mean for the $\boldsymbol{\beta}$ coefficients.

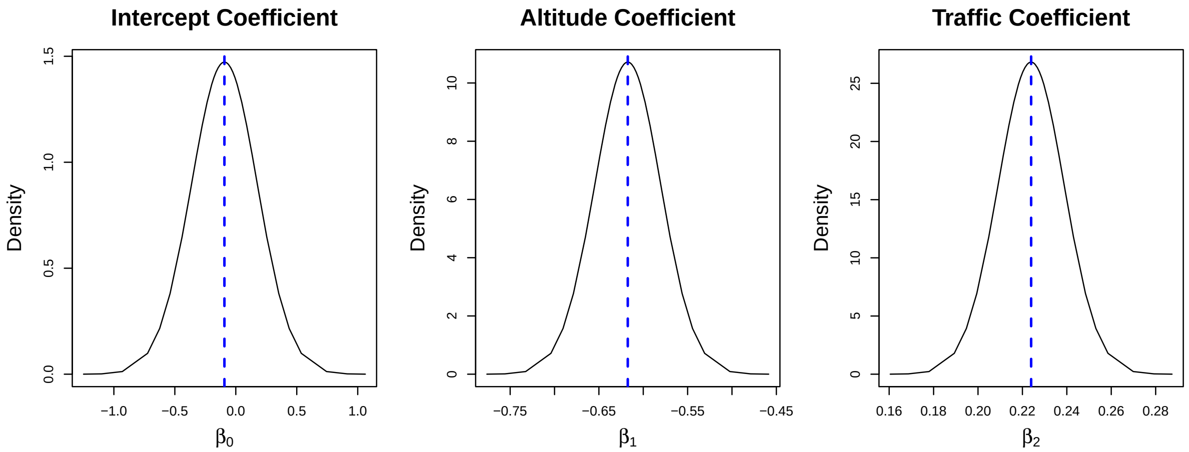


Figure 3.6: Marginal posterior densities for the $\boldsymbol{\beta}$ coefficients. The dashed blue lines represent the posterior mean of the densities.

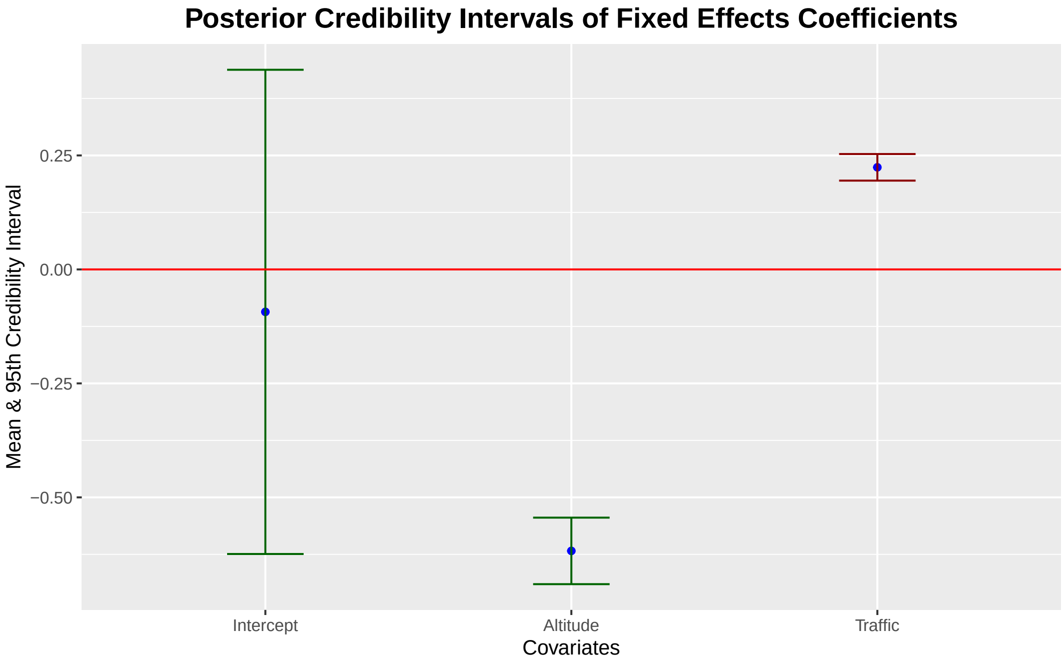


Figure 3.7: Posterior 95% credibility interval for the regression parameters.

This plot shows that the regression parameters corresponding to both *Altitude* and *Traffic* are relevant in explaining the response variable since their posterior 95% credibility intervals do not contain the value 0.

Figure 3.8 shows the marginal posterior densities of the standard deviation σ_ε of the Gaussian error term in (3.2) and of the autoregressive parameters a_1, a_2, a_3. As we can see from the plot, all the posterior marginal distributions of the three autoregressive parameters are relevant since their mass is concentrated on values different than 0. Choosing an AR(3) was a trade-off between model accuracy and computational costs. Deeper autoregressive dynamics theoretically improve accuracy, but estimating more parameters drastically increases computational costs. Additional temporal components did not improve estimation and only increased the time and memory needed to fit the data.

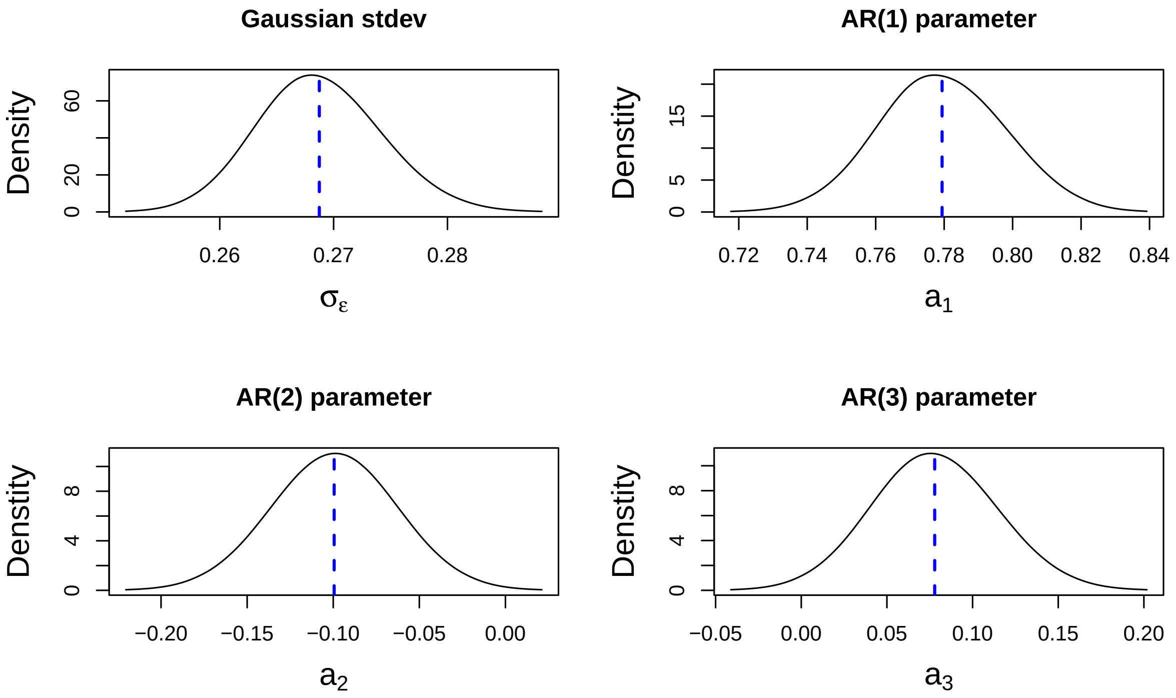


Figure 3.8: Marginal posterior densities of σ_ε, a_1, a_2 and a_3.

Figure 3.9 shows the marginal posterior distributions for the SPDE model, namely the spatial parameters. On the left panel, we see the distribution of the standard deviation of the Matérn field σ_ω (see (1.17) and (1.19)) and, on the right, the effective range ρ, expressed in meters. By choosing *UTMX* and *UTMY*, we could get an interpretable range estimation since distances in this system are computed in meters. Parameter ρ indicates the distance, in meters, beyond which the spatial correlation becomes negligible, as already discussed in Section 1.1.1. In this case, the posterior distribution is centered around 180 km. This result indicates a strong spatial correlation among pollution data, which decreases slowly with the distance.

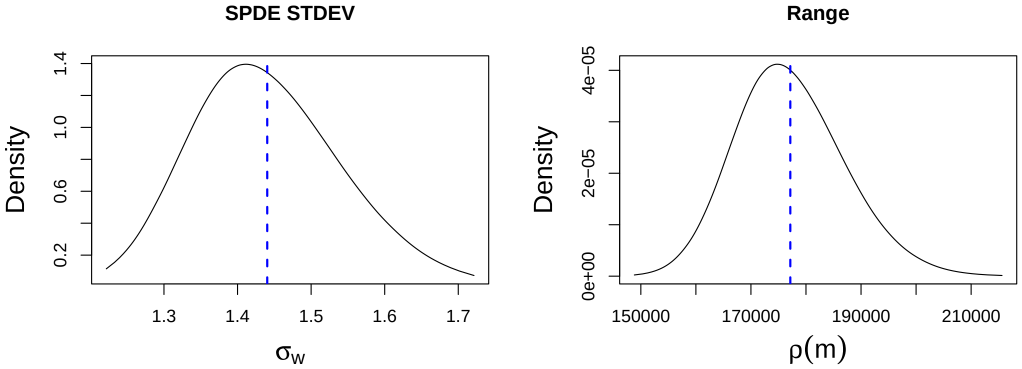


Figure 3.9: Marginal posterior densities of the SPDE parameters, standard deviation σ_ω (left panel) and range ρ (right panel).

We recap all the summary statistics of the estimated marginal posterior distributions for all the parameters of the model in Table 3.1

Parameter	Mean	SD	q_0.025	q_0.5	q_0.975
σ_ε	0.27	0.01	0.26	0.27	0.28
$Intercept$	-0.09	0.27	-0.62	-0.09	0.44
$Altitude$	-0.62	0.04	-0.69	-0.62	-0.54
$Traffic$	0.22	0.01	0.19	0.22	0.25
a_1	0.78	0.02	0.74	0.78	0.82
a_2	-0.1	0.04	-0.17	-0.1	-0.03
a_3	0.08	0.04	0.01	0.08	0.15
$\rho\ (km)$	177.154	9.988	159.255	176.477	198.411
σ_ω	1.44	0.1	1.26	1.43	1.65

Table 3.1: Summary of the marginal posterior distributions of all the parameters in **Model 1** .

Prediction of future data

In this section, we see how to compute the prediction of future data; we include the corresponding code. In particular, we tried to predict the evolution of the logPM10 concentration for 10 days after the 90 fitted. The `inlabru` package makes available a simple function that performs this task, as we can see in the code below:

```
pred.badia = predict(fit, badia, ~ Intercept + SPDE + Altitude +
                                               Traffic,
                              n.samples = 1000)
```

The main argument of the function is the `bru` object `fit`, together with the data needed for the prediction, encoded in the *data-frame* `badia`. We decided to pick one (or more) stations and predict the evolution of the concentration of PM10. Then we pass to the function `predict` the regression formula and the number of samples we want it to simulate, to make a Monte Carlo mean estimation.

Figure 3.10 shows that our model manages to reconstruct the values of PM10 for the two chosen stations (*Badia*, a non-traffic station, and *Timavo*, a traffic station), but it cannot predict well the future values of the pollutant. The mean of the posterior predictive distribution of $y(\mathbf{s}_i, t+1), ..., y(\mathbf{s}_i, t+m)$, with $m = 10$ is very different from the true value observed and the uncertainty (represented by the grey area) explodes immediately after the border between present and future.

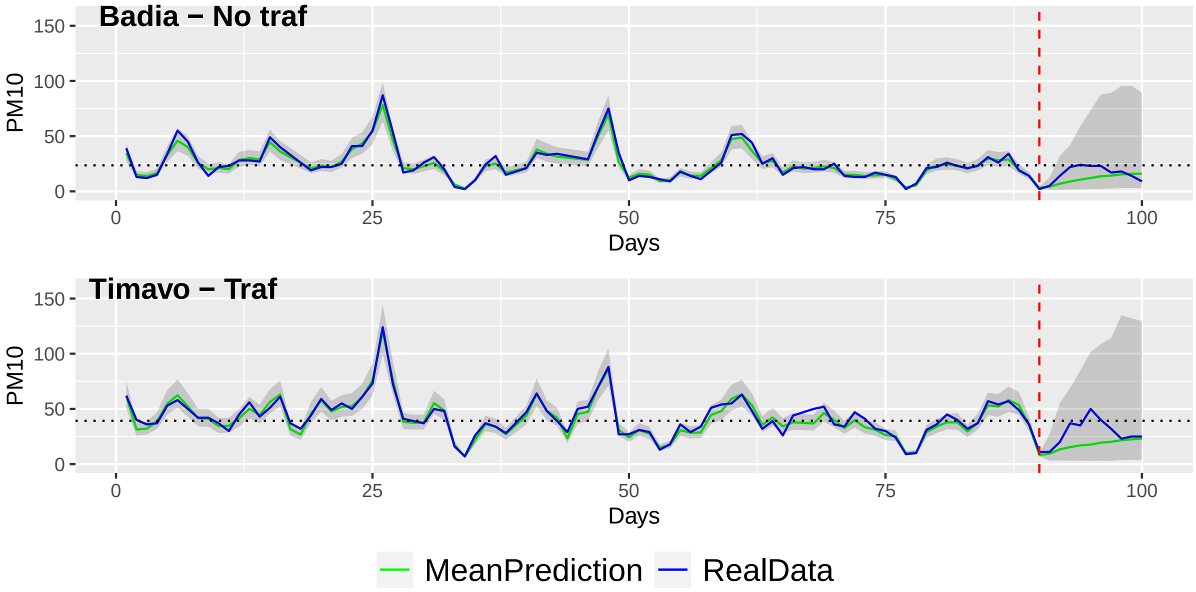


Figure 3.10: Time series of the true value of PM10 concentration (blue line) compared to the mean over 1000 samples of predicted value (green line). The grey area represents the posterior mean's 95% credibility interval. The red dashed line represents the border between past and future data (the 90^{th} day). The horizontal black dashed line is the overall mean of the pollution observed at the stations.

3.1.3. Bayesian inference with Model 2 in Emilia-Romagna

This section contains a slightly different model than the one presented in Section 3.1.2. In particular, we pre-processed the data by creating a new variable, **Diff**, which stores the sequential difference in log-transformed PM10 concentrations. We computed this variable by taking the log transformation of daily PM10 concentrations, calculating the difference between each day and the previous day for each station, and detrending the variable by subtracting the mean. The differencing was performed to make the time series stationary and remove its mean.

We can see in Figure 3.11 how this transformation changes the outlook of the data. The first effect is to make the observations more symmetric around the value 0, which is the base level. From the dark-red line, which represents the daily average, we can see that the consecutive daily variations of the concentration of the PM10 are wider for colder seasons, as we could also see from Figure 3.2. During the summer, pollution tends to be more stable, reflected by the lower positive and negative excursions in the central region of Figure 3.11.

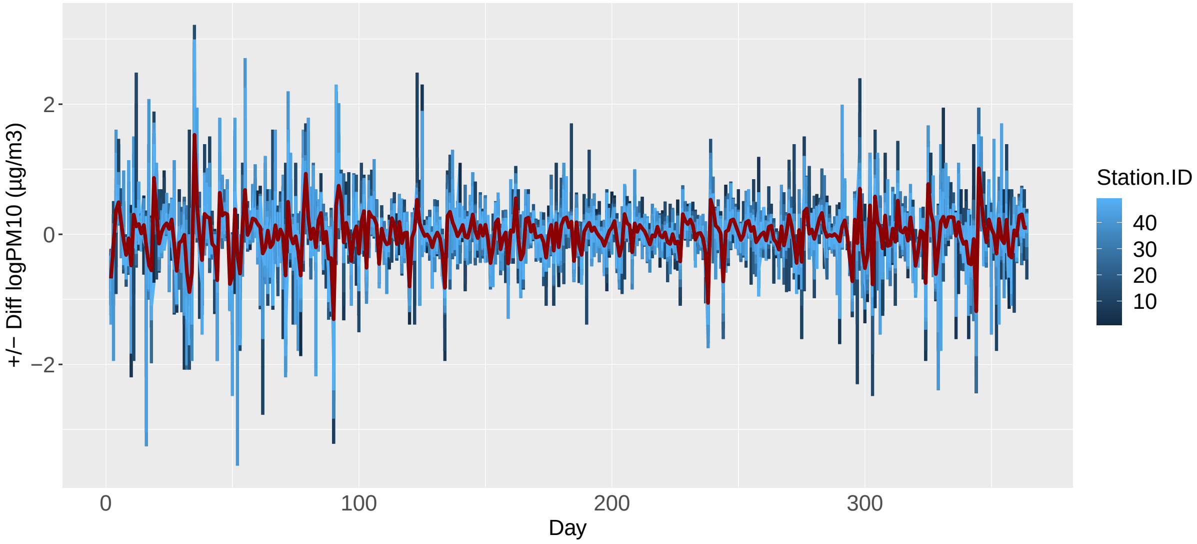


Figure 3.11: Daily values of logPM10 expressed in terms of consecutive differences. Each station is colored with a different shade of blue. The superimposed dark-red line represents the daily mean.

The goal of this section is not to repeat the model building and the code necessary to run it, but just to illustrate some interesting results that reflect the key differences that the dataset transformation yields. We apply the same model as in (3.2), but we consider as data the first 120 days of 2018, predicting 20 days in the future. We tried to fit more data as possible, and this number of days was the highest we could reach without the program crashing for using too much memory. The response variable is now the daily consecutive detrended difference of the log-transformed PM10. We also point out that we used the same mesh as in Section 3.1.2 since we had no incentive to change it: a more coarse mesh would make the model lose accuracy, while an even more refined one would cause a significant increment in computational costs while having no impact on the accuracy.

Posterior Analysis

We start by looking at the marginal posterior distributions of the parameters of the model, which are the same as the ones in the model presented in Section 3.1.2. Figure 3.12 shows the posterior mean and 95% credibility interval for the regressor coefficients, and the results are different than the ones obtained from the previous model. The Intercept coefficient is still centered around 0, which is expected since we detrended the consecutive differences dataset. However, non of the regressors is significant, since their 95% credibility interval contains the value 0, in contrast to the results of **Model 1**, reported in Figure 3.7. One possible explanation of this result is that, by studying this particular transformation of the data, we concentrate on the daily changes in the concentration of PM10 and not on the level itself (although log-transformed and scaled). In this scenario, the altitude and type of the Station (traffic or not) have no impact on the consecutive differences in the response variable. In fact, after seeing these results, we built another model without these regressors, and the results were practically the same.

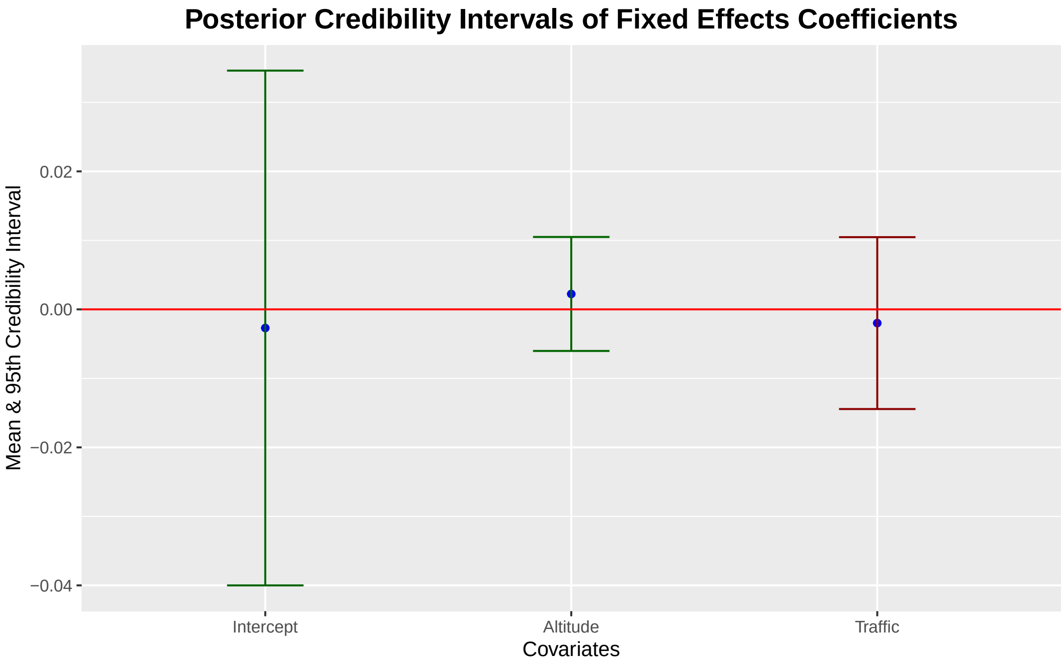


Figure 3.12: 95% posterior credibility intervals of regression parameters.

We can now turn our attention to the marginal posteriors of the other parameters of the model. From the top-left panel, we can see that the standard deviation of the Gaussian measurement error $\varepsilon(\mathbf{s}_i, t)$ defined in the same way as in (3.2), in concentrated around slightly higher values. This indicates more variability in the dataset, justified by the fact that this transformation only detrended the observations, not scaled them. The other figures in the plot depict the marginal posteriors of the autoregressive parameters a_1, a_2 and a_3, and they are much different than those shown in Figure 3.8. They are all significantly negative, suggesting a negative correlation between the transformed observation at day t and those at $t-1$, $t-2$, and $t-3$. This result is very interesting and, in our opinion, reflects the oscillating behavior of the consecutive differences dataset around the base level 0. The sign of these parameters suggests that if the concentration difference is positive on a certain day, it will probably become negative in later days (and the opposite too), reflecting the symmetry about level 0.

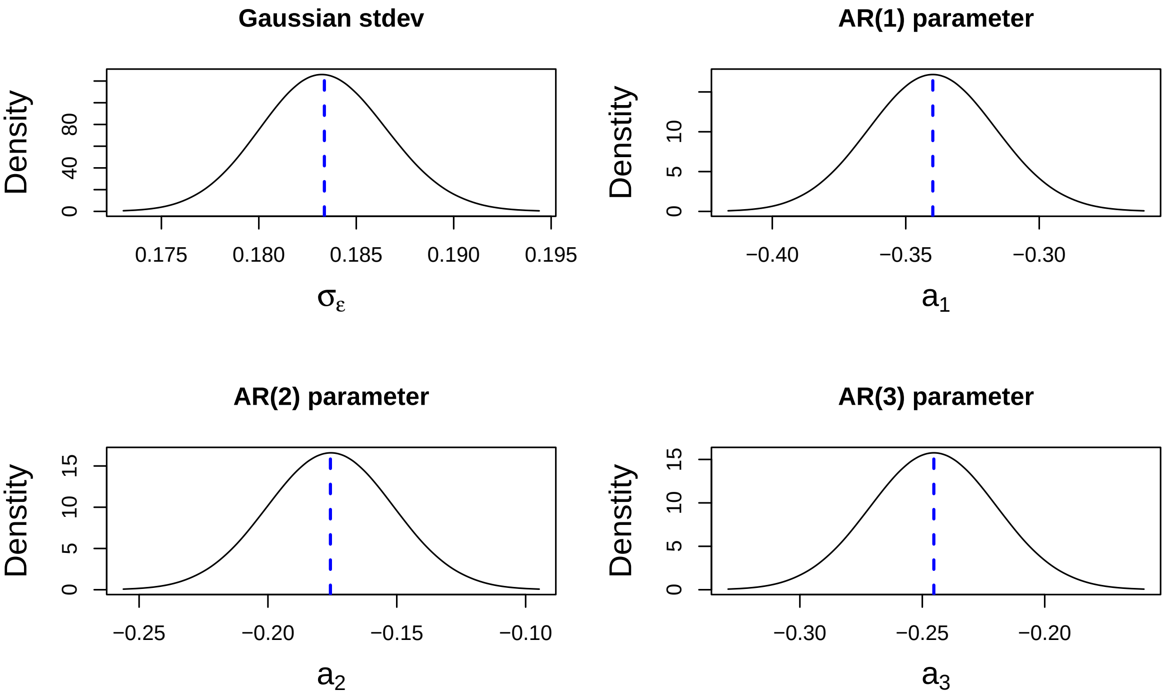


Figure 3.13: Marginal posterior distributions of parameters σ_ε, a_1, a_2 and a_3.

Figure 3.14 reports the marginal posteriors for the parameters σ_ω and ρ of the Matérn covariance structure modeling the spatial part of the GMRF. In this case, the difference between the two models is not remarkable. Indeed, as we can see by comparing these results to the one plotted in Figure 3.9, both the standard deviation of the spatial residual and the estimated effective range are more or less the same. We have a decrease in the posterior estimate of ρ, from about 180 km to 170 km, but it is not nearly enough to suggest a relevant change in the spatial field structure.

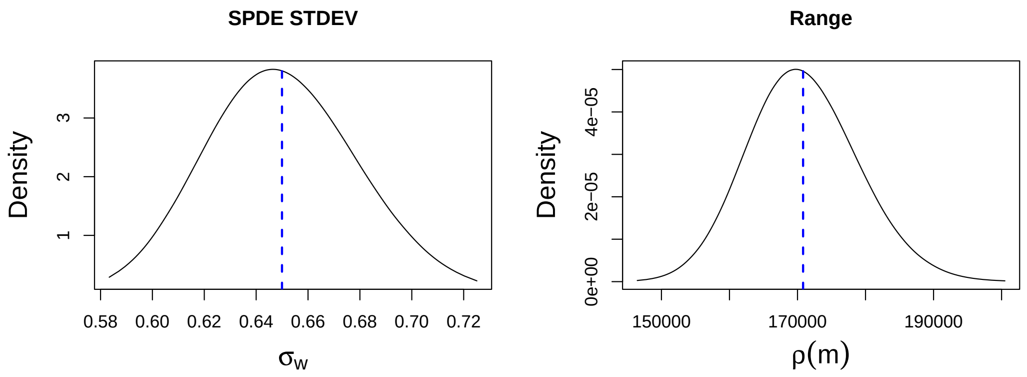


Figure 3.14: Marginal posterior distributions of the SPDE parameters, standard deviation σ_ω (left panel) and range ρ (right panel).

We also report summary statistics of the marginal posterior of the parameters in Table 3.2

Parameter	Mean	SD	q0.025	q0.5	q0.975
σ_ε	0.42	0.01	0.41	0.42	0.44
$Intercept$	-0.01	0.04	-0.09	-0.01	0.08
$Altitude$	0.01	0.01	-0.01	0.01	0.02
$Traffic$	0.01	0.01	-0.01	0.01	0.02
a_1	-0.34	0.02	-0.39	-0.34	-0.29
a_2	-0.18	0.02	-0.22	-0.18	-0.13
a_3	0.25	0.03	-0.3	-0.25	-0.19
ρ (km)	170.242	8.120	155.259	169.860	187.126
σ_ω	1.5	0.07	1.38	1.5	1.63

Table 3.2: Summary of the marginal posterior distributions for all the parameters in **Model 2**.

Prediction of future data

In this section, we finally present posterior predictive estimation of "future" data, which has been carried out with the same code presented in Section 3.1.2. As already mentioned, this model considers 120 days for the estimation and 20 days for future prediction. The results are plotted in Figure 3.15. We chose the same two stations of Section 3.1.2 to highlight the model differences.

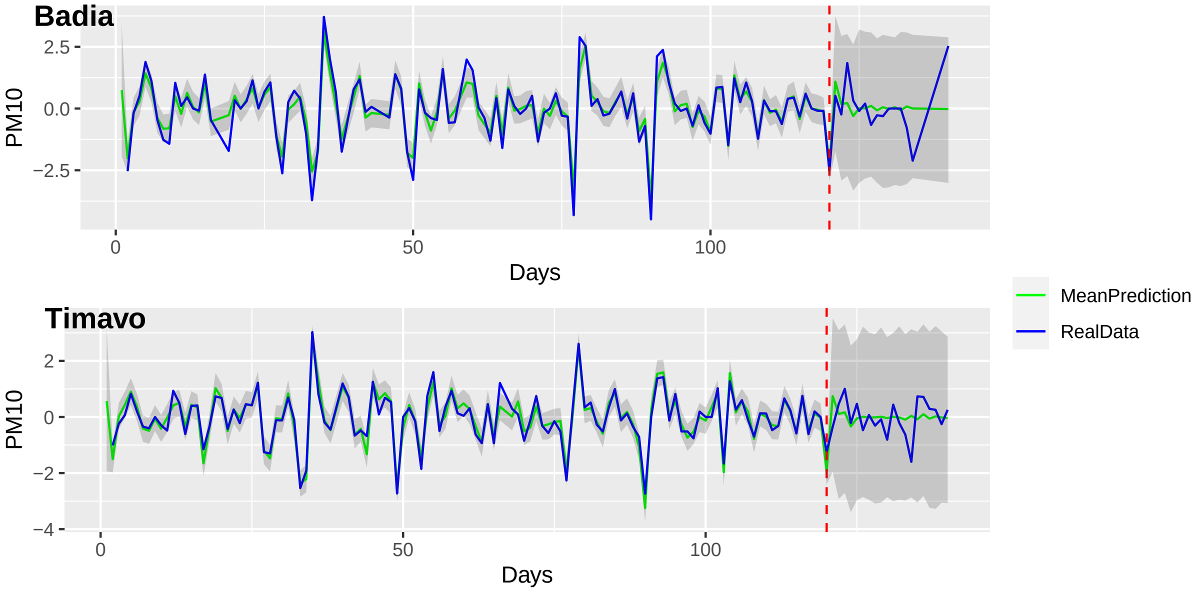


Figure 3.15: Time series of the detrended consecutive differences of logPM10 concentration (blue line) compared to the mean over 1000 samples of predicted value (green line). The grey area represents the posterior mean's 95% credibility interval. The red dashed lines represent the border between past and "future" data (the 120^{th} day).

From the figure, it is clear that this model can still reconstruct the response variable, though transformed, with a narrow credibility interval (the grey area around the green line). The problems again arise when trying to predict the future evolution of the pollutant concentration. However, after the 120^{th} day (right of the red dashed line), the prediction shows an interesting behavior where the mean attempts to follow the actual values but fails to predict the peaks and valleys of the time series as time progresses. This is due to the autoregressive component of the model relying on less accurate predictions and not on past data. Unlike Figure 3.10, the 95% credibility interval does not explode after the boundary between seen and unseen data in the future domain, but remains constant and contains the true value of the response variable.

3.2. Posterior analysis of the Lombardia dataset

In this section, we present posterior inference of two models similar to those in Sections 3.1.2 and 3.1.3, but performed on a different dataset collected in the northern region of Lombardia, highlighted in blue in Figure 3.16. The blue dots represent the locations of the ARPA measurement stations that collected the air pollution data. As mentioned in Section 3.1, different pollutants were observed, but we focused our analysis on the concentration of PM10, measured in $\mu g/m^3$.

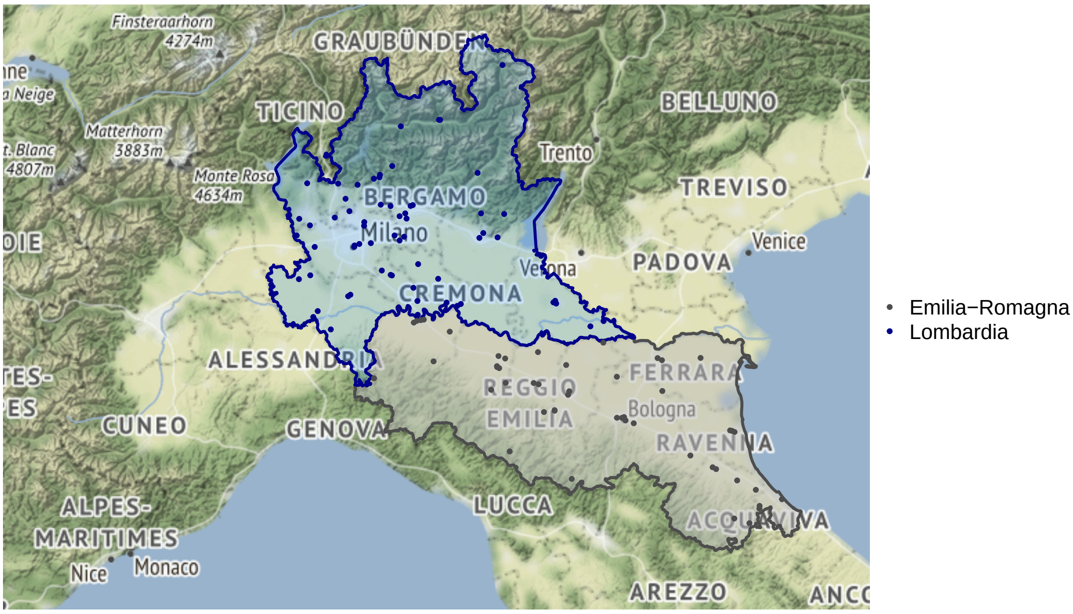


Figure 3.16: Maps of Lombardia (highlighted in blue) and Emilia-Romagna (in gray). Colored points denote the sites of the monitoring stations.

The dataset that we consider, as we will see later in Section 3.2.1, was built by merging two different datasets. One considers the daily values of PM10, as in the case of Emilia Romagna. We now have data from 64 monitoring stations located across Lombardia territory, with the same structure and descriptive variables discussed in Section 3.1.1. The second dataset contains information about the meteorological conditions; these additional variables represent the reason we carried out separate analyses for the two regions. It is well known that weather conditions can sensibly influence air pollution (see, for example, Grange et al. (2018)) and our goal was to use a Bayesian inference model and the INLA method to find statistical evidence to support this claim.

Before briefly exploring the final dataset in Section 3.2.1, we need to roughly describe how it was built and how the meteorological information was integrated into the original pollution dataset. This task was carried out by Ph.D. student Michela Frigeri, whom we thank for her help and contribution. First of all, the data about the weather come from 200 stations across the region, managed by the same agency responsible for the pollution data, namely ARPA, Lombardia.

The PM10 monitoring stations and the meteorological stations are situated at different locations. This problem is solved by linking each PM10 station with the nearest meteorological site. Moreover, as the meteo data were collected in different time intervals (every 10 minutes), we computed daily averages to create a consistent temporal scale. It is important to note that we addressed this issue of varying time intervals straightforwardly, but it may require more careful consideration. Anyway, now we have a complete dataset containing both descriptive covariates of the stations and daily meteorological data. In Section 3.2.1, we present in detail the new meteorological variables and a brief graphical exploration of the dataset.

3.2.1. Exploratory Data Analysis

As we explained in Section 3.2, we now work with a bigger dataset, that includes daily measures of different weather variables, in addition to the descriptive ones listed in Section 3.1.1. These new variables are:

- **TotalRain** $[mm]$ **:** total millimeters of rain fallen during the day;
- **RainIntensity** $[mm/h]$ **:** measure of the intensity of the rain fallen during the day. It is computed by dividing the total daily rain by the number of hours it has rained;
- **ConstRain** $[-]$ **:** categorical variable indicating whether it has rained at least 8 h (value 1) or less than 8 h (value 0);
- **AvgWind** $[m/s]$ **:** average daily velocity of the wind;
- **MaxWind** $[m/s]$ **:** peak of wind velocity recorded during the day;
- **QuietDay** $[-]$ **:** categorical variable indicating whether the wind was recorded for at least 8 h during the day (value 0) or for less than this time (value 1);
- **AvgRad** $[W/m^2]$ **:** daily average of sun radiation collected by the station;
- **AvgTemp** $[°C]$ **:** average temperature, recorded in Celsius;
- **MaxTemp** $[°C]$ **:** maximum temperature, recorded in Celsius;
- **AvgHumidity** $[\%]$ **:** daily average of humidity recorded at the station.

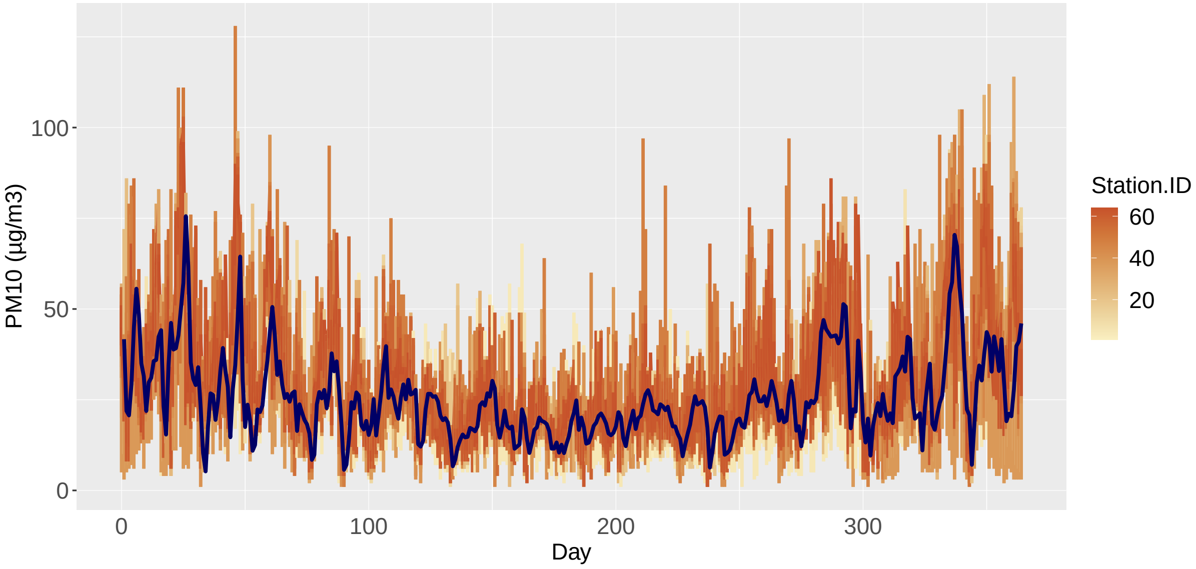


Figure 3.17: Temporal evolution of PM10 concentration in Lombardia. Each station is depicted with a different shade of orange. The superimposed dark-blue line represents the daily average.

We underline that all the continuous predictors presented above have been scaled by applying the usual formula (3.1).

We start by looking at the temporal evolution of the concentration of PM10 collected at different stations during the year 2018. In Figure 3.17, we can see the same patterns as in Figure 3.2 for the ER case, specifically the seasonal variation of the pollutant concentration. This shape is particularly evident in the daily mean, depicted in dark blue. The concentration of PM10 is lower during the summer, while it increases in the other seasons, giving the plot a typical U shape.

Figure 3.18 shows the transformed PM10 concentration values, computed in the same ways as discussed in section 3.1.3. The plots have similarities to those of the transformed data collected in Emilia-Romagna, as seen in Figures 3.4 and 3.11.

At this point, we would like to investigate using exploratory data analysis, the relations between the level of PM10 and other variables that characterize the measurement station or the meteorological information in the dataset.

We start with the relationship between the (standardized) **Altitude** variable and the pollutant concentration. As we have also seen for data collected in Emilia-Romagna, in Section 3.1.2, this variable is a relevant regressor for the response variable, i.e. the PM10 concentration. This characteristic is also evident from Figure 3.19. We divided stations

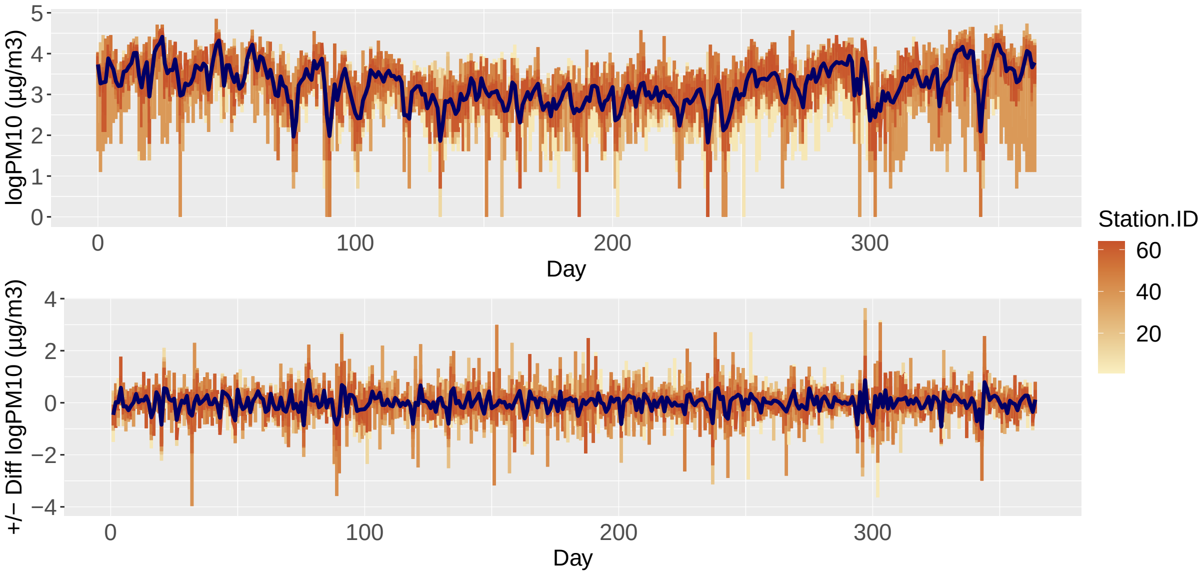


Figure 3.18: Top panel: log-transformed concentration of PM10. Bottom panel: daily consecutive differences, detrended. Each station corresponds to a shade of orange, while the dark-blue lines represent the daily mean.

into four altitude levels and found that higher altitude stations have lower logPM10 values (dark orange in the figure) while lower altitude stations have higher pollutant concentrations (yellow in the figure)

Reworded: Figure 3.20 indicates a subtle but noticeable differentiation between stations situated in high-traffic regions (depicted in orange) and those that are not (represented in teal). This observation will be substantiated by statistical proof in Section 3.2.2, similar to what has already been done with the Emilia-Romagna dataset (see Section 3.1.2).

The following paragraphs will focus on the meteorological data available in the dataset. As mentioned earlier in this section, these variables are recorded daily. Therefore, we have chosen to concentrate on one particular station and illustrate the temporal fluctuations of several variables.

Figure 3.21 displays two similar graphs, plotting logPM10 data collected in the city center of Brescia. The graphs are color-coded based on two factors: the daily average sun radiation (top) and the average temperature (bottom). We chose these factors to demonstrate the similarity in their behavior. It is worth noting that, during summer, longer exposure to the sun leads to higher temperatures and greater radiation levels. On the other hand, during colder seasons, we observe lower temperatures and less radiation. This observation becomes more interesting when considering the concentration of pollutants, which appears to be negatively correlated with both factors. Sections 3.2.2 and 3.2.3 present

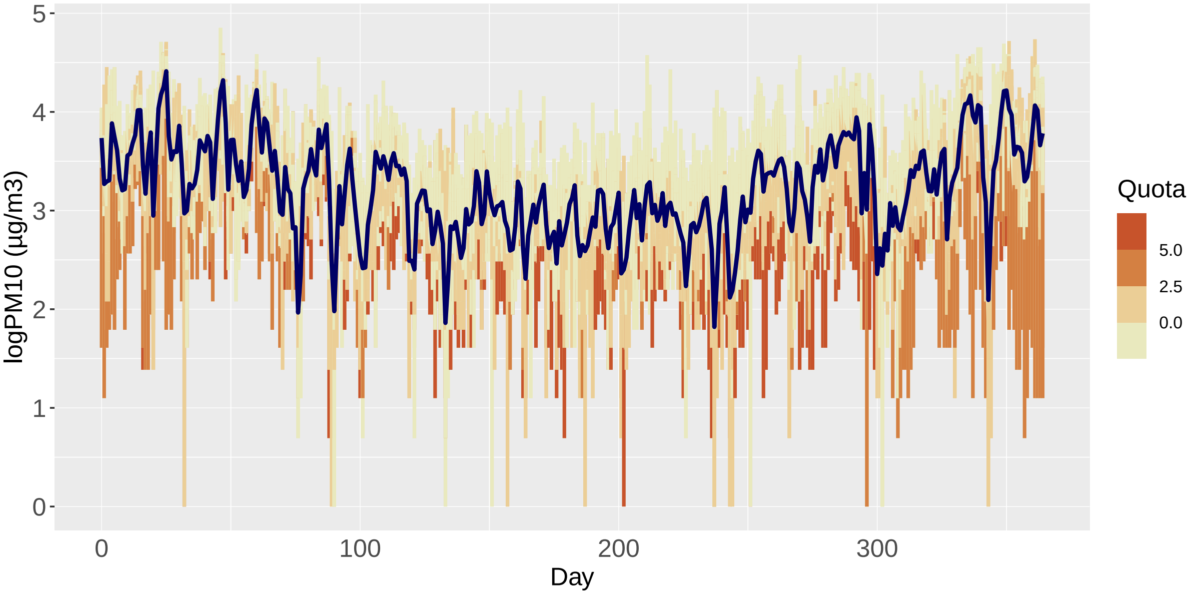


Figure 3.19: LogPM10 concentrations, grouped by values of covariate Altitude. Stations belonging to four different levels of **Altitude** are colored with different shades of orange.

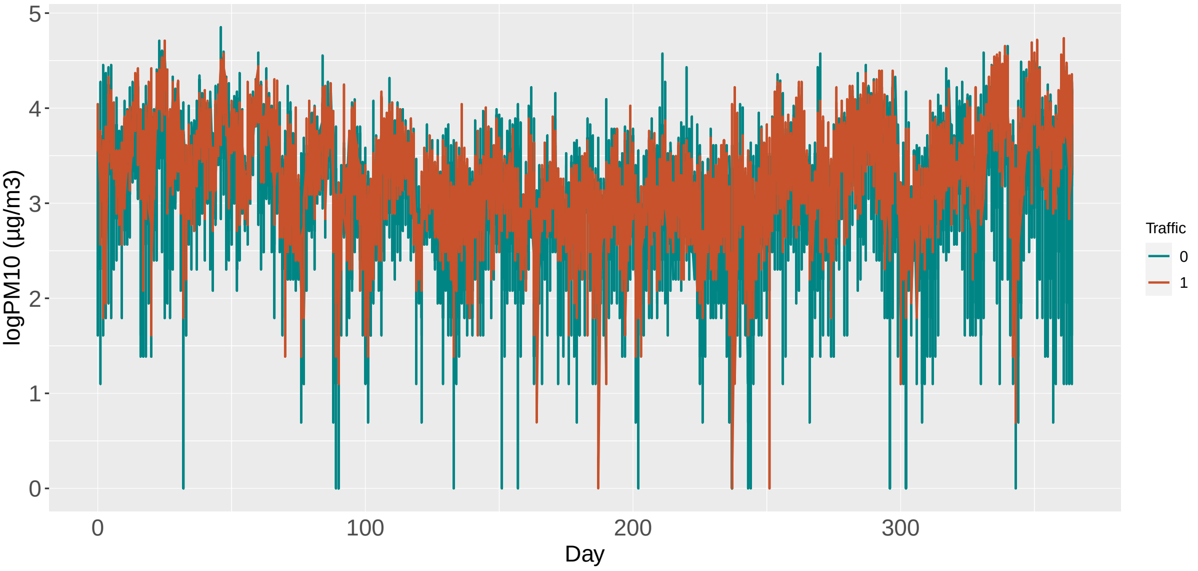


Figure 3.20: LogPM10 stations divided according to the label **Traffic**. In orange the locations with value one (trafficked), while in teal those with value zero (not trafficked).

statistical evidence to support this assertion.

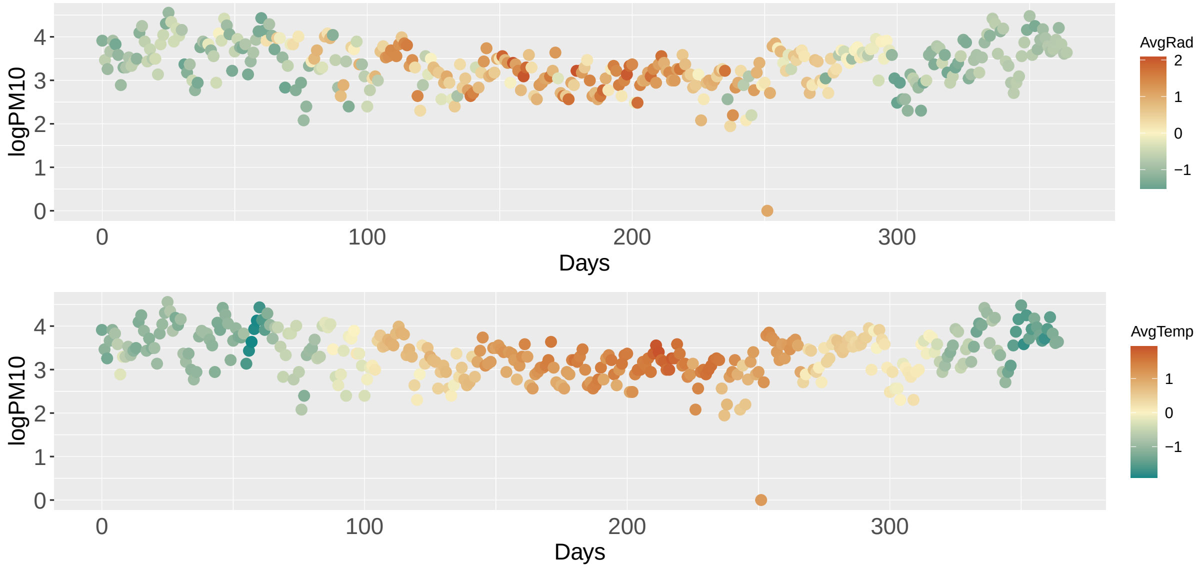


Figure 3.21: Daily trend of logPM10 collected in the station of Brescia. Top panel: colored days in concerning **AvgRad** variable; bottom panel concerning **AvgTemp** variable.

Finally, before presenting the last model used in the inference computation, we need to discuss and motivate the choices of meteorological variables to include. The first approach was to avoid possible collinearity and, therefore, to avoid repetitive variables. In this way, we decided not to include variables describing more or less the same event. For example, from **TotalRain**, **RainIntensity** and **ConstRain**, we only considered **TotalRain**, which proved to be more significant from an initial model exploration. In the same way, from **AvgWind**, **MaxWind** and **QuietDay**, we only chose **AvgWind**. In the next section, we present the final model.

3.2.2. Bayesian inference with Model 1 in Lombardia

The first Bayesian model applied to Lombardia data is similar to the model discussed in Section 3.1.2. The response variable, the standardized logarithm of PM10 concentration, is assumed to have a Gaussian distribution, and the spatio-temporal process underlying is a Gaussian field with a Matèrn covariance structure for the spatial component and an autoregressive process of order one for the temporal part. Moreover, beyond **Altitude** and **Traffic**, we added the meteorological information in the vector of covariates.

The vector of covariates, after the selection based upon exploratory data analysis and other preliminary considerations (see the end of Section 3.2.1), is composed of the following variables:

- constant level, identified by the **Intercept**,
- $z_1(\mathbf{s}_i)$, denoting the **Altitude** of the station,
- $z_2(\mathbf{s}_i)$, denoting whether the station is of type **Traffic** or not,
- $z_3(\mathbf{s}_i, t)$, denoting the daily **AvgRad**,
- $z_4(\mathbf{s}_i, t)$, denoting the daily **AvgTemp**,
- $z_5(\mathbf{s}_i, t)$, denoting the daily **AvgWind**,
- $z_6(\mathbf{s}_i, t)$, denoting the daily **TotalRain**.

It is evident from the list that the meteorological variables ($z_3,\ z_4,\ z_5,\ z_6$) are dependent on time, while the ones describing the stations ($z_1,\ z_2$) are not. This property of the covariates raised a question during the model development process. Specifically, whether the relative coefficients should vary with time or if a conventional fixed-effects structure could be applied. However, introducing time-dependent coefficients results in an exponential increase in the model's size when dealing with long time series. Furthermore, we tested this configuration considering observations collected during a few days and found the outcomes discouraging. Therefore, we decided to maintain the coefficient constant over time, as depicted in the model equation 3.4.

To sum up, the whole additional model is the following:

$$\begin{aligned}\{y(\mathbf{s}_i,t) = \beta_0 + \beta_1 z_1(\mathbf{s}_i) &+ \beta_2 z_2(\mathbf{s}_i) + \beta_3 z_3(\mathbf{s}_i,t) + \\ +\beta_4 z_4(\mathbf{s}_i,t) + \beta_5 z_5(\mathbf{s}_i,t) &+ \beta_6 z_6(\mathbf{s}_i,t) + \xi(\mathbf{s}_i,t) + \varepsilon(\mathbf{s}_i,t)\}, \\ \varepsilon(\mathbf{s}_i,t) \overset{iid}{\sim} \mathcal{N}(0,\sigma^2_\varepsilon), &\quad i = 1,...,64,\ t = 1,...,90,\end{aligned} \tag{3.4}$$

where the regression coefficients β_3, β_4, β_5 and β_6 are associated to the variables $z_3(\mathbf{s}_i, t)$, $z_4(\mathbf{s}_i, t)$, $z_5(\mathbf{s}_i, t)$ and $z_6(\mathbf{s}_i, t)$. Finally, we point out that in this case, the stations are 64, and that we consider 90 days of observations for the estimation of marginal posterior distributions, and additional 10 days for the prediction, as in model (3.2). To estimate the spatial GP through the GMRF in the SPDE approach (see 1.3), we built a mesh containing 806 locations, depicted in Figure 3.22, with the same `inla` functions as presented in Section 3.1.2.

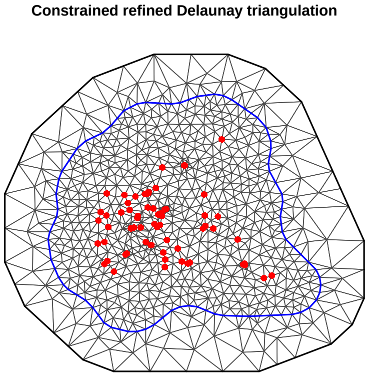


Figure 3.22: Triangulation of the domain containing Lombardia, with a total of 806 vertices. In red the starting coordinates of the observation sites and in blue the *non-convex hull* around the border.

Posterior Analysis

We start analyzing the results from the posterior distributions of parameters related to the spatio-temporal part of the model, defined as a separable stochastic process, composed by an autoregressive process of order 1 with parameter a_1, and a Matérn-like Gaussian field for the spatial process, parametrized by the standard deviation σ_ω and by the range ρ, the latter expressed in meters.

In Figure 3.23 we plotted the estimated posterior distributions of parameters σ_ε and a_1. In particular, if we look at the distribution in the right panel, we can see how it is concentrated around values close to 1.

In Figure 3.24, we have plotted the estimated marginal posterior of parameters σ_ω and ρ. The parameter in the left panel is the standard deviation of the covariance structure defined in (1.7) while the one in the right panel is the range ρ, computed empirically as in (1.9). Both distributions are slightly different than the ones in the model for Emilia-Romagna presented in Section 3.1.2 and plotted in Figure 3.13. In particular, both parameters have distributions concentrated around higher values, namely 1.7 for σ_ω and 240 for ρ. One possible explanation lies in the dataset's characteristics: Lombardia is a larger region, and data were observed in more stations than in Emilia-Romagna. This may have caused a greater estimated variance for the Gaussian spatial residual, as well as a wider range. Nonetheless, both parameters are within the same order of magnitude, implying that the two distinct models are consistent.

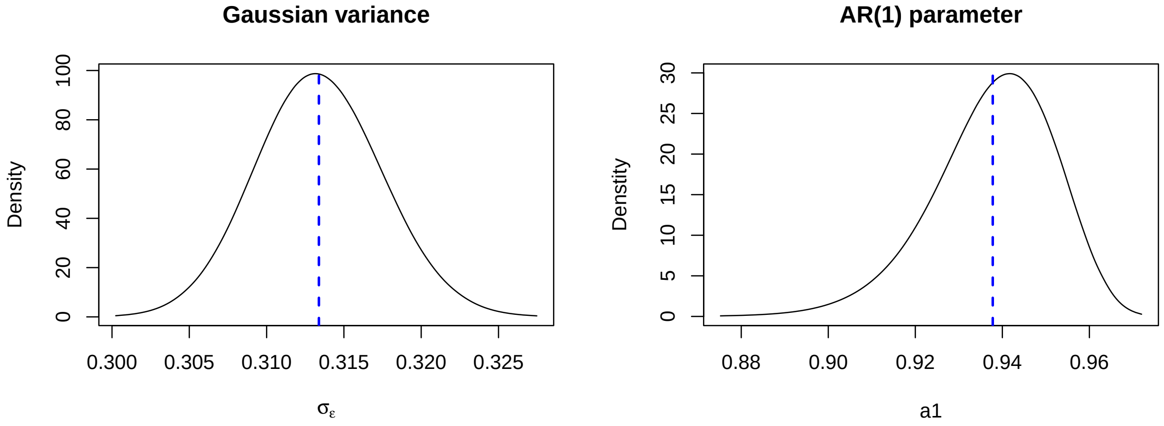


Figure 3.23: Marginal posterior densities of the standard deviation of the Gaussian error term σ_ε (Right panel) and of the parameter of the AR(1) process, a_1 (right panel).

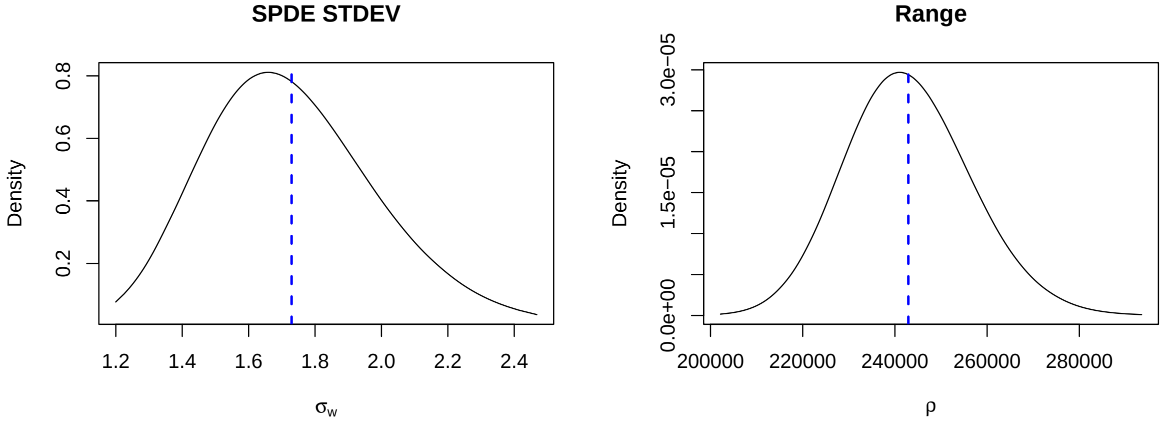


Figure 3.24: Marginal posterior densities of the standard deviation of the covariance structure σ_ω (left panel) and of the range ρ, expressed in meters (right panel).

We can now focus on the additional variables of the model, namely meteorological data. The goal is to verify if some weather variables are relevant for the regression. We plotted the obtained results in Figure 3.25.

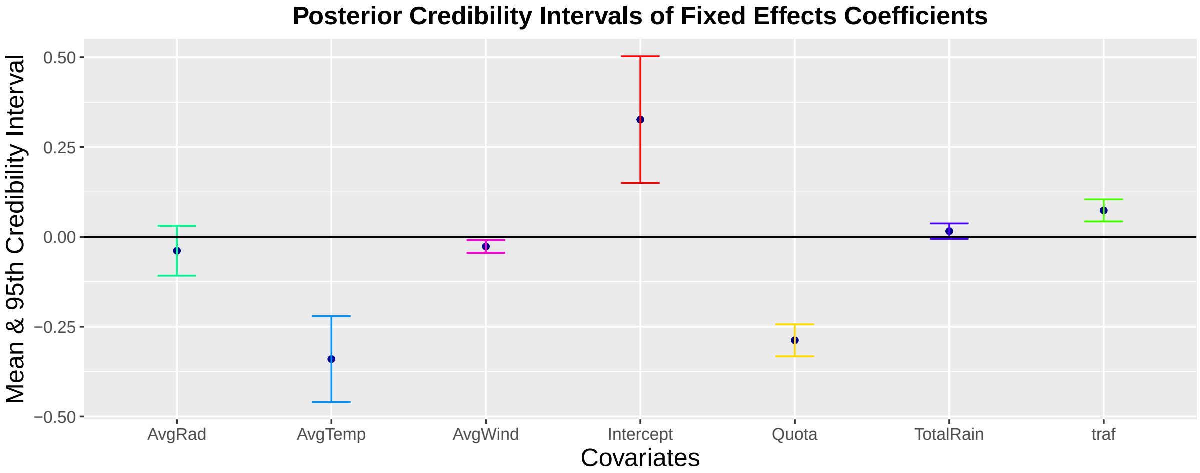


Figure 3.25: 95% credibility intervals of regression parameters.

As we can see in Figure 3.25, the variables **Altitude** and **Traffic** remain relevant, with a negative and positive impact on the concentration of pollution, respectively. Moreover, some meteorological variables that we included in the model are relevant, and in the way that we expected. Both the **AvgTemp** and **AvgWind** variables have negative coefficients, which provides reassurance for distinct reasons. The negative correlation between logPM10 and temperature is evident from Figure 3.21 and is supported by empirical data indicating lower pollution concentrations during summer months. Conversely, the negative coefficient for wind reflects common-sense notions: increased wind should promote cleaner air in areas with high emissions, as pollutants can be dispersed. While this view may be challenged based on factors such as wind direction and local pollution densities, the available information supports this result as reasonable.

Moreover, as shown in Figure 3.25, the coefficient for **TotalRain** appears to have a slightly positive impact on the concentration of PM10. Although the result is not entirely significant, since the value 0 falls within the 95% credibility interval, this finding seems reasonable. During colder and wetter seasons, such as winter, air pollutants tend to have higher concentrations than during hotter and drier seasons, like summer. Finally, we want to look at the coefficient related to the **AvgRad**: based on the information presented in Figure 3.21, it may be reasonable to assume that this variable has a similar behavior to **AvgTemp**. However, it appears that **AvgRad** is irrelevant in explaining the response variable since it is almost identical to **AvgTemp**. Therefore, one of the variables is redundant and should be removed from the model. Further details will be discussed in Section 3.2.3.

Finally, we observe an unusual behavior of the intercept coefficient, which is estimated to be significantly positive, even though the data have been standardized and the logPM10 values have a sample mean equal to 0. To address this issue, we tried to manually remove the intercept from the model. However, the software automatically included a coefficient for the constant level, leading to even poorer results. We hypothesize that adding more covariates and components resulted in identifiability problems, despite the other posterior estimates appearing reasonable.

We sum up all the estimated marginal posterior distributions in the following Table 3.3:

Parameter	Mean	SD	q0.025	q0.5	q0.975
σ_ε	0.31	0.004	0.31	0.31	0.32
$Intercept$	0.33	0.09	0.15	0.33	0.50
$Altitude$	-0.29	0.023	-0.33	-0.29	-0.24
$Traffic$	0.074	0.016	0.043	0.07	0.10
$AvgRad$	0.04	0.04	-0.11	-0.04	0.03
$AvgTemp$	-0.34	0.06	-0.46	-0.34	-0.22
$TotalRain$	0.02	0.01	-0.01	0.02	0.04
$AvgWind$	-0.027	0.01	-0.04	-0.027	-0.01
ρ (km)	242.92	13.7	217.53	242.33	271.32
σ_ω	1.73	0.25	1.30	1.71	2.28
a_1	0.94	0.01	0.91	0.94	0.96

Table 3.3: Summary of the marginal posterior distributions for all the parameters in **Model 1**.

Prediction of future data

This section contains the results obtained when predicting "future" data. Precisely, we used data from the first 90 days to estimate the posterior distributions and additional 10 days to make predictions. In Figure 3.26, we plotted the results for two stations that we picked, one representing stations in the trafficked area (Brescia) and one not (Bormio).

From Figure 3.26, we can see the same results presented in Figure 3.10, relative to the model analyzed in Section 3.1.2. The model can reconstruct the evolution of the PM10 concentration until the time border between past and future, while it cannot forecast unseen data. The outcome was somewhat disappointing as we believed that by using meteorological data, we would be able to enhance our ability to predict pollutant levels more effectively.

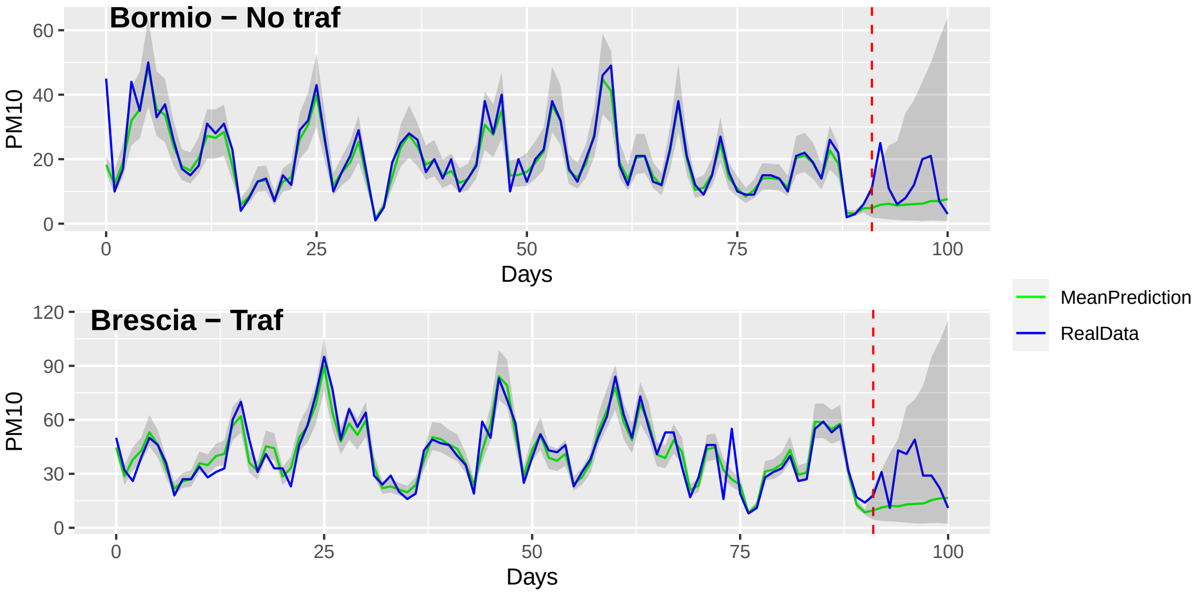


Figure 3.26: Time series of the true value of PM10 concentration (blue line) compared to the mean over 1000 samples of predicted value (green line). The grey area represents the posterior mean's 95% credibility interval. The red dashed line represents the border between past and "future" data (the 90^{th} day).

3.2.3. Bayesian inference with Model 2 in Lombardia

This section presents the final model developed to fit the data collected in Lombardia, described in Section 3.2.1. The general goal was to create a model that could analyze a wide range of data points across time to improve the accuracy of future data predictions. From previous models, the most effective one for this task has been the model discussed in Section 3.1.3, so we decided to build a similar one to improve the outcomes. In our opinion, its success can be attributed to two key factors: the differencing of the response variable and the inclusion of an autoregressive process of order three. In that case, though, we were unwilling to compromise on spatial accuracy by building a coarser mesh, to include more days in the analysis. However, for this model, we ultimately decided to sacrifice some spatial accuracy to extend the model in the temporal dimension and include more months.

Following several attempts to find a trade-off between temporal and spatial accuracy, we ultimately decided to include a six-month dataset and use a coarser mesh comprised of 478 vertices. Figure 3.3 provides a comparison between the two meshes, with the figure on the left representing Model 1 (see Section 3.2.2) and the figure on the right showcasing the coarser mesh used in the model presented in this section.

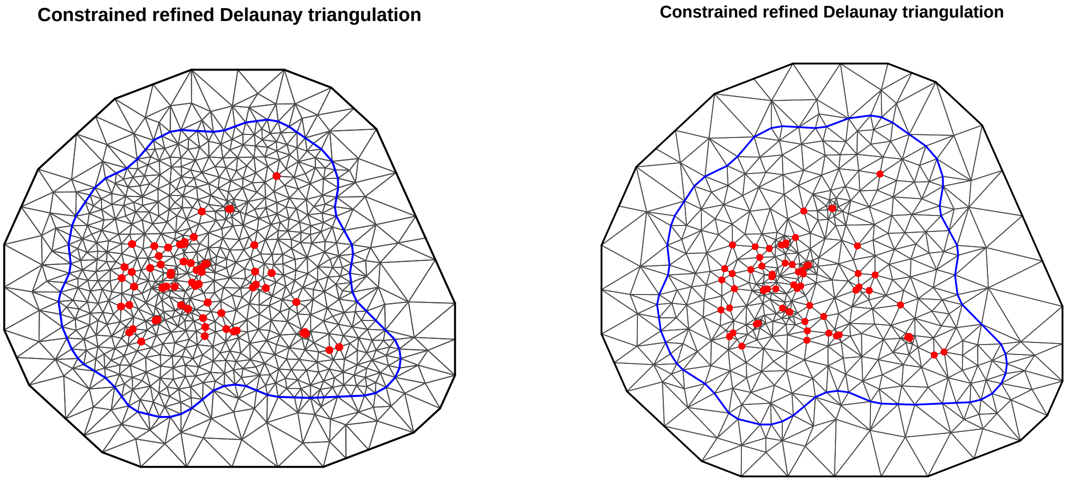

Constrained refined Delaunay triangulation
Constrained refined Delaunay triangulation

Figure 3.27: Left panel: mesh built for Model 1, 806 vertices. Right panel: a coarser mesh for Model 2, 478 vertices. In both cases, the red dots represent the monitoring stations and the blue line is the *non-convex hull* around the border.

To sum up, we built an additive regression model (as usual), with a spatio-temporal part composed of a Matérn-like Gaussian process and a temporal autoregressive process of order 3, to make better predictions. As regressors, we kept the same ones of Model 1, presented in Section 3.2.2, except for the variable **AvgRad**, which we found to be not significant (see Figure 3.25). The model equation is then almost identical to the one in (3.4), with the novelties introduced above. Moreover, as mentioned in the previous paragraph, we extended the days taken into consideration: in this case, $t = 1, ..., 180$, with additional 20 days for future prediction. In this context, it's worth mentioning that the significant reduction in the number of vertices in the mesh raises concerns about the accuracy of our results. To determine whether the achieved accuracy is sufficient, we'll examine our ability to explain the response variable and compare the results obtained for the spatial residual. Towards the end of this chapter, we will also compare the two estimations of the GMRF used to approximate the Gaussian Process that underlies the spatial distribution of PM10 concentration.

Posterior Analysis

We start presenting the results by looking at the marginal posterior densities of the parameters included in the model.

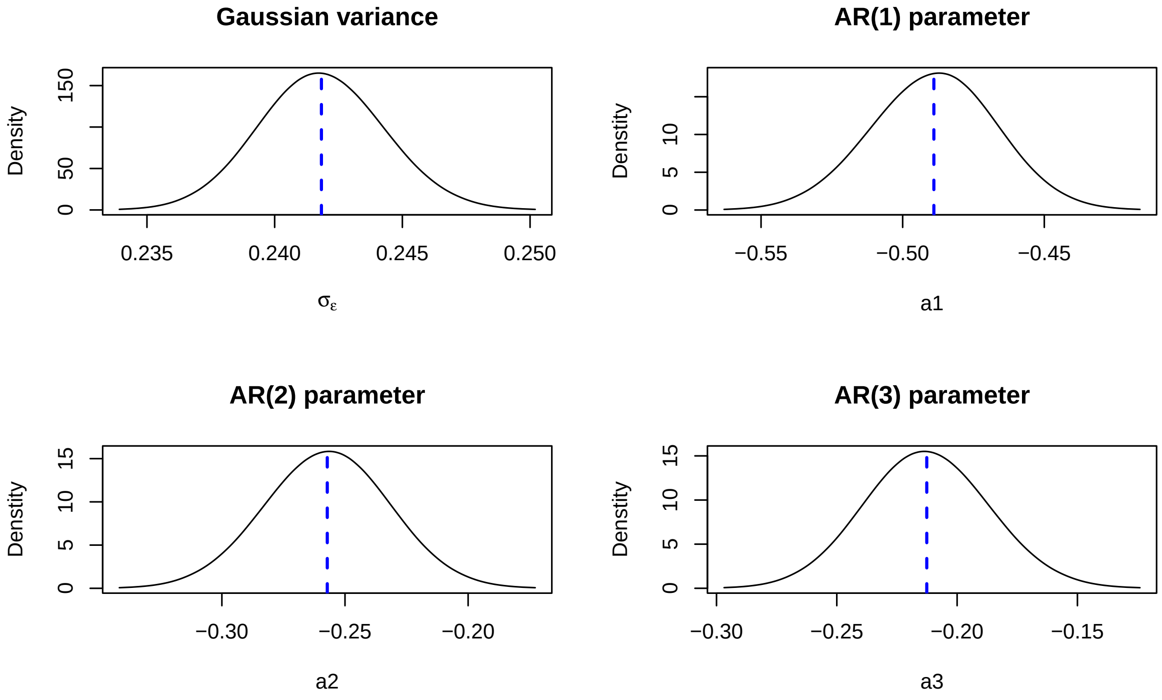


Figure 3.28: Marginal posterior densities of σ_ε, a_1, a_2, a_3.

Figure 3.28 shows the marginal posteriors of the Gaussian white noise's standard deviation and the parameters of the autoregressive process. We can see how the results are similar to the model presented in Section 3.1.3. In particular, all three autoregression parameters are significant, and the mass of their distributions is concentrated around negative values. This means that the temporal part of the spatio-temporal process is negatively correlated to the previous three days. This result is coherent with the meaning of the response variable, which represents the daily difference of pollutant concentration, which oscillates around zero (see the interpretation of Figure 3.13, presented in Section 3.1.3).

In Figure 3.29, we plotted the marginal posterior distributions of the parameters of the spatial part of the model, related to the Gaussian Process with Matérn covariance function. Specifically, we see the distributions of the standard deviation σ_ω and the range ρ, expressed in meters and related to parameter k. Notably, the densities are centered around significantly lower values than in Model 1 (refer to Figure 3.24). This discrepancy is caused by the coarser mesh, which results in fewer GMRF evaluation points, leading to a lower standard deviation. The estimated range is roughly 190 km, which is slightly lower than the previous estimate but still of a similar magnitude.

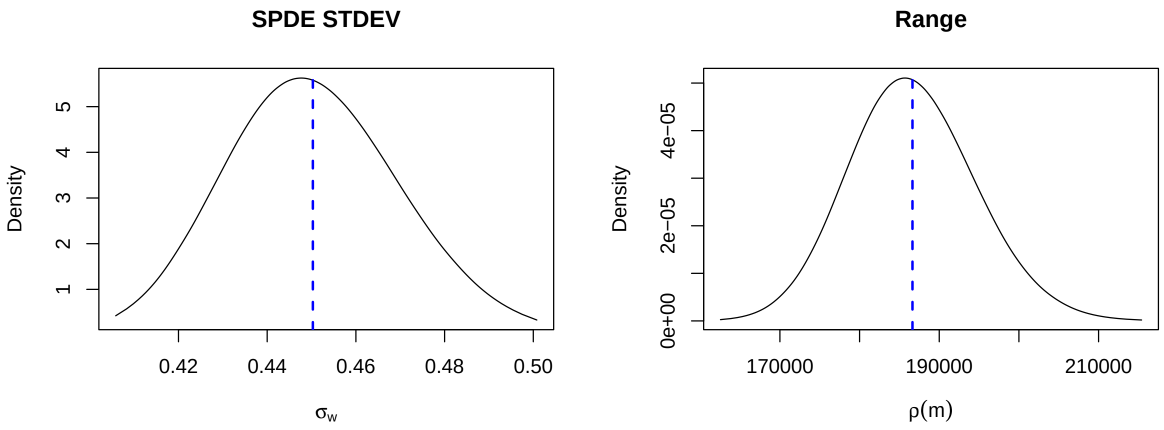


Figure 3.29: Marginal posterior densities of σ_ω, $\rho(m)$.

We can now look at the credibility intervals of the regression variables of the model, shown in Figure 3.30. It is immediate to notice the differences between the results of this model with the results of Model 1, plotted in Figure 3.25. To interpret this dissimilarity, it is crucial to remember that the response variable has been transformed differently, from the standardized logPM10 to the detrended consecutive difference of logPM10 (see Figure 3.18). This manipulation of the data can explain the differences in the results. Now the regression is targeted to the daily increase or decrease of logPM10 and not to the concentration of the pollutant itself. In this perspective, results shown in Figure 3.30 are arguably consistent and reflect the transformation of the response variable, starting with the **Intercept** coefficient, which is centered around the value zero, as it should be.

Moreover, we can see how regression variables that express general characteristics of the stations, namely **Altitude** and **Traffic**, turn out to be not relevant. Daily changes in the concentration of PM10 do not depend on these variables. On the other hand, our statistical analysis supports the intuitive notion that weather conditions play a role in the fluctuation of air pollution levels. As anticipated, the coefficients for **AvgWind** and **TotalRain** are consistently negative. This is not surprising: it is common sense that through rain and wind, the air gets cleaner. The coefficient related to **AvgTemp** regressor is still negative (as in Model 1), but its 95% credibility interval contains the value zero, so it cannot be considered significant, even if the interval is very asymmetric toward negative values. Finally, we want to point out how the absolute values of these credibility intervals are much smaller than those related to Model 1, plotted in Figure 3.30. This change is caused by the alternative data transformation, which forces the response variable to be close to zero, with lower variability.

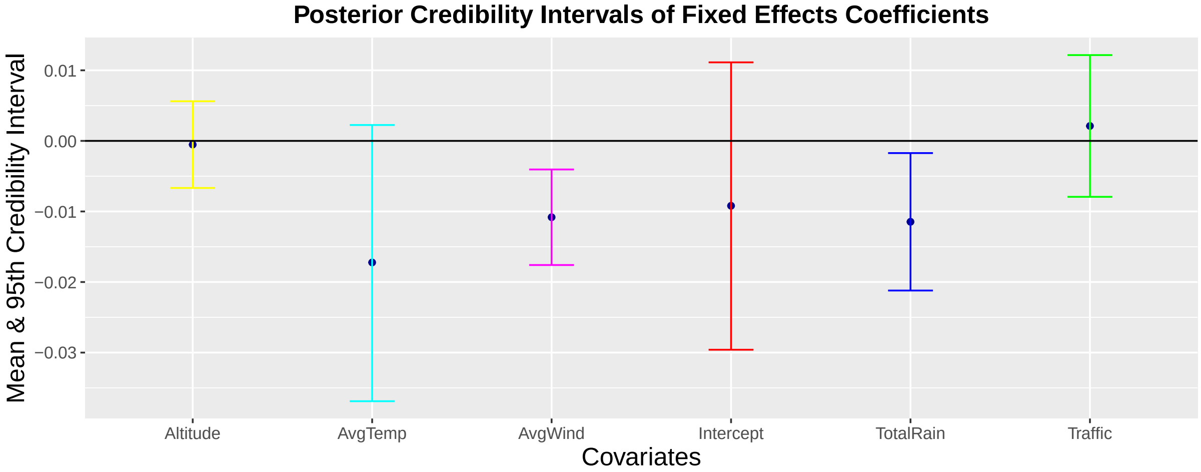


Figure 3.30: 95% credibility intervals of regression parameters.

Table 3.4 contains the summary of all the marginal posterior distributions of the parameters, summing up the posterior analysis of the model.

Parameter	Mean	SD	q_0.025	q_0.5	q_0.975
σ_ε	0.24	0.002	0.24	0.24	0.25
a_1	-0.49	0.022	-0.53	-0.49	-0.45
a_2	-0.26	0.026	-0.31	-0.26	-0.21
a_3	-0.21	0.026	-0.26	-0.21	-0.16
$Intercept$	-0.01	0.010	-0.03	-0.01	0.01
$Altitude$	0.0005	0.003	-0.01	-0.0005	0.005
$Traffic$	0.002	0.01	-0.01	0.002	0.01
$AvgTemp$	-0.02	0.01	-0.04	-0.02	0.002
$TotalRain$	-0.01	0.005	-0.02	-0.01	-0.002
$AvgWind$	-0.01	0.003	-0.02	-0.01	-0.004
$\rho\ (km)$	186.64	7.94	171.78	186.35	202.97
σ_ω	0.45	0.02	0.42	0.45	0.5

Table 3.4: Summary of the posterior marginal distributions of all the parameters in **Model 2**.

Prediction of future data

In this section, we present the results for the prediction of "future" data, consisting of 20 days after the 180 days used to estimate the posterior densities of model parameters. As for Model 1, we took two stations, one labeled as traffic and the other not, to see how our model behave in the reconstruction of the response variable and the prediction of future days. The method, implemented internally by the package *R-INLA* is, as we have already seen in Section 3.1.2, to simulate 1000 values from the posterior predictive distribution of $y(\mathbf{s}_i, t+1), ..., y(\mathbf{s}_u, t+m)$, with $m = 20$, and to compute the mean of these values, in a classical Monte Carlo approach. The results are shown in Figure 3.31.

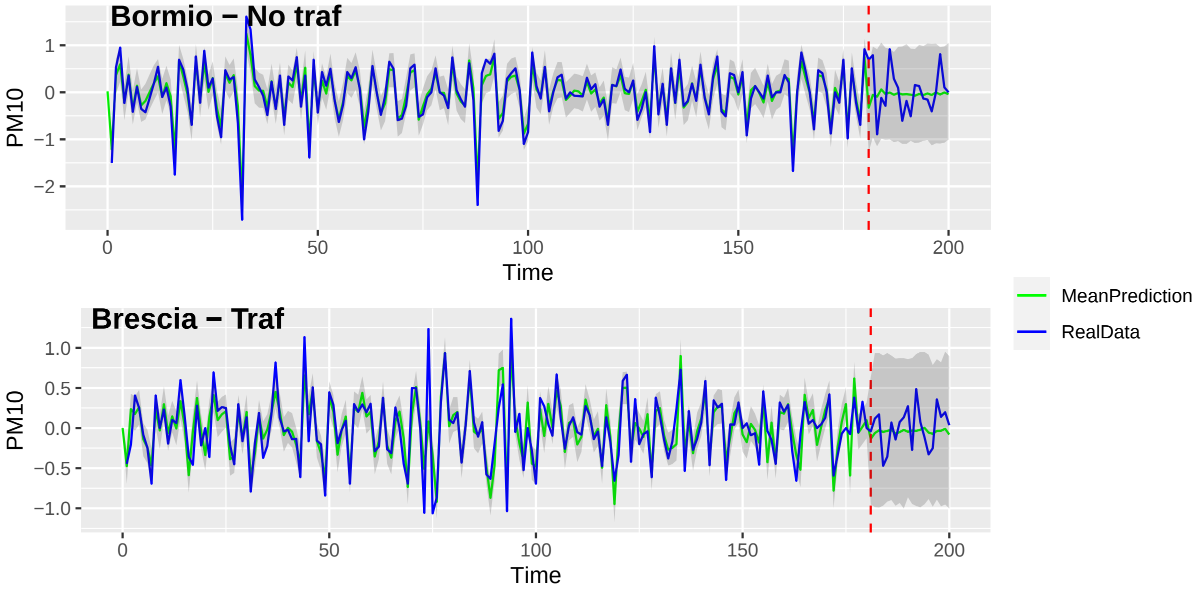


Figure 3.31: Time series of consecutive differences of logPM10 concentration (blue line) compared to the mean over 1000 samples of predicted value (green line). The grey area represents the posterior mean's 95% credibility interval. The red dashed line represents the border between past and "future" data (the 180^{th} day).

Unfortunately, as in the previous attempts, presented in Section 3.2.2, the model fails to perform future prediction satisfactorily. Although the response variable is well-explained before the past-future time border, the model only manages to stabilize around the mean after a few days of attempting to predict. As previously observed in Section 3.1.3, the only improvement is that the uncertainty of the prediction, as indicated by the green area, remains relatively constant rather than exploding.

Comparison between spatial residuals

This Section provides a practical example of estimating the GMRF that approximates the underlying Gaussian Field. To achieve this, we present in Figure 2 point estimations of the GMRF computed by the program for each point of the mesh. Specifically, we are focused on examining the computation of the spatio-temporal field identified by $\boldsymbol{\xi}(\cdot|\cdot)$, which is defined through a spatial and a temporal component as shown in equation (1.17). The function `bru`, among other output objects, generates a matrix that contains all the estimated values of the spatial Gaussian Field $\boldsymbol{\omega}(\cdot|\cdot)$ for each time instant. The matrix has dimensions of $nT \times 5$, where n is the number of points on the mesh, T is the number of days considered in the model, and 5 represents the values of mean, standard deviation, 0.025, 0.5, and 0.975 quantiles of the estimated values.

To plot the spatial residual, we chose to extract the one computed for the first day since the residuals related to the next days may be not interpretable. In fact, by taking spatial residuals for other days, we would need to pay attention to the temporal correlation with the previous days, embedded in the autoregressive model.

In Figure 3.32, we plotted the estimations of the spatial residual for Model 1 and Model 2.

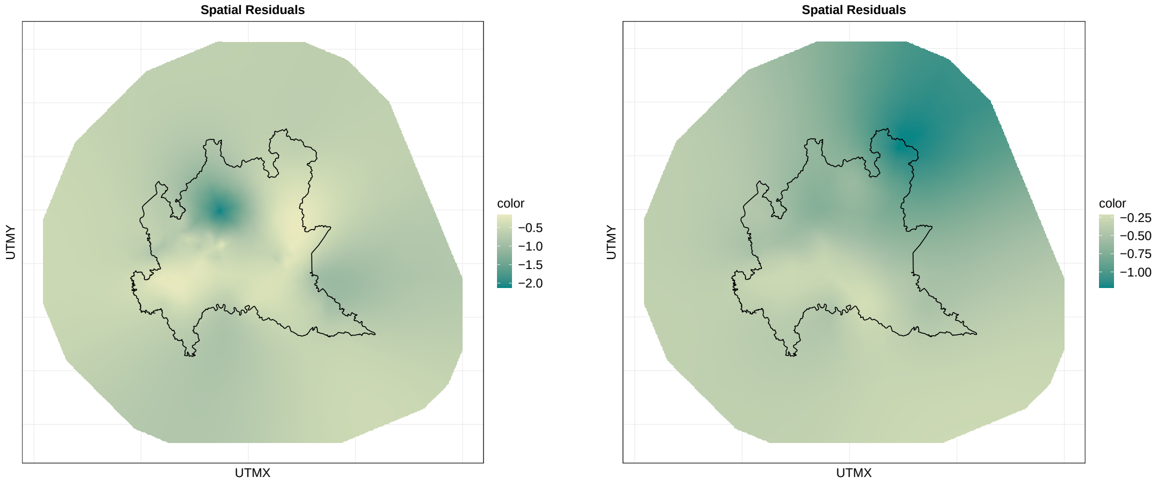


Figure 3.32: Left panel: spatial residual estimated by **Model 1**. Right panel: spatial residual estimated by **Model 2**. The data are plotted for the whole mesh, and the superimposed black line represents the true border of Lombardia.

When interpreting Figure 3.32, we have to remember the different transformations of the response variable, which reflect on the meanings of the spatial residuals, as an element of the additive regression model. The estimated values of the approximating GMRF for day one are displayed in the left panel, associated with Model 1. It is immediate to see that these values tend to be more negative in the northern part, where mountains are located, while in the plane, where cities are concentrated, we appreciate a relatively higher value for the residual.

When looking at the right panel, related to Model 2, we can see roughly the same behavior, with fewer negative residuals in areas associated with higher pollution and more negative residuals in the northern parts of the region. The second plot's interpretation is more difficult since we are studying a model that tries to explain the daily difference in logPM10 concentrations. Nonetheless, we decided to include these plots to give an idea of the structure of the objects created when using the *R-INLA* package.

3.3. Conclusions

In this section, we intend to draw some conclusions on the work presented in this Master Thesis. Firstly, we need to spend a few words about the approach implemented by the *INLA* method, built to compute Bayesian inference without resorting to expensive MCMC techniques. Of course, as always, this statistical model has advantages and disadvantages.

According to our experience, one of the most challenging aspects of using the INLA method for Bayesian inference was comprehending the computational algorithms that underlie the method, which results in the computation of marginal posterior distributions. Coding has posed another challenge when using the R-INLA package effectively. Considering the first problem, the numerical approach and the SPDE strategy for spatio-temporal data is complicated. The marginal posterior approximations are difficult to understand, especially compared to the elegant MCMC computations. Compared to simulating from true unknown posterior distributions, the approximation strategy involves more complex calculations that are not as intuitive. Chapter 2 only scratches the surface of these calculations. To be transparent, it should be noted that I was not familiar with the *INLA* approach, whereas I had prior knowledge and expertise with conventional Bayesian methods. This lack of familiarity may have played a role in our difficulties.

Regarding the *R-INLA* package, I have found it difficult to comprehend the architecture of the software. Specifically, the transition from theoretical models to the actual coding required to execute the practical models has posed a significant challenge for me. Additionally, the package implementation may be considered somewhat inflexible, as it limits the selection of components for building an additive regression model to a predetermined set. However, by combining the available options, a large variety of models, including our specific one, can still be constructed. Ultimately, I want to reiterate that the package's authors, in particular, Professors Finn Lindgren and Håvard Rue, were helpful and constantly available for clarifications and suggestions about the code writing. I greatly thank them.

Let us now focus on the results obtained through the Bayesian analysis of the air pollution datasets for Emilia-Romagna and Lombardia. First of all, we point out that this software has been quicker than standard MCMC methods in the analysis of such large datasets, reaching, in our opinion, satisfactory accuracy. The models we built have been able to explain the response variable in the regression context, and the results from the study of posterior distributions of the regression variables have been interesting. Nonetheless, some analysis outcomes, such as future predictions, have been disappointing. As we have seen throughout Chapter 3, the attempt at making future predictions of the concentration of PM10 has been unsatisfactory, even when we introduced meteorological data for Lombardia (see Section 3.2). We hoped that by integrating this information we would have made better predictions, but these expectations have not been met.

However, we want to point out that this deficit of the model in the prediction task does not necessarily mean that the problem lies in the *INLA* software or its underlying method. After all, the task of predicting such complex and suddenly changing variables is problematic. I suppose that the complex nature of the air pollution phenomenon and my inability to build better models is responsible for the poor results obtained in the future prediction task.

Finally, I cannot avoid expressing general satisfaction as a result of our work, especially when considering how many new skills have been learned. Firstly, it has been a challenging task to understand an alternative Bayesian technique and the implementing library. Moreover, I had to learn how to migrate the coding and the analysis to a Virtual Machine, which gave us much more computational power and, therefore, the possibility to explore different models without wasting huge amounts of time.

To conclude, we recommend that other analysts consider using the *INLA* program when building a Bayesian model for inference, particularly if there is an opportunity to compare the results with those obtained through conventional MCMC techniques, to find the best trade-off between computational costs and accuracy.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716-723. doi: 10.1109/TAC.1974.1100705

Bachl, F. E., Lindgren, F., Borchers, D. L., & Illian, J. B. (2019). inlabru: an R package for Bayesian spatial modelling from ecological survey data. *Methods in Ecology and Evolution*, *10*(6), 760-766.

Banerjee, S., Carlin, B. P., & Gelfand, A. E. (2015). *Hierarchical modeling and analysis for spatial data.* CRC Press.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society series B-methodological*, *36*, 192-225.

Cameletti, M., Lindgren, F., Simpson, D., & Rue, H. (2013). Spatio-temporal modeling of particulate matter concentration through the spde approach. *AStA Advances in Statistical Analysis*, *97*(2), 109–131.

Frigeri, M. (2022). Spatio-temporal models for particulate matter in the Po valley. (M.Sc thesis, Politecnico di Milano)

Gelfand, A. E., & Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, *85*, 398-409.

Geman, S., & Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-6*, 721-741.

Gomez-Rubio, V. (2020). *Bayesian inference with inla.* CRC Press.

Grange, S. K., Carslaw, D. C., Lewis, A. C., Boleti, E., & Hueglin, C. (2018). Random forest meteorological normalisation models for swiss PM10 trend analysis. *Atmospheric Chemistry and Physics*, *18*(9), 6223–6239.

Hastings, W. K. (1970, 04). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*(1), 97-109. doi: 10.1093/biomet/57.1.97

Lindgren, F., Bolin, D., & Rue, H. (2022). The spde approach for gaussian and non-gaussian fields: 10 years and still running. *Spatial Statistics*, *50*, 100599.

Lindgren, F., Rue, H., & Lindström, J. (2011). 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.

Lonati, G., & Riva, F. (2021). effetti degli interventi di contrasto alla diffusione del COVID19 sulla qualità dell'aria in Pianura Padana. *Ingegneria dell'Ambiente*, *8*(1/2021), 24-39.

Manisalidis, I., Stavropoulou, E., Stavropoulos, A., & Bezirtzoglou, E. (2020). Environmental and health impacts of air pollution: A review. *Frontiers in Public Health*, *8*.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*(6), 1087-1092.

Rue, H., & Held, L. (2005). *Gaussian markov random fields: Theory and applications.* Taylor & Francis.

Rue, H., Martino, S., & Chopin, N. (2009). 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.

Sahu, S. (2022). *Bayesian modeling of spatio-temporal data with r.* CRC.

Simpson, D., Rue, H., Riebler, A., Martins, T., & Sørbye, S. (2017). Penalising model component complexity: a principled, practical approach to constructing priors. *Statistical Science*, *32*(1), 1–28.

van Niekerk, J., Bakka, H., Rue, H., & Schenk, O. (2021). New frontiers in Bayesian modeling using the inla package in R. (1).

Whittle, P. (1954). On Stationary Processes on the PlaneS. *Biometrika*, *41*(3-4), 434-449.

Whittle, P. (1963). Stochastic-processes in several dimensions. *Bulletin of the International Statistical Institute*, *40*(2), 974–994.

A | Appendix: R Code

A.1. Emilia-Romagna: Model 1

```r
library(INLA)
library(inlabru)
library(rgdal)
library(sf)

inla.setOption(pardiso.license = "...")

#################
#DATA IMPORT
#################

#import the ShapeFile
italy = readOGR("ShapeFile/Reg01012021_g/Reg01012021_g_WGS84.shp", GDAL1_integer64_policy = TRUE)
italy <- spTransform(italy, CRS("+proj=longlat +ellps=WGS84 +datum=WGS84"))
map <- italy[italy$DEN_REG == "Emilia-Romagna",]

#converto to UTM
map <- map %>%
  st_as_sf()
map <- map %>% st_transform(32633)

emilia <- read.csv("emilia_2018_dummies.csv")

#standardize log pm
emilia$logPM10 <- log(emilia$Val)
mean_log <- mean(emilia$logPM10)
sd_log <- sd(emilia$logPM10)
emilia$logPM10 <- scale(emilia$logPM10, mean_log, sd_log)

#standardize quote
mean_quota <- mean(emilia$Quota)
```

```r
sd_quota <- sd(emilia$Quota)
emilia$Quota <- scale(emilia$Quota, mean_quota, sd_quota)

##convertion to UTM
ll.coord <- st_as_sf(data.frame(long = emilia$Long, lat = emilia$Lat),
                     coords = c("long","lat"))

st_crs(ll.coord) <- st_crs(4326)

ll.coord <- ll.coord %>% st_transform(32633)

emilia[,c("UTMX","UTMY")] <- st_coordinates(ll.coord)
emilia <- emilia[, -c(1,2,3,4,5,6,7,10,11,13,14)]

bound <- st_boundary(map)

#################
#MODEL BUILDING
#################

#mesh
coo <- cbind(emilia$UTMX, emilia$UTMY)
bnd <- inla.nonconvex.hull(st_coordinates(map)[,1:2])
mesh <- inla.mesh.2d(loc = coo, boundary = bnd, max.edge = c(16000,50000),cutoff = 100)

spde <- inla.spde2.pcmatern(mesh = mesh, alpha = 2,
                            prior.range = c(10000,0.01),
                            prior.sigma = c(3,0.01))

coordinates(emilia) <- ~ UTMX + UTMY

#3 months estimation + 10 days future prediction
data <- emilia[which(emilia$time <= 90),]

cmp = logPM10 ~ Intercept(1) + SPDE(coordinates, model = spde,
                                    group = time,ngroup = 100, control.group = list(model = "ar", order = 3)) +
  Quota + traffico

lik <- inlabru::like(formula = logPM10 ~ Intercept + SPDE + Quota + traffico,
                     family = "gaussian",
                     data = data)
```

```

start_time <- Sys.time()
fit = bru(cmp, lik,
          options = list(
            verbose = TRUE,
            inla.mode = "experimental",
            control.inla = list(strategy = "simplified.laplace", int.
   strategy = "eb"),
            num.threads = "12:2"
          ))
end_time <- Sys.time()
fit_time = end_time - start_time

#################
#POSTERIOR ANALYSIS
#################

#MARGINAL POSTERIORS
sig.eps.marg <- inla.tmarginal(function(x) (1/sqrt(x)),
                               fit$marginals.hyperpar$‘Precision for the
    Gaussian observations‘)

a1.temporal.marg <- inla.tmarginal(function(x) x,
                                   fit$marginals.hyperpar$‘Group PACF1
   for SPDE‘)
a2.temporal.marg <- inla.tmarginal(function(x) x,
                                   fit$marginals.hyperpar$‘Group PACF2
   for SPDE‘)
a3.temporal.marg <- inla.tmarginal(function(x) x,
                                   fit$marginals.hyperpar$‘Group PACF3
   for SPDE‘)

field.residuals <- inla.spde.result(fit, name = "SPDE", spde = spde, do.
   tranf = T )
spde.sig.w.marg <- sqrt(field.residuals$marginals.variance.nominal[[1]])
spde.range.marg <- field.residuals$marginals.range.nominal[[1]]
fixed.marg <- fit$marginals.fixed


#POSTERIOR SUMMARIES
spde.sig.w.summary <- as.data.frame(inla.zmarginal(spde.sig.w.marg))

spde.range.summary <- as.data.frame(inla.zmarginal(spde.range.marg))

```

```r
sig.eps.summary <- as.data.frame(inla.zmarginal(sig.eps.marg))

a1.temporal.summary <- as.data.frame(inla.zmarginal(a1.temporal.marg))
a2.temporal.summary <- as.data.frame(inla.zmarginal(a2.temporal.marg))
a3.temporal.summary <- as.data.frame(inla.zmarginal(a3.temporal.marg))

#################
#FUTURE PREDICTION
#################

time.total <- fit$size.random[[1]]$ngroup
time.border <- length(levels(as.factor(data$time)))

#BADIA (no traffic station)
badia <- emilia[which(emilia$Station.ID==1 & emilia$time <= time.total),]

start_time <- Sys.time()
pred.badia = predict(fit, badia, ~ Intercept + SPDE + Quota + traffico,
                     n.samples = 1000)
end_time <- Sys.time()

pred_time_badia <- end_time - start_time

#TIMAVO (traffic station)
timavo <- emilia[which(emilia$Station.ID==45 & emilia$time <= time.total) ,]

start_time <- Sys.time()
pred.timavo = predict(fit, timavo, ~ Intercept + SPDE + Quota + traffico,
                      n.samples = 1000)
end_time <- Sys.time()
pred_time_timavo <- end_time - start_time
```

A.2. Emilia-Romagna: Model 2

```r
library(data.table)
library(INLA)
library(inlabru)
library(rgdal)
library(sf)

inla.setOption(pardiso.license = "...")

#################
#DATA IMPORT
#################

#import the ShapeFile
italy = readOGR("ShapeFile/Reg01012021_g/Reg01012021_g_WGS84.shp", GDAL1_integer64_policy = TRUE)
italy <- spTransform(italy, CRS("+proj=longlat +ellps=WGS84 +datum=WGS84"))
map <- italy[italy$DEN_REG == "Emilia-Romagna",]

#converto to UTM
map <- map %>%
  st_as_sf()
map <- map %>% st_transform(32633)

emilia <- read.csv("emilia_2018_dummies.csv")

#standardize quote
mean_quota <- mean(emilia$Quota)
sd_quota <- sd(emilia$Quota)
emilia$Quota <- scale(emilia$Quota, mean_quota, sd_quota)

##convertion to UTM
ll.coord <- st_as_sf(data.frame(long = emilia$Long, lat = emilia$Lat),
                     coords = c("long","lat"))

st_crs(ll.coord) <- st_crs(4326)

ll.coord <- ll.coord %>% st_transform(32633)

emilia[,c("UTMX","UTMY")] <- st_coordinates(ll.coord)
emilia <- emilia[, -c(1,2,3,5,6,7,10,11,13,14)]

```

```
#Build consecutive difference variable
emilia$logPM10 <- log(emilia$Val)
setDT(emilia, key = "Station.ID")[, dif := logPM10-data.table::shift(
   logPM10), "Station.ID"]
emilia$dif <- scale(emilia$dif, center = mean(emilia$dif,na.rm=T))

bound <- st_boundary(map)

#################
#MODEL BUILDING
#################

#mesh
coo <- cbind(emilia$UTMX, emilia$UTMY)
bnd <- inla.nonconvex.hull(st_coordinates(map)[,1:2])
mesh <- inla.mesh.2d(loc = coo, boundary = bnd, max.edge = c
   (16000,50000),cutoff = 100)

spde <- inla.spde2.pcmatern(mesh = mesh, alpha = 2,
                            prior.range = c(10000,0.01),
                            prior.sigma = c(3,0.01))

coordinates(emilia) <- ~ UTMX + UTMY

#4 months estimation + 20 days future prediction
data <- emilia[which(emilia$time <= 120),]

cmp = dif ~ Intercept(1) + SPDE(coordinates, model = spde,
                                group = time,ngroup = 140, control.group
    = list(model = "ar", order = 3)) +
  Quota + traffico

lik <- inlabru::like(formula = dif ~ Intercept + SPDE + Quota + traffico
   ,
                     family = "gaussian",
                     data = data)

start_time <- Sys.time()
fit = bru(cmp, lik,
          options = list(
            verbose = TRUE,
            inla.mode = "experimental",
            control.inla = list(strategy = "simplified.laplace", int.
   strategy = "eb"),
            num.threads = "12:2"
          ))
end_time <- Sys.time()
fit_time = end_time - start_time

#################
#POSTERIOR ANALYSIS
#################

#MARGINAL POSTERIORS
sig.eps.marg <- inla.tmarginal(function(x) (1/sqrt(x)),
                               fit$marginals.hyperpar$‘Precision for the Gaussian observations‘)

a1.temporal.marg <- inla.tmarginal(function(x) x,
                                   fit$marginals.hyperpar$‘Group PACF1 for SPDE‘)
a2.temporal.marg <- inla.tmarginal(function(x) x,
                                   fit$marginals.hyperpar$‘Group PACF2 for SPDE‘)
a3.temporal.marg <- inla.tmarginal(function(x) x,
                                   fit$marginals.hyperpar$‘Group PACF3 for SPDE‘)

field.residuals <- inla.spde.result(fit, name = "SPDE", spde = spde, do.tranf = T )
spde.sig.w.marg <- sqrt(field.residuals$marginals.variance.nominal[[1]])
spde.range.marg <- field.residuals$marginals.range.nominal[[1]]
fixed.marg <- fit$marginals.fixed


#POSTERIOR SUMMARIES
spde.sig.w.summary <- as.data.frame(inla.zmarginal(spde.sig.w.marg))

spde.range.summary <- as.data.frame(inla.zmarginal(spde.range.marg))

sig.eps.summary <- as.data.frame(inla.zmarginal(sig.eps.marg))

a1.temporal.summary <- as.data.frame(inla.zmarginal(a1.temporal.marg))
a2.temporal.summary <- as.data.frame(inla.zmarginal(a2.temporal.marg))
a3.temporal.summary <- as.data.frame(inla.zmarginal(a3.temporal.marg))

#################
#FUTURE PREDICTION
```

```r
#################

time.total <- fit$size.random[[1]]$ngroup
time.border <- length(levels(as.factor(data$time)))

#BADIA (no traffic station)
badia <- emilia[which(emilia$Station.ID==1 & emilia$time <= time.total) ,]

start_time <- Sys.time()
pred.badia = predict(fit, badia, ~ Intercept + SPDE + Quota + traffico,
                     n.samples = 1000)
end_time <- Sys.time()

pred_time_badia <- end_time - start_time

#TIMAVO (traffic station)
timavo <- emilia[which(emilia$Station.ID==45 & emilia$time <= time.total ) ,]

start_time <- Sys.time()
pred.timavo = predict(fit, timavo, ~ Intercept + SPDE + Quota + traffico ,
                      n.samples = 1000)
end_time <- Sys.time()
pred_time_timavo <- end_time - start_time
```

A.3. Lombardia: Model 1

```r
library(INLA)
library(inlabru)
library(rgdal)
library(sf)
inla.setOption(pardiso.license = "...")

#################
#DATA IMPORT (PM, station coordinates and borders)
#################

#import the ShapeFile
italy = readOGR("ShapeFile/Reg01012021_g/Reg01012021_g_WGS84.shp", GDAL1_integer64_policy = TRUE)
italy <- spTransform(italy, CRS("+proj=longlat +ellps=WGS84 +datum=WGS84"))
map <- italy[italy$DEN_REG == "Lombardia",]

#converto to UTM
map <- map %>%
  st_as_sf()
map <- map %>% st_transform(32633)

lombardia <- read.csv("lombardia_2018.csv")
mean_log <- mean(lombardia$logPM10)
sd_log <- sd(lombardia$logPM10)

lombardia$logPM10 <- scale(lombardia$logPM10, mean_log, sd_log)

ll.coord <- st_as_sf(data.frame(long = lombardia$Long, lat = lombardia$Lat),
                     coords = c("long","lat"))

st_crs(ll.coord) <- st_crs(4326)

ll.coord <- ll.coord %>% st_transform(32633)
lombardia[,c("UTMX","UTMY")] <- st_coordinates(ll.coord)

bound <- st_boundary(map)

#mesh
coo <- cbind(lombardia$UTMX, lombardia$UTMY)
bnd <- inla.nonconvex.hull(st_coordinates(map)[,1:2])
```

```r
mesh <- inla.mesh.2d(loc = coo, boundary = bnd, max.edge = c(16000,50000),cutoff = 100)

spde <- inla.spde2.pcmatern(mesh = mesh, alpha = 2,
                            prior.range = c(10000,0.01),
                            prior.sigma = c(3,0.01))


coordinates(lombardia) <- ~ UTMX + UTMY

#3 months estimation + 10 days future prediction
data <- lombardia[which(lombardia$Days <= 90),]

cmp = logPM10 ~ Intercept(1) + SPDE(coordinates, model = spde,
                                    group = Days, ngroup = 100, control.group = list(model = "ar", order = 1)) +
  Quota + traf  + AvgRad + AvgTemp + TotalRain + AvgWind

lik <- inlabru::like(formula = logPM10 ~ Intercept + SPDE + Quota + traf +
                       AvgRad + AvgTemp + TotalRain + AvgWind,
                     family = "gaussian",
                     data = data)

start_time <- Sys.time()
fit = bru(cmp, lik,
          options = list(
            verbose = TRUE,
            inla.mode = "experimental",
            control.compute = list(waic = TRUE, cpo = TRUE, config = TRUE),
            control.inla = list(strategy = "simplified.laplace", int.strategy = "eb"),
            num.threads = "12:2"
          ))
end_time <- Sys.time()
fit_time <- end_time - start_time

#################
#POSTERIOR ANALYSIS
#################

#MARGINAL POSTERIORS
sig.eps.marg <- inla.tmarginal(function(x) (1/sqrt(x)),
                              fit$marginals.hyperpar$‘Precision for the Gaussian observations‘)

a1.temporal.marg <- inla.tmarginal(function(x) x,
                                   fit$marginals.hyperpar$‘Group PACF1 for SPDE‘)
field.residuals <- inla.spde.result(fit, name = "SPDE", spde = spde, do.tranf = T )
spde.sig.w.marg <- sqrt(field.residuals$marginals.variance.nominal[[1]])
spde.range.marg <- field.residuals$marginals.range.nominal[[1]]
fixed.marg <- fit$marginals.fixed

#POSTERIOR SUMMARIES
print("posterior summaries")

spde.sig.w.summary <- as.data.frame(inla.zmarginal(spde.sig.w.marg))
spde.sig.w.summary <- spde.sig.w.summary[,c("mean","sd","quant0.025","quant0.5","quant0.975")]

spde.range.summary <- as.data.frame(inla.zmarginal(spde.range.marg))
spde.range.summary <- spde.range.summary[,c("mean","sd","quant0.025","quant0.5","quant0.975")]

sig.eps.summary <- as.data.frame(inla.zmarginal(sig.eps.marg))
sig.eps.summary <- sig.eps.summary[,-c(4,6)]

a1.temporal.summary <- as.data.frame(inla.zmarginal(a1.temporal.marg))
a1.temporal.summary <- a1.temporal.summary[,-c(4,6)]

colnames(sig.eps.summary) <- c("mean","sd","0.025quant","0.5quant","0.975quant")
colnames(a1.temporal.summary) <- c("mean","sd","0.025quant","0.5quant","0.975quant")

spde.param.summ <- rbind(spde.range.summary,spde.sig.w.summary)
colnames(spde.param.summ) <- c("mean","sd","0.025quant","0.5quant","0.975quant")

fixed.summary <- fit$summary.fixed[,1:5]
hyper.summary <- rbind(sig.eps.summary,a1.temporal.summary,
                       spde.param.summ, fixed.summary)
rownames(hyper.summary) <- c("Gaussian iid stdev", "AR(1) param",
                             "Spde Range", "Spde stdev", "Intercept","Quota","Traffico",
                             "AvgRad","AvgTemp", "TotalRain", "AvgWind")

#################
#FUTURE PREDICTION
#################

time.total <- fit$size.random[[1]]$ngroup
time.border <- length(levels(as.factor(data$Days)))

#BORMIO PREDICTION
bormio <- lombardia[which(lombardia$NomeStazione == "Bormio" & lombardia$Days <= time.total) ,]

start_time <- Sys.time()
pred.bormio = predict(fit, bormio, ~ Intercept + SPDE + Quota + traf + AvgRad + AvgTemp
                      + TotalRain + AvgWind,
                      n.samples = 1000)
end_time <- Sys.time()
pred.time.bormio <- end_time - start_time

#BRESCIA PREDICTION
Brescia.broletto <- lombardia[which(lombardia$NomeStazione== "Brescia - Broletto" & lombardia$Days <= time.total) ,]
start_time <- Sys.time()

pred.brescia = predict(fit, Brescia.broletto, ~ Intercept + SPDE + Quota + traf + AvgRad + AvgTemp
                       + TotalRain + AvgWind,
                       n.samples = 1000)
end_time <- Sys.time()
pred.time.brescia <- end_time - start_time
```

A.4. Lombardia: Model 2

```r
library(INLA)
library(inlabru)
library(rgdal)
library(sf)
library(data.table)
inla.setOption(pardiso.license = "...")

#################
#DATA IMPORT (PM, station coordinates and borders)
#################

#import the ShapeFile
italy = readOGR("ShapeFile/Reg01012021_g/Reg01012021_g_WGS84.shp", GDAL1_integer64_policy = TRUE)
italy <- spTransform(italy, CRS("+proj=longlat +ellps=WGS84 +datum=WGS84"))
map <- italy[italy$DEN_REG == "Lombardia",]

#converto to UTM
map <- map %>%
  st_as_sf()
map <- map %>% st_transform(32633)

lombardia <- read.csv("lombardia_2018.csv")

ll.coord <- st_as_sf(data.frame(long = lombardia$Long, lat = lombardia$Lat),
                     coords = c("long","lat"))

st_crs(ll.coord) <- st_crs(4326)

ll.coord <- ll.coord %>% st_transform(32633)
lombardia[,c("UTMX","UTMY")] <- st_coordinates(ll.coord)

setDT(lombardia, key = "NomeStazione")[, dif := logPM10-shift(logPM10), "NomeStazione"]
lombardia$dif <- lombardia$dif - mean(lombardia$dif, na.rm=T)

bound <- st_boundary(map)

#mesh
coo <- cbind(lombardia$UTMX, lombardia$UTMY)
```

```r
bnd <- inla.nonconvex.hull(st_coordinates(map)[,1:2])
mesh <- inla.mesh.2d(loc = coo, boundary = bnd, max.edge = c(25000,70000),cutoff = 100)

spde <- inla.spde2.pcmatern(mesh = mesh, alpha = 2,
                            prior.range = c(10000,0.01),
                            prior.sigma = c(3,0.01))


coordinates(lombardia) <- ~ UTMX + UTMY

#6 months estimation + 20 days future prediction
data <- lombardia[which(lombardia$Days <= 180),]

cmp = dif ~ Intercept(1) + SPDE(coordinates, model = spde,
                                    group = Days, ngroup = 200, control.group = list(model = "ar", order = 3)) +
  Quota + traf  + AvgTemp + TotalRain + AvgWind

lik <- inlabru::like(formula = dif ~ Intercept + SPDE + Quota + traf  + AvgTemp + TotalRain + AvgWind ,
                     family = "gaussian",
                     data = data)

start_time <- Sys.time()
fit = bru(cmp, lik,
          options = list(
            verbose = TRUE,
            inla.mode = "experimental",
            num.threads = "12:2"
          ))
end_time <- Sys.time()
fit_time <- end_time - start_time


#################
#POSTERIOR DESNSITIES OF PARAMETER AND HYPERPARAM
#################

print("posterior marginals")
#POSTERIOR MARGINALS
sig.eps.marg <- inla.tmarginal(function(x) (1/sqrt(x)),
                               fit$marginals.hyperpar$`Precision for the Gaussian observations`)
```

```

a1.temporal.marg <- inla.tmarginal(function(x) x,
                                   fit$marginals.hyperpar$‘Group PACF1 for SPDE‘)
a2.temporal.marg <- inla.tmarginal(function(x) x,
                                  fit$marginals.hyperpar$‘Group PACF2 for SPDE‘)
a3.temporal.marg <- inla.tmarginal(function(x) x,
                                  fit$marginals.hyperpar$‘Group PACF3 for SPDE‘)

field.residuals <- inla.spde.result(fit, name = "SPDE", spde = spde, do.tranf = T )
spde.sig.w.marg <- sqrt(field.residuals$marginals.variance.nominal[[1]])
spde.range.marg <- field.residuals$marginals.range.nominal[[1]]
fixed.marg <- fit$marginals.fixed

#POSTERIOR SUMMARIES
#look at sigma (sdev not variance)
print("posterior summaries")

spde.sig.w.summary <- as.data.frame(inla.zmarginal(spde.sig.w.marg))
spde.sig.w.summary <- spde.sig.w.summary[,c("mean","sd","quant0.025","quant0.5","quant0.975")]

spde.range.summary <- as.data.frame(inla.zmarginal(spde.range.marg))
spde.range.summary <- spde.range.summary[,c("mean","sd","quant0.025","quant0.5","quant0.975")]

sig.eps.summary <- as.data.frame(inla.zmarginal(sig.eps.marg))
sig.eps.summary <- sig.eps.summary[,-c(4,6)]

a1.temporal.summary <- as.data.frame(inla.zmarginal(a1.temporal.marg))
a1.temporal.summary <- a1.temporal.summary[,-c(4,6)]
a2.temporal.summary <- as.data.frame(inla.zmarginal(a2.temporal.marg))
a2.temporal.summary <- a2.temporal.summary[,-c(4,6)]
a3.temporal.summary <- as.data.frame(inla.zmarginal(a3.temporal.marg))
a3.temporal.summary <- a3.temporal.summary[,-c(4,6)]


colnames(sig.eps.summary) <- c("mean","sd","0.025quant","0.5quant","0.975quant")
colnames(a1.temporal.summary) <- c("mean","sd","0.025quant","0.5quant","0.975quant")
```

```r
colnames(a2.temporal.summary) <- c("mean","sd","0.025quant","0.5quant","0.975quant")
colnames(a3.temporal.summary) <- c("mean","sd","0.025quant","0.5quant","0.975quant")

spde.param.summ <- rbind(spde.range.summary,spde.sig.w.summary)
colnames(spde.param.summ) <- c("mean","sd","0.025quant","0.5quant","0.975quant")

fixed.summary <- fit$summary.fixed[,1:5]
rownames(fixed.summary) <- c("Intercept", "Altitude", "Traffic", "AvgTemp", "TotalRain", "AvgWind")

hyper.summary <- rbind(sig.eps.summary,a1.temporal.summary, a2.temporal.summary, a3.temporal.summary,
                       spde.param.summ, fixed.summary)
rownames(hyper.summary) <- c("Gaussian iid stdev", "AR(1) param", "AR(2) param", "AR(3) param",
                             "Spde Range", "Spde stdev", "Intercept","Altitude","Traffic",
                             "AvgTemp", "TotalRain", "AvgWind")

#################
#FUTURE PREDICTION
#################

time.total <- fit$size.random[[1]]$ngroup
time.border <- length(levels(as.factor(data$Days)))

#BORMIO PREDICTION
bormio <- lombardia[which(lombardia$NomeStazione == "Bormio" & lombardia$Days <= time.total) ,]

start_time <- Sys.time()
pred.bormio = predict(fit, bormio, ~ Intercept + SPDE +  Quota + traf + AvgTemp + TotalRain + AvgWind,
                      n.samples = 1000)
end_time <- Sys.time()
pred.time.bormio <- end_time - start_time

#BRESCIA PREDICTION
Brescia.broletto <- lombardia[which(lombardia$NomeStazione== "Brescia - Broletto" & lombardia$Days <= time.total) ,]
start_time <- Sys.time()
```

```r

pred.brescia = predict(fit, Brescia.broletto, ~ Intercept + SPDE + Quota + traf  + AvgTemp + TotalRain + AvgWind,
                           n.samples = 1000)
end_time <- Sys.time()
pred.time.brescia <- end_time - start_time
```

List of Figures

List of Tables

Acknowledgements

I would like to express my sincere gratitude to Professors Finn Lingren and Håvard Rue for their invaluable assistance with the code and their expertise in using the *INLA* software. Additionally, I would like to extend my appreciation to Ph.D. student Matteo Gianella for his constant support and patience in helping me set up and use the virtual machine provided by DataCloud. Finally, I would like to acknowledge Ph.D. student Michela Frigeri for her outstanding contribution in preparing, cleaning, and merging the datasets, which made my life much easier.