POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

AMBRA

Final report of the group project of the Advance Computer Architecture course

Sergio Pardo, Angela Remolina, Camilo Sinning

Advisor:
Prof. Christian Pilato

Academic year:
2023-2024

Abstract:
This report presents the application of a novel variant of Bayesian Additive Regression Trees (BART) proposed by Kim (2022) for spatial data analysis, specifically tailored to sparse spatial observations. The project aims to analyze and predict air pollution data in Lombardy, Italy, using the AgrImOnIA dataset. The main objectives of this project include understanding and adapting Kim's R code for the MCMC algorithm to suit the project's requirements, integrating additional data sources to construct a dataset of predictors, constructing a Structurally Informed Adjacency Matrix (SIAM), and finally, using the Spatial BART (SBART) model for predicting PM 2.5 concentrations in regional municipalities. The report discusses the methodology employed, challenges faced, and insights gained throughout the project.

Contents

1. Introduction

The AgrImOnIA project (which stands for Agriculture Impact On Italian Air) is a research endeavor initiated and coordinated by the Università di Bergamo, in collaboration with numerous other universities. Its primary aim is to investigate the impact of agriculture on air quality in Italy, with a particular focus on the Lombardy region.
As a team, our assignment was to utilize a relatively new Bayesian non-parametric model to forecast pollution levels in the municipalities of Lombardy using data collected by the AgrImOnIA project. Additionally, we were tasked with evaluating the model's validity and effectiveness using this project as a testing ground.
To accomplish this, we utilized two components of the dataset. The first part comprises air quality and weather data collected from weather stations, while the second part is the AgrImOnIA Grid Covariate (AGC) dataset. The latter was essential for assigning covariate values to each municipality, enabling us to train the model and make predictions.
In this report, we will initially describe the model, beginning with its simplest form, and then progressively incorporate the modifications introduced by various researchers until we arrive at the final version that we utilized. Subsequently, we will describe the procedures undertaken to modify the model to meet our requirements and how we computed all the data necessary for running it. Finally, we will analyze the results and offer observations on both the acquired data and the model's ability to efficiently and effectively predict data.
All the code and results can be found in the GitHub repository at the following link: `https://github.com/Andrea01Fraschini/SBARTProject`.

2. The BART Model (Chipman et al.)

The Bayesian Additive Regression Trees model (BART for short) [**?**], is a model developed to estimate an unknown function, denoted as $f(\underline{x}_i)$, where $\underline{x}_i = (x_{i1}, \ldots, x_{ip})$ represents the covariates associated with a response variable y_i in a set of observations $\underline{y} = (y_1, \ldots, y_n)$ (with n being the number of observations).
The main features of this model are its resistance to over-fitting and by keeping track of predictor inclusion frequencies, it can also be used for model-free variable selection.
The function $f(\underline{x}_i)$ is approximated using a *sum of regression trees*.

$$y_i = f(\underline{x}_i) + \varepsilon_i \approx \sum_{t=1}^{m} g_t(\underline{x}_i) + \varepsilon_i \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma^2),\ i = 1, \ldots, n \tag{1}$$

where $g_t(\underline{x}_i) = g(\underline{x}_i; T_t, M_t)$ represents a regression tree model, given a tree structure T_t and a set of leaf parameters M_t and m is the number of trees.
Given this model, the marginal distribution of the observed data $\underline{y}$ is given by a Gaussian distribution with the sum of the trees as its mean and σ^2 as its variance. The full model is:

$$Y_i \mid \underline{x}_i, \mathcal{T}, \mathcal{M}, \sigma^2 \overset{ind}{\sim} \mathcal{N}\left(\sum_{t=1}^{m} g\left(\underline{x}_i; T_t, M_t\right), \sigma^2\right) \qquad i = 1, ..., n \tag{2}$$

$$\mu_{t1}, \ldots, \mu_{tb_t} \mid T_t \overset{iid}{\sim} \mathcal{N}(\mu_\mu, \sigma_\mu^2) \qquad t = 1, \ldots, m \tag{3}$$

$$T_1, \ldots, T_m \overset{iid}{\sim} Tree(\alpha, \beta) \tag{4}$$

$$\sigma^2 \sim inv - gamma\left(\frac{\nu}{2}, \frac{\nu\lambda}{2}\right), \tag{5}$$

where $\mathcal{T}$ is the set of tree structures, $T_t \in \mathcal{T}$, $\mathcal{M}$ is the set of leaves, $M_t \in \mathcal{M}, M_t = \{\mu_{t1}, ..., \mu_{tb_t}\}$, $b_t = |M_t|$, and $Tree$ is a (non-standard) notation used to represent the process that generates the trees. All parameters are a priori fixed based on the default settings suggested by Chipman et al. [**?**].

2.1. Tree generation process

2.1.1 Notation

We will refer to the nodes of the trees as *inner* if the node has at least a child node, and as *leaf* if the node has no child nodes. Furthermore, we indicate as *splitting rule* a rule of type $x_i \leq c_{ij}$ associated with an inner node, where x_i is called *splitting variable* and is one of the covariates ($i = 1, ..., p$, p number of covariates), c_{ij} is called *splitting constant* and is one of the available values of x_i ($j = 1, ..., n$).

2.1.2 Marginal for T_t's

The marginal for the T_t's, $Tree$, is specified by three aspects:

1. the probability that a node at depth d is an inner node is given by:

$$\alpha(1+d)^{-\beta}, \qquad \alpha \in (0,1), \beta \in [0,+\infty)$$

   By default, $\alpha = 0.95, \beta = 2$ (see Chipman et al. [**?**]).

2. the probability of choosing a splitting variable for an inner node is given by a discrete uniform distribution over all of the covariates.

3. the probability of choosing a splitting constant, conditionally on the splitting variable, is also given by a discrete uniform distribution over the values available for that variable.

2.1.3 Alteration of the tree structures

The tree structures are generated iteratively by applying incremental alterations. The MCMC is initialized with root nodes with value $\mu_0 = \bar{Y}/m$ ($\bar{Y}$ sample mean), at each iteration, a new alteration is proposed between GROW, PRUNE, CHANGE, and SWAP based on a selection probability associated to each.

- GROW: grows a leaf node into an inner node and two new leaf nodes.
- PRUNE: reverse of GROW, prunes two leaf nodes sharing a parent and changes the parent to a leaf node.
- CHANGE: changes the splitting rule in an inner node.
- SWAP: swaps the rules between a parent node and a child node, both parent and child must be inner nodes.

Once an alteration is chosen, a new tree structure is proposed and accepted (or rejected) using a step of Metropolis-Hastings.

2.2. Prior distribution

The prior induced by the model is:

$$\begin{aligned}
\pi((T_1, M_1), \ldots, (T_m, M_m), \sigma) &= \left[\prod_{i=1}^{m} \pi(T_j, M_j)\right] \pi(\sigma) \\
&= \left[\prod_{i=1}^{m} \pi(M_j \mid T_j)\pi(T_j)\right] \pi(\sigma) \\
&= \left[\prod_{i=1}^{m} \pi(T_i) \prod_{j=1}^{b_t} \pi(\mu_{ij} \mid T_i)\right] \pi(\sigma)
\end{aligned} \tag{6}$$

The prior is designed in such a way to introduce the following interpretable effects:

- **Shrinkage effect**: the distribution over $\mu_{tj} \mid T_t$ reduces the individual effects of trees by assigning more mass to μ_{tj} around 0.
- **Shallowing effect**: the distribution over the depth of tree nodes enforces shallow structures and discourages big trees.

3. Spatially adjusted BART Model (SBART)

The Spatial BART model (SBART) offers an extension for the base BART that allows to work with spatial data, specifically areal data. This model, first proposed in [**?**], can be expressed as follows:

$$Y_i \mid \underline{x}_i, \mathcal{T}, \mathcal{M}, \sigma^2 \overset{ind}{\sim} \mathcal{N}\left(\sum_{t=1}^{m} g\left(\underline{x}_i; T_t, M_t\right) + \theta_i, \sigma^2\right) \qquad i = 1, ..., n \tag{7}$$

$$\theta_i | \underline{\theta}_{(-i)}, \rho, \tau^2 \sim \mathcal{N}\left(\frac{\rho}{\sum_{k=1}^{n} w_{ik}} \sum_{k \neq i} w_{ik}\theta_k, \frac{\tau^2}{\sum_{k=1}^{n} w_{ik}}\right) \qquad i = 1, \ldots, n \tag{8}$$

$$\mu_{t1}, \ldots, \mu_{tb_t} \mid T_t \overset{iid}{\sim} \mathcal{N}(\mu_\mu, \sigma_\mu^2) \qquad t = 1, \ldots, m \tag{9}$$

$$T_1, \ldots, T_m \overset{iid}{\sim} Tree(\alpha, \beta) \tag{10}$$

$$\sigma^2 \sim inv - gamma\left(\frac{\nu}{2}, \frac{\nu\lambda}{2}\right) \tag{11}$$

$$\tau^2 \sim inv - gamma(a, b) \tag{12}$$

$$\rho \sim Uniform(0, 1), \tag{13}$$

where $\underline{\theta} = (\theta_1, \ldots, \theta_n)$ describes a spatial random effect with a CAR prior, ρ is a smoothing parameter that determines the spatial dependency between neighbors, w_{ij} denotes the (i, j)th element of the $n \times n$ adjacency matrix $\mathbf{W}$ such that $w_{ij} = 1$ if i and j $(i \neq j)$ are spatially connected ($w_{ij} = 0$ otherwise).

4. Kim's SBART Model

Kim in [**?**] proposes a new version of SBART that replaces the adjacency matrix $\mathbf{W}$ with a Structurally Informed Adjacency Matrix (SIAM) which proposes to take into account the specific spatial structure of the application. Such information may vary based on the purpose of the model.
Generally, we denote with $i \leftrightarrow j$ that the ith and jth locations are spatially connected, and we redefine $w_{ij} = I(i \leftrightarrow j)$. We also introduce a flexible distance function $f \in \mathcal{F}_d = \{f_1, f_2, ..., f_q\}$ to represent geographic proximity between locations.
The proposed SIAM matrix, as in [**?**], is defined as:

$$\textbf{SIAM}: \quad \mathbf{W}(d) = (w_{ij}), \qquad w_{ij} = \frac{1}{f(d_{ij})} I(i \leftrightarrow j), \qquad f \in \mathcal{F}_d, \tag{14}$$

where d_{ij} is the distance between i and j. Further details about the computation of the SIAM can be found in Appendix **??**. Given this new definition of the adjacency matrix, and a different CAR marginal for $\underline{\theta}$, the model proposed by Kim in [**?**] is:

$$Y_i \mid \underline{x}_i, \mathcal{T}, \mathcal{M}, \sigma^2 \overset{ind}{\sim} \mathcal{N}\left(\sum_{t=1}^{m} g\left(\underline{x}_i; T_t, M_t\right) + \theta_i, \sigma^2\right) \qquad i = 1, ..., n \tag{15}$$

$$\theta_i | \underline{\theta}_{(-i)}, \rho, \tau^2 \sim \mathcal{N}\left(\frac{\rho \sum_{k=1}^{n} w_{ik}\theta_k}{\rho \sum_{k=1}^{n} w_{ik} + 1 - \rho}, \frac{\tau^2}{\rho \sum_{k=1}^{n} w_{ik} + 1 - \rho}\right) \qquad i = 1, \ldots, n \tag{16}$$

$$\mu_{t1}, \ldots, \mu_{tb_t} \mid T_t \overset{iid}{\sim} \mathcal{N}(\mu_\mu, \sigma_\mu^2) \qquad t = 1, \ldots, m \tag{17}$$

$$T_1, \ldots, T_m \overset{iid}{\sim} Tree(\alpha, \beta) \tag{18}$$

$$\sigma^2 \sim inv - gamma\left(\frac{\nu}{2}, \frac{\nu\lambda}{2}\right) \tag{19}$$

$$\tau^2 \sim inv - gamma(a, b) \tag{20}$$

$$\rho \sim Uniform(0, 1) \tag{21}$$

Along with the already-mentioned changes, Kim also introduces a sparsity-inducing prior that substitutes the discrete selection distribution over the covariates described in Section **??**. This sparsity-inducing prior comes in the shape of a Dirichlet distribution and aims at reducing the number of included covariates in the model.

5. The AgrImOnIA dataset

The AgrImOnIA dataset contains daily registrations from 2016-2021 in the Lombardy region of the following aspects:

- Air quality.
- Weather data.
- Land allocation.
- Emissions.
- Livestock.

In specific, the dataset is divided into two main parts:
The first one can be found in the file called `AGC_Dataset_v_3_0_0.csv`, in this file we can find the columns illustrated in Table **??**. This dataset has the particularity that the covariates are registered in a grid, so for each day, we have a grid of measurements, the points where the measurements were taken can be seen in Figure **??**.

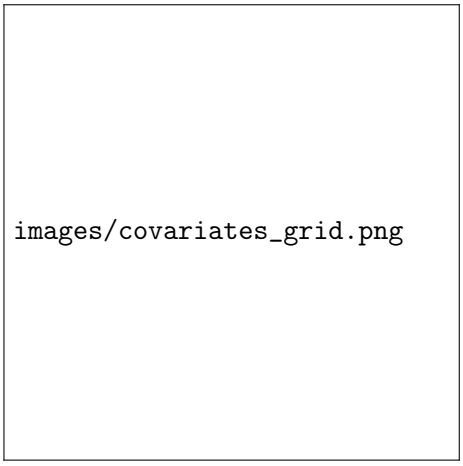


Figure 1: Agrimonia Grid Covariates points.

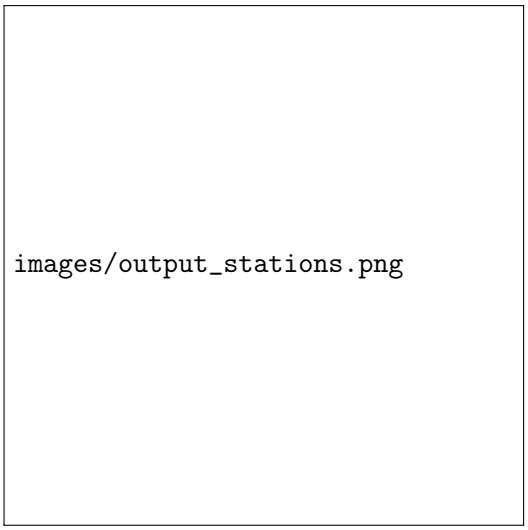


Figure 2: Stations of the AgrImOnIA dataset.

The second part of the dataset is the file `Agrimonia_Dataset_v_3_0_0.csv` that contains every data-point generated by stations located in the Lombardy region. The columns of this dataset are illustrated in Table **??**. The stations where the measurements were taken can be seen in Figure **??**.

5.1. $PM_{2.5}$

The measurements concerning $PM_{2.5}$ where taken by the stations and therefore can only be found in the second part of the dataset under the name `AQ_pm25`. In Figure **??** we show the time series of $PM_{2.5}$ values measured in $\mu g/m^3$.

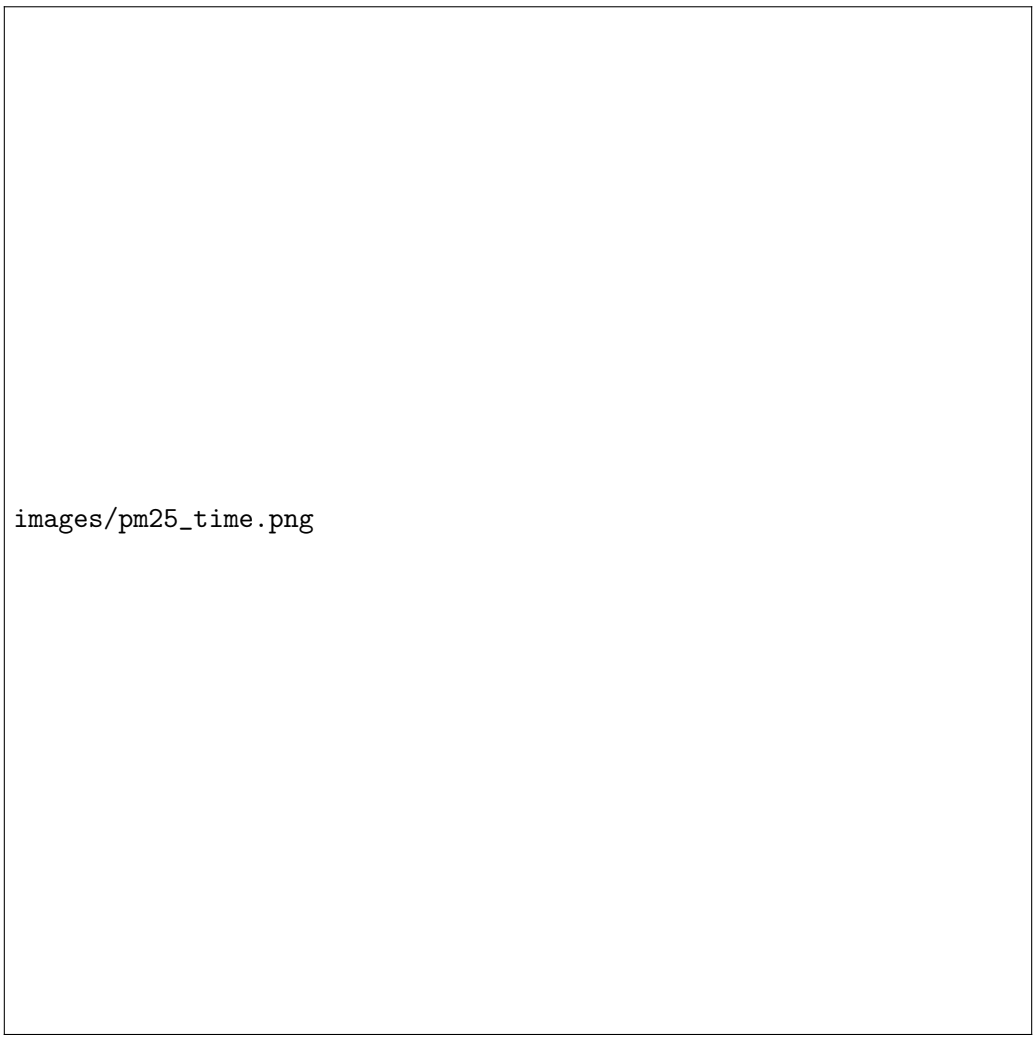


Figure 3: Time series of $PM_{2.5}$ levels between all the stations.

5.1.1 Missing values

The absence of stations in every municipality, coupled with intermittent gaps lasting days or even weeks in the data from certain stations, made so that a considerable amount of data was missing not only spatially, which is what we're trying to deal with in this project, but temporally. Additionally, many stations also seem to have stopped registering $PM_{2.5}$ values after 2021 which forced us to exclude the latter part of the dataset from our analysis. In Figure **??** we show for each day the number of missing value across all stations in the `AQ_pm25` column.

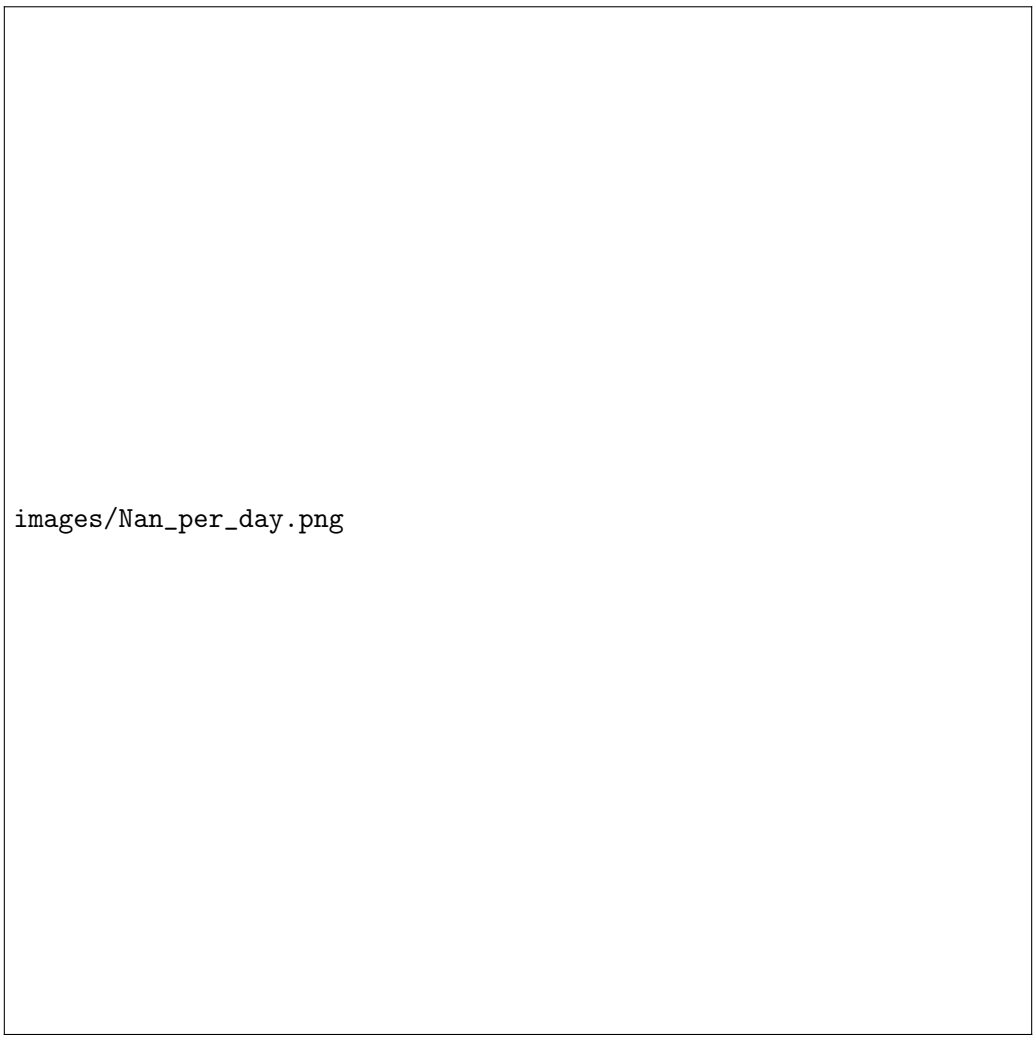


Figure 4: Missing values of $PM_{2.5}$ levels.

It's important to add that not all stations registered $PM_{2.5}$ levels. We only considered those who took at least one measurement of $PM_{2.5}$.

6. Methodology

The work for this project was carried out in three main phases:

- Data pre-processing.
- Building of adjacency matrix and wind adjacency matrix.
- Model training.

Each phase was designed to ensure that the model was in the optimal condition to predict the $PM_{2.5}$ levels with the highest accuracy possible within a feasible computational time. Mainly due to this last consideration, two main approaches were used to pre-process the data, resulting in two different models from which we obtained results.

7. First approach

At first, we considered all 1504 municipalities in Lombardy as our areas in the model as this was the original goal of the project. In this section, we will explain the results that we obtained and why we chose to opt for a different approach in section **??**.

7.1. Data pre-processing

The first step in the data pre-processing was to aggregate the data from the AgrImOnIA dataset and get the mean for each covariate within the considered period. This was done due to the limitations of the model to handle time-series data.
Then, since our goal is to predict the $PM_{2.5}$ levels for the Lombardy region, we needed a way to mix the grid covariates with the municipalities where we have observed data from the stations. For this purpose, we used an interpolation technique to transform the data from point-referenced to areal.
Through this method, our objective was to assign to each municipality within the Lombardy region a unique value for each covariate. Given the available covariates grid provided by the AgrImOnIA Dataset, we interpolated values for municipalities by taking a weighted average approach, as summarized in Figure **??**. In the figure, the black dots represent the available grid of covariates, while the red ones represent the expanded grid used to compute the weights for the average.

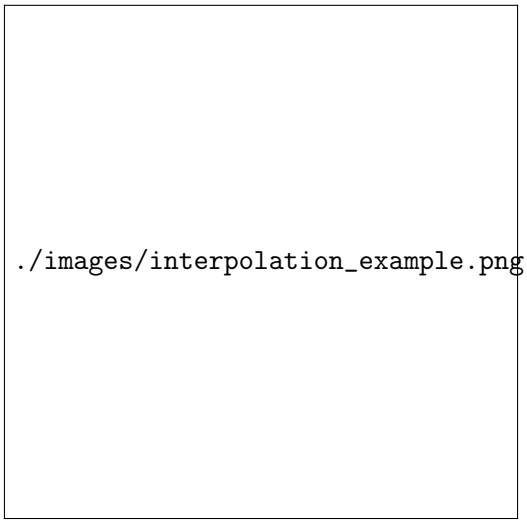


Figure 5: Interpolation example.

For each point on the AgrImOnIA grid, we generate a square with a side length equal to the distance between two points, and with the center located at the current point (like shown in Figure **??** by the dashed lines). Subsequently, we calculate the areas of intersection between each municipality and the squares containing it.
Then we compute the interpolated value by weighting the value of each point at the center of the intersected squares by the area of the intersection, normalized by the total area of the municipality. In the example in Figure **??** this would be:

$$v_{new} = \frac{A_1 \cdot P_a + A_2 \cdot P_b + A_3 \cdot P_c + A_4 \cdot P_d}{A1 + A2 + A3 + A4},$$

Where A_i's are the areas of the intersections, the P_i's are the values of the points of the covariates grid. As a result of this procedure, the interpolated value of the covariates is now associated to the municipality and not to a point on the map.

7.2. Model training

For this step, we used as a base the code provided by Kim [**?**]. The code was modified to fit the data into the model and to make it more modular.
The data generated in the previous steps was given to the model as input and, due to computational limitations, we trained the model with a limited configuration of 2000 iterations (less than the recommended, 10000), 200 trees, and 200 warmup iterations. This configuration was chosen to reduce the computational time, as the original configuration would take approximately 20 hours to run the model.

7.3. Results

We forecast the $PM_{2.5}$ values for all municipalities within the Lombardy region. Initially, we approached this task by examining a one-year interval with a limited number of iterations.

We considered various date ranges but ultimately decided to consider the year 2018. Specifically, considering all the covariates, and employing a configuration of 2000 iterations, 200 trees, and 200 warm-up iterations, we obtained the results illustrated in Figure **??**. These results were obtained using the 10-meter height and 30-degree wind adjacency matrix.

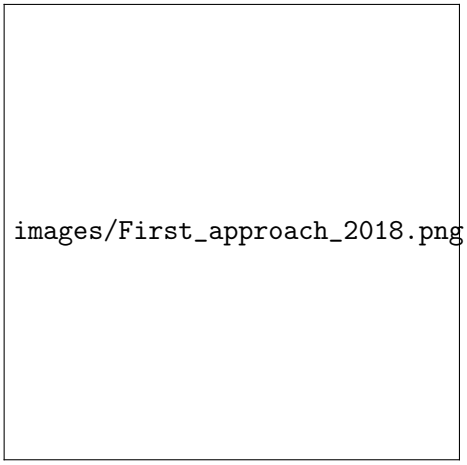


Figure 6: Predicted $PM_{2.5}$ values for 2018.

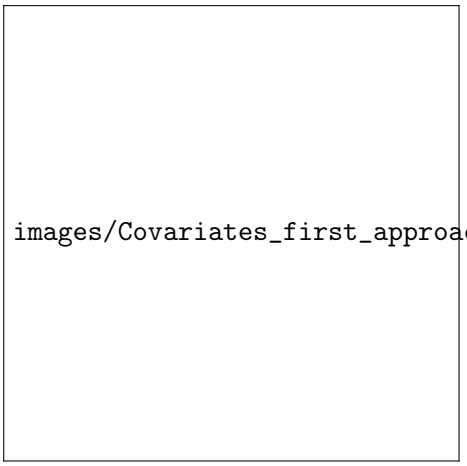


Figure 7: Frequency of covariates selection for prediction of $PM_{2.5}$ values for 2018.

Figure **??**. highlights influential covariates in our $PM_{2.5}$ concentration model. Notably, "LA_hvi" (Leafy Vegetation Index) and `EM_nox_traffic` (Traffic-related NO_x Emissions) emerge as significant factors. As explained in section **??**, "LA_hvi" measures the density of leafy vegetation, particularly focusing on low-lying types within the area, while "EM_nox_traffic" quantifies NO_x emissions from on-road transportation.
The covariates selected align with our expectations. However, it's important to note that our model was run with a reduced number of iterations (2000) of which the first 1000 iterations were designated as burn-in iterations and therefore excluded. Considering this, along with the difference in the number of iterations used, raises doubts about whether our model performs effectively.
Provided with more iterations and an appropriate time to converge, we expect the model to perform similarly to the results presented in the following sections. Unfortunately, due to the formulation of the model, it is not immediately obvious how to optimize this approach to be run faster as it requires a 1054×1504 adjacency matrix resulting in 2.262.016 potential values to be computed at each iteration.

8. Second approach

Due to the computational limitations of the first approach, we decided to use a simplified version of the Lombardy region. In this approach, we maintained the municipalities where we observed data and replaced the others with squares that aligned with the AgrImOnIA Covariate Grid.
This approach offers several advantages. Firstly, it leads to a more homogeneous prediction, as the covariates relate to natural features and are not influenced by the political boundaries of the region, such as municipalities. This new approach offers another advantage: it treats the areas around Milan with equal consideration to those in the northern part of Lombardy. Meanwhile, in the initial model, the numerous municipalities around Milan received more consideration compared to the fewer ones in the mountains.

8.1. Data pre-processing

For the second interpolation method, we integrate the covariate grid with the municipalities where we have observed data. As shown in Figure **??**. In the figure, the red crosses represent the AGC grid, and the colored shapes represent the municipalities with stations. On the right, is an example of the shapes created by mixing the grid and the actual municipality shapes.

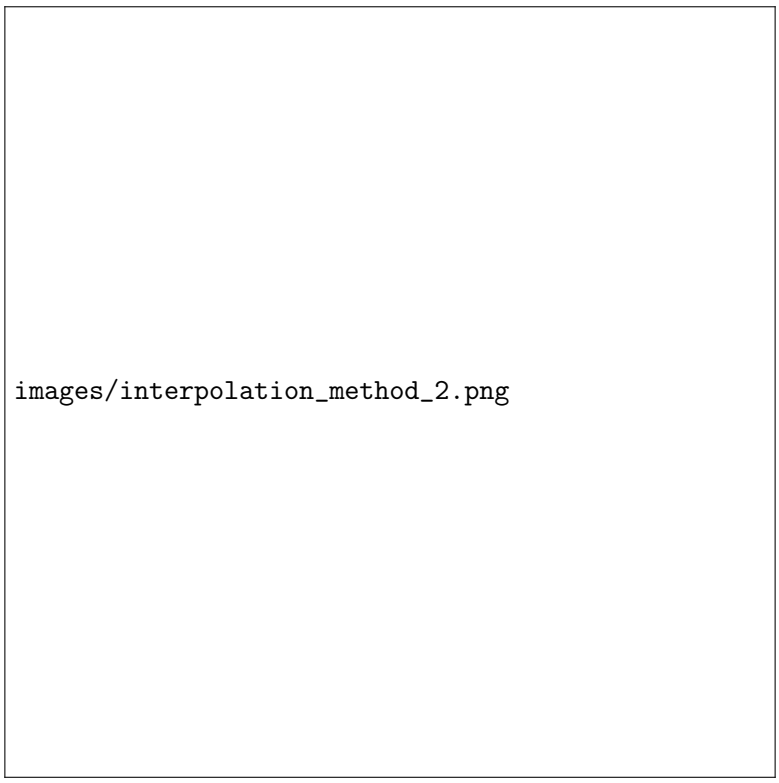


Figure 8: Example of shapes used in the second interpolation method.

This allows us to reduce the overall number of areas to predict, as the grid has a lower density compared to some of the areas in Lombardy.
For generating the grid and implementing the interpolation algorithm in this method, we can refer to the procedure described in Section **??**. By doing this, we will obtain the Lombardy region partitioned as shown in Figure **??**

(a) Areas considered in the second approach. The gray areas barred in red are the ones added to replace the municipalities without data

(b) Known $PM_{2.5}$ values for the Lombardy region

Figure 9: Lombardy region partitioned using the second method.

Note that the new areas are squares, and their rectangular shape is due to the map projection chosen (EPSG 4326 coordinate reference system).

8.2. Model training

The model training was done using the same code as in the first approach, but with the new data generated in the previous steps. The configuration used this time followed the recommended values, 10000 iterations, 200 trees, and 1000 warm-up iterations coupled with a burn-in period equal to half of the total iterations (5000) and a thinning factor of 10 to reduce autocorrelation. This new approach allowed us to use more potential of the model and to obtain better results without the restrictions of political boundaries.

8.3. Results

Using the second interpolation method described by Section **??**, we forecast the $PM_{2.5}$ values for all the squares that we created within the Lombardy region. We experimented with various date ranges and configurations within the year 2018 and 2019, considering all the covariates.
Firstly, we ran the model for dates ranging from 01/01/2018 to 31/12/2018, and we obtained the results illustrated in Figure **??**. These results were obtained using the 10-meter height and 30-degree wind adjacency matrix.

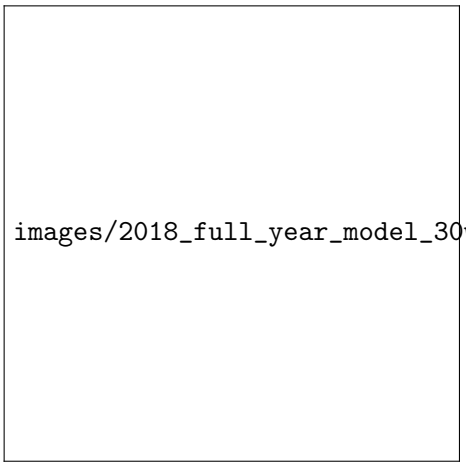


Figure 10: Predicted $PM_{2.5}$ values for 2018.

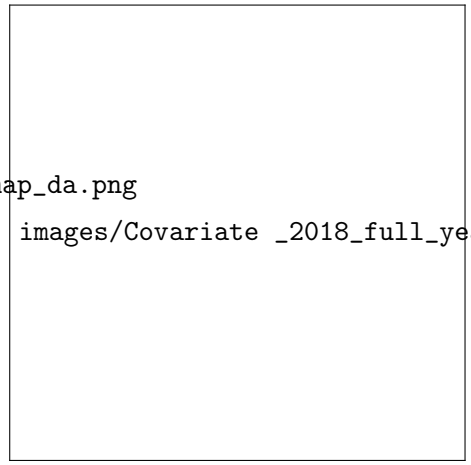


Figure 11: Frequency of covariates selection for prediction of $PM_{2.5}$ values for 2018.

Figure **??** Highlights influential covariates in the $PM_{2.5}$ concentration model. Notably, some of the most influential variables are "EM_nox_sum" and "EM_nh3_Agri_soils". These covariates indicate respectively the emissions of NO_x, and the emissions of, NH_3 across all sectors. It's also worth noticing the importance of the covariates "LA_land_use" and "LI_pigs".

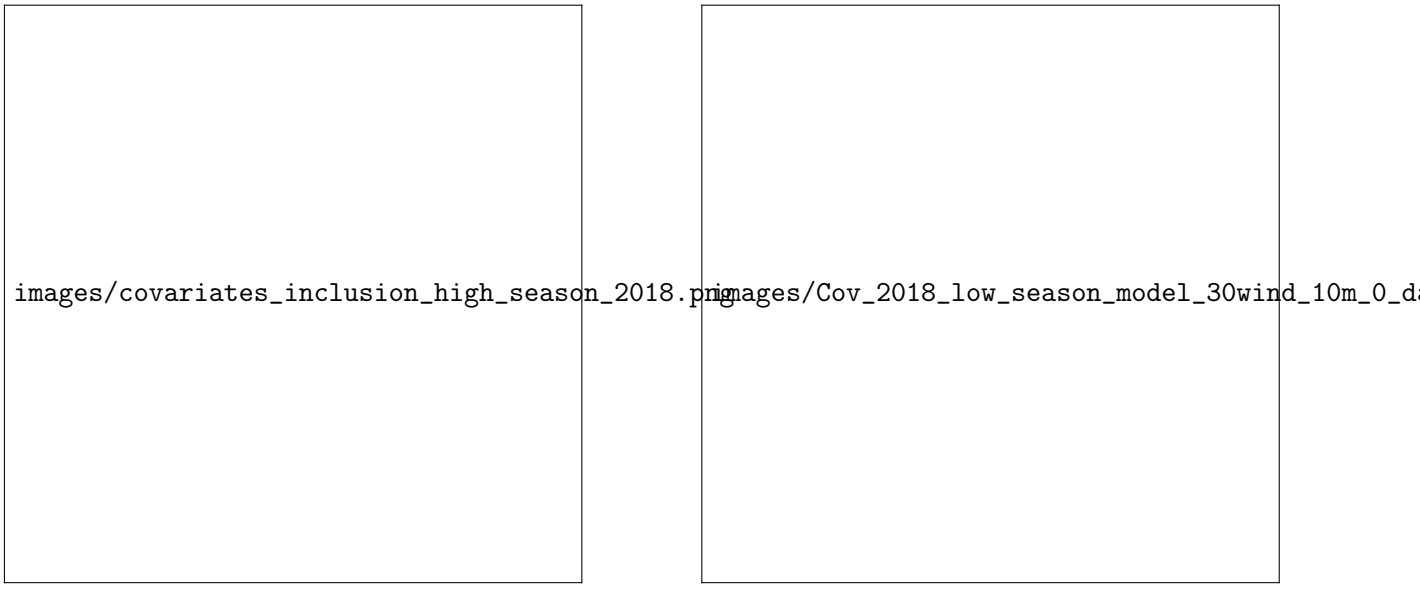
Given its lower computational requirements, this model enables us to conduct analyses on $PM_{2.5}$. In the next sections, we will explore these analyses.

8.3.1 Comparison between high and low season

In our analysis of the time series data for $PM_{2.5}$, as outlined in Section **??**, we observed significant variations in its values between the "warm" and "cold" periods. Consequently, we opted to conduct a more thorough examination of these disparities by dividing the data into two distinct periods: the high season (from 01/10/2018 to 31/03/2019) and then the low season (from 01/03/2018 to 30/09/2018). As the previous result shows, these were obtained considering the 30° and 10m height wind adjacency matrix.

(a) High season 2018 (b) Low season 2018

Figure 12: Predicted $PM_{2.5}$ values.



(a) High season 2018 (b) Low season 2018

Figure 13: Frequency of covariates selection for prediction of $PM_{2.5}$.

Figure **??** Highlights influential covariates in the $PM_{2.5}$ concentration model. Notably, some of the most influential variables depend on NH_3, also both of them are highly influenced by the covariate "WE_wind_speed_10m_max".

8.3.2 Comparison between wind adjacency matrix considering wind at 10 meters and 100 meters

In this section, we present the outcomes obtained considering in both scenario the low period of 2018 and the 30 degrees adjacency matrix. Specifically, we explore the impact of wind direction at two distinct heights: once with the predominant wind direction analyzed at 10 meters and alternatively at 100 meters.

(a) 100m (b) 10m

Figure 14: Predicted $PM_{2.5}$ values for 2018 low season.

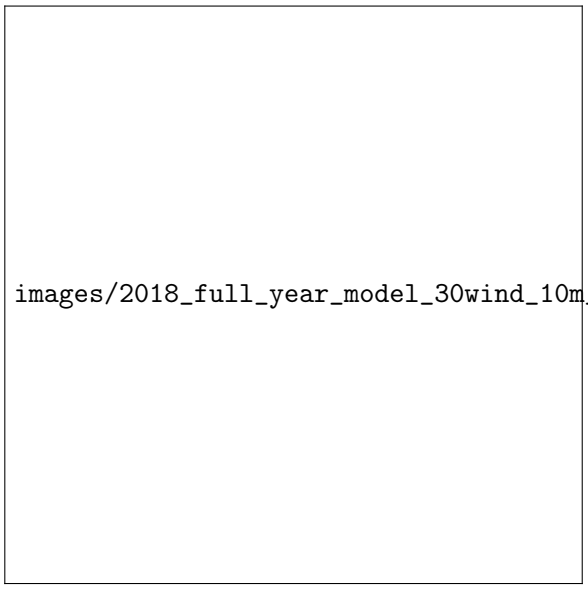
(a) 100m (b) 10m

Figure 15: Frequency of covariates selection for prediction of $PM_{2.5}$ for 2018 low season.

8.3.3 Comparison considering Milan and not considering it

Given our initial expectation of significant influence from the density of pigs on the model, the absence of such influence prompted a decision to conduct further runs without incorporating the $PM_{2.5}$ values for Milan. For this analysis, we considered the whole 2018 year.

(a) Not considering Milan



(b) Considering Milan

Figure 16: Predicted $PM_{2.5}$ values

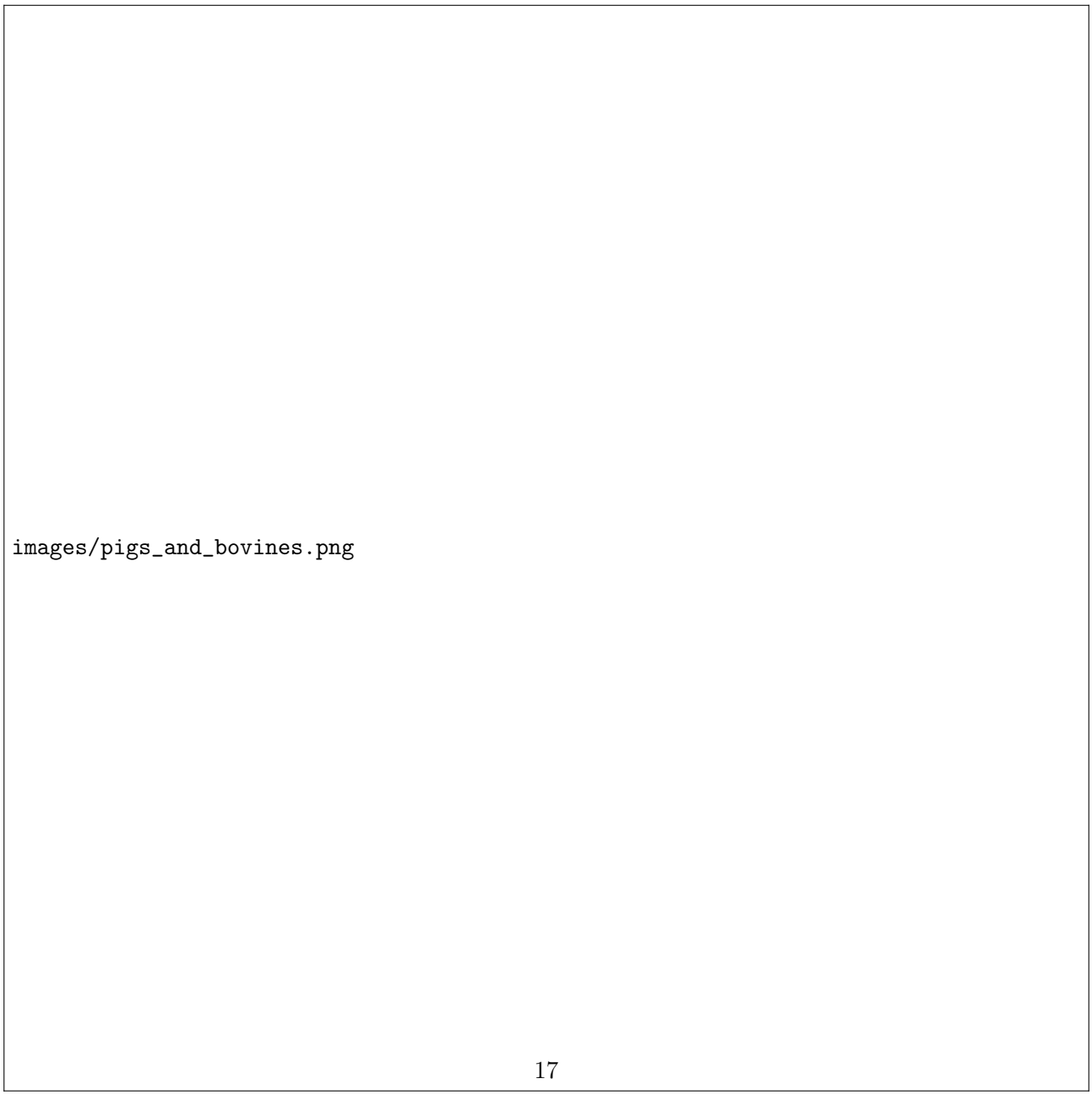


Figure 17: Distribution of swine (on the left) and bovine (on the right) in the Lombardy region

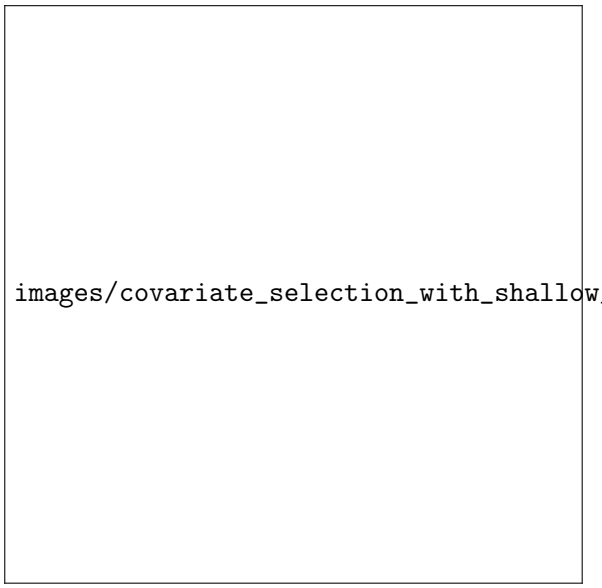
8.4. Extras and Experiments

(a) Traceplot for predictions of the first missing location (after an initial burn-in of 5000 iterations)

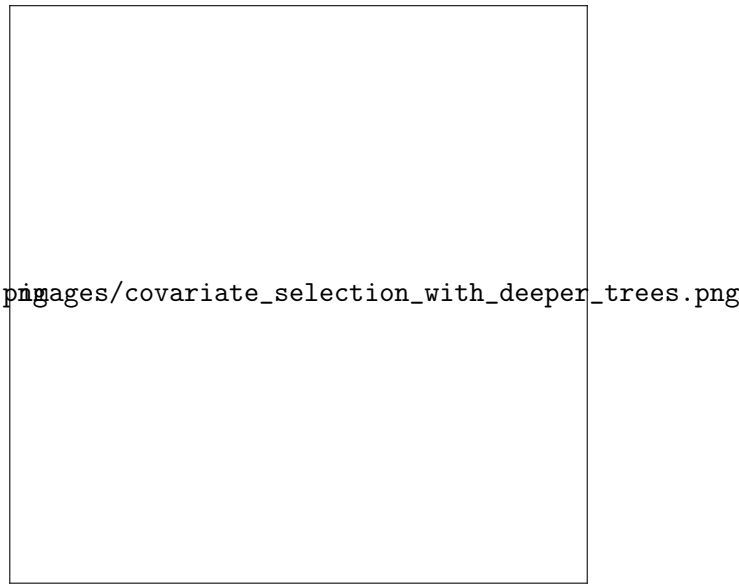
(b) Posterior density plot for the first missing location

Figure 18: The plots show a brief analysis of the MCMC samples produced for the first missing location.

To further test the model, we run a brief experiment by changing the probability enforcing shallow trees going from $T_i \sim Tree(0.95, 2)$ to $T_i \sim Tree(0.95, 1)$ ($\beta = 1$), hence preferring deeper trees. The resulting model produced interesting results regarding the covariate selection by picking with higher probability those that had already been chosen, making evident the use of the Dirichlet distribution for sparsity.



(a) Covariates selected with standard settings ($\beta = 2$)



(b) Covariates selected with deeper trees ($\beta = 1$)

Figure 19: Covariate selection with different tree parameters

9. Conclusions

In this project, we applied the Model described by Kim [**?**] to predict the $PM_{2.5}$ levels in the Lombardy region. We used two different approaches to preprocess the data and train the model. The first approach

involved making predictions for each municipality in Lombardy separately, while the second method divided the region using a grid system in the areas that don't have a station. Following these two approaches, we defined the adjacency matrix and the SIAM as described by Kim [**?**].
The results obtained from the first approach were promising, but the computational limitations of the model made it difficult to obtain more predictions with higher accuracy. The second approach allowed us to use more of the potential of the model and to obtain better results without the restrictions of political boundaries.
The results obtained from the second approach were more promising, and we were able to run, train, and predict the model in several different periods. This allowed us to observe the influence of different covariates in the model and to understand the importance of some of them.
In most results obtained from the second approach, we could observe that the predictions for $PM_{2.5}$ in the northern region of Lombardy were lower than in the southern region. This is consistent with the relief map of Lombardy, as the northern region is mountainous and has a lower population density and less agricultural and industrial activity, as shown in Figure **??** and Figure **??**.

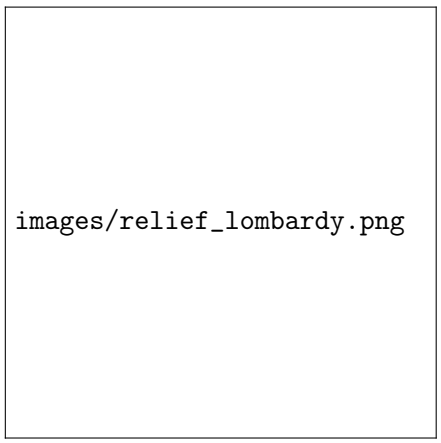


Figure 20: Relief map of Lombardy.

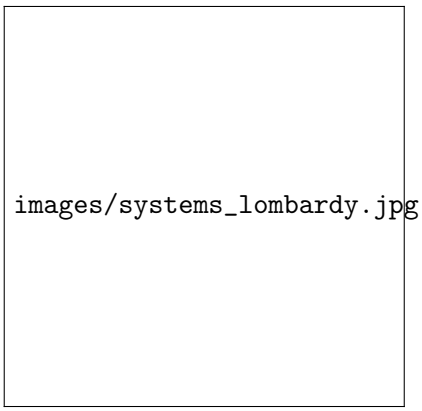


Figure 21: Agricultural and industrial systems in Lombardy [**?**]

Moreover, we can identify several zones where the $PM_{2.5}$ value is higher, such as the area around Milan and the major cities of Lombardy (Figure **??**).
Transitioning to the results of covariate selection in our model, putting together all findings across various periods yields the visualization depicted in Figure **??**.

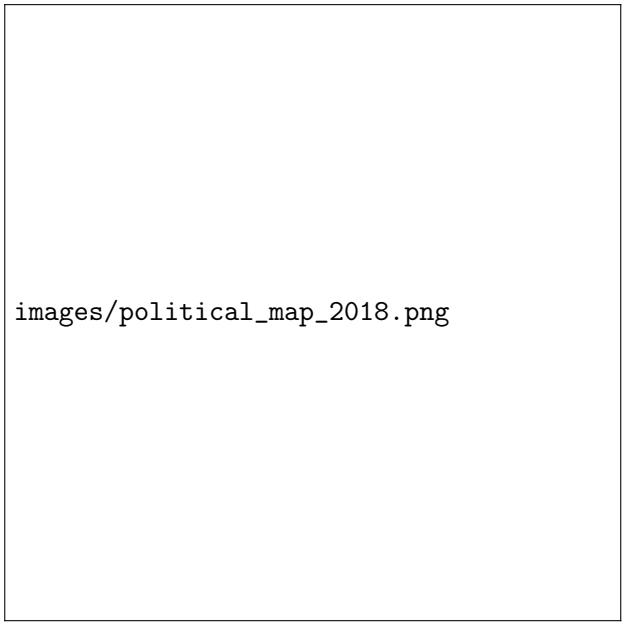


Figure 22: Cities in Lombardy with the highest PM2.5 levels.

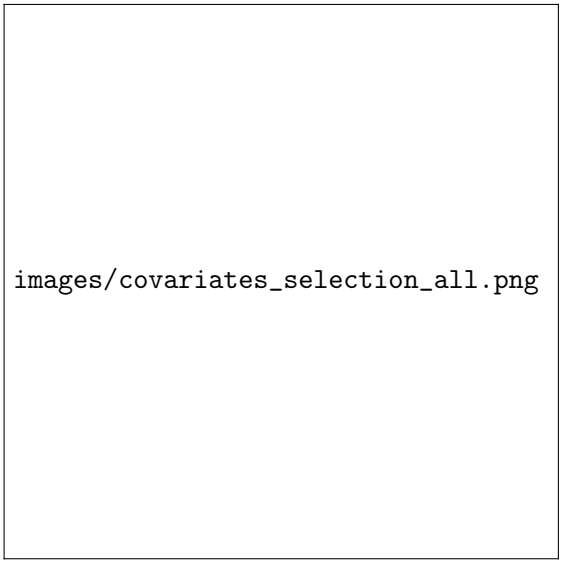


Figure 23: Frequency of covariate selection for predicting $PM_{2.5}$ concentrations across all analyzed periods.

Among the prominently selected covariates are "EM_so2_sum", "LI_pigs", "EM_nh3_livestock_mm",

"EM_nh3_agr_soils", "WE_rh_min", and "WE_wind_speed_100m_mean." These covariates correspond to the following descriptors:

- "EM_so2_sum": Total sulfur dioxide emissions.
- "LI_pigs": Number of pigs.
- "EM_nh3_livestock_mm": Ammonia emissions from livestock.
- "EM_nh3_agr_soils": Ammonia emissions from agricultural soils.
- "WE_rh_min": Minimum relative humidity.
- "WE_wind_speed_100m_mean": Mean wind speed at 100 meters.

These selected covariates align with existing literature. Notably, ammonia (NH_3) emissions from both livestock and agricultural soils have been identified as substantial contributors to fine particulate matter ($PM_{2.5}$) levels in the atmosphere, as shown in Figure **??**. Ti et al. [**?**] emphasize NH_3 emissions' detrimental effects on environmental quality and public health, advocating for agricultural NH_3 emission mitigation to reduce $PM_{2.5}$ pollution effectively. Similarly, Erisman et al. [**?**] highlight NH_3 emissions from agriculture as significant contributors to PM concentrations, underscoring the need for NH_3 reduction alongside other gas emissions to mitigate $PM_{2.5}$ pollution effectively.

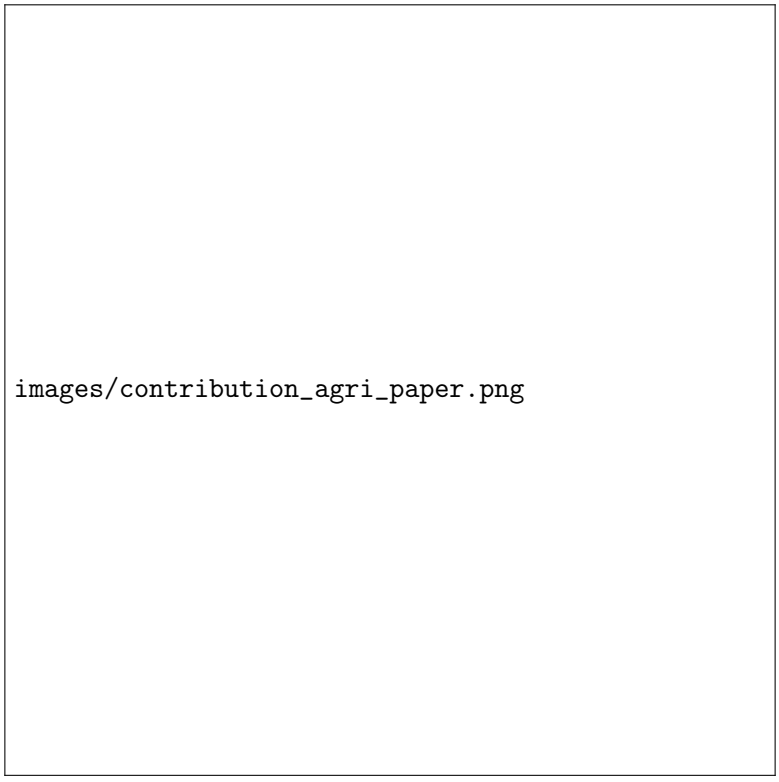


Figure 24: Contribution of agriculture to each contaminant [**?**].

Moreover, environmental factors like relative humidity (RH) and wind speed further support the inclusion of "WE_rh_min" and "WE_wind_speed_100m_mean" as covariates. Ryu et al. [**?**] demonstrate RH's role, influenced by actively transpiring plants, in enhancing smoke $PM_{2.5}$ removal efficiency, emphasizing its significance in determining PM level. Additionally, studies like the one done by Liu et al. [**?**] in Qinhuangdao (China) indicate the crucial roles of precipitation, wind direction, and speed in wet scavenging and PM dispersion.
However, it's notable that some covariates known to be important in the literature weren't selected by our model. For instance, while precipitation is recognized as pivotal in $PM_{2.5}$ removal from the atmosphere, it wasn't selected as a significant covariate in our model. Similarly, the presence of bovines, known for their contribution to NH_3 emissions, wasn't deemed significant by our model. Additionally, an also high selected covariate was "LA_lvi" which refers to low vegetation type, and information about an existing direct relationship with $PM2.5$ was not found in the literature.
Focusing on the model itself, it's clear that the model does not take into account time-series, and to overcome this limitation, the model should be optimized. Primarily, we recommend moving the implementation to a

faster and more stable programming language. This adjustment would allow us to use more potential of the model and to achieve better results without the restrictions of computational power.
In conclusion, the results obtained from the model are promising due to the consistency of the results obtained, both taking into account the current literature concerning the covariates and the known information about the Lombardy region. However, the model has limitations that need to be addressed, such as the computational limitations and the current lack of capacity to take into account time-series data.

Acknowledgements

Our thanks go to Alessandro Carminati who suggested the second approach to reduce the overall computational complexity of the algorithm and to Paolo Maranzano who suggested the interpolation method for the covariates grid.

A. Metropolis Hastings for Tree proposal

The following section details the derivation of the acceptance probabilities for the Metropolis-Hastings steps in the proposal of new tree structures following the procedure described by [**?**] and [**?**]. Consider the full conditional for the tree structures:

$$[(T_t, M_t)|\underline{R}_{-t}, \sigma] \tag{22}$$

We can obtain a draw for (**??**) by integrating out the leaf parameters M_t, thus obtaining $[T_t|\underline{R}_{-t}, \sigma]$. The marginalization is fairly simple given the choice of a conjugate normal prior for the μ_{ti}'s. We now want to build a reversible Markov Chain (MC) with stationary density π. Thus, we want to guarantee

$$\mathbb{P}(T \to T^*)\alpha(T, T^*)\pi(T|\underline{R}_{-t}, \sigma) = \mathbb{P}(T^* \to T)\alpha(T^*, T)\pi(T^*|\underline{R}_{-t}, \sigma). \tag{23}$$

Through the usual considerations, we get the following acceptance probability:

$$\alpha(T, T^*) = \min\left\{\frac{\mathbb{P}(T^* \to T)\pi(T^*|\underline{R}, \sigma)}{\mathbb{P}(T \to T^*)\pi(T|\underline{R}, \sigma)}, 1\right\}. \tag{24}$$

Note that computing the posterior for the trees is difficult, if not impossible, therefore we employ Bayes' rule:

$$\pi(T|\underline{R}, \sigma) = \frac{\mathbb{P}(\underline{R}|T, \sigma)\pi(T|\sigma)}{\mathbb{P}(\underline{R}|\sigma)} = \frac{\mathbb{P}(\underline{R}|T, \sigma)\pi(T)}{\mathbb{P}(\underline{R}|\sigma)} \tag{25}$$

We now substitute (**??**) into (**??**) and get (**??**). The computation for the acceptance ratio for each one of the three alterations can be split into:

$$\alpha(T, T^*) = \min\left\{1, \underbrace{\frac{\mathbb{P}(T^* \to T)}{\mathbb{P}(T \to T^*)}}_{\text{transition ratio}} \times \underbrace{\frac{\pi(\underline{R}|T^*, \sigma)}{\pi(\underline{R}|T, \sigma)}}_{\text{likelihood ratio}} \times \underbrace{\frac{\pi(T^*)}{\pi(T)}}_{\text{structure ratio}}\right\}, \tag{26}$$

where each ratio depends on the type of alteration. The ratios will be described in further detail in the following sections.

A.1. GROW alteration

With the GROW alteration, we choose a terminal node to expand and we grow it into two new terminal nodes. The expanded node will then become a singly-internal node, thus it will be associated with a splitting rule.
Commencing with the transition ratio, this signifies the proportion of the likelihood of transitioning from the pre-existing tree structure T to the updated one T^*, against the probability of the reverse scenario occurring. We denote these probabilities as $\mathbb{P}(T \to T^*)$ and $\mathbb{P}(T^* \to T)$ respectively. Thus, we write

$$\begin{aligned}\mathbb{P}(T \to T^*) =& \mathbb{P}(GROW) \times \mathbb{P}(\text{select node } \eta \text{ to grow}) \times \\ & \mathbb{P}(\text{select the } j\text{th predictor as splitting variable}) \times \\ & \mathbb{P}(\text{select the } i\text{th value of } x_j \text{ as splitting constant}) \\ =& \rho_{GROW}\frac{1}{b}\frac{1}{p(\eta)}\frac{1}{n_j(\eta)},\end{aligned} \tag{27}$$

where b is the number of terminal nodes for T, ρ_{GROW} is the probability of choosing the GROW alteration, $p(\eta)$ is the number of predictors still available to split on, $n_j(\eta)$ is the number of *unique* values left for the chosen predictor, accounting for previous splits.
The opposite operation is equivalent to a PRUNE alteration, therefore we write:

$$\mathbb{P}(T^* \to T) = \mathbb{P}(PRUNE) \times \mathbb{P}(\text{select node } \eta \text{ to prune}) = \rho_{PRUNE}\frac{1}{w^*}, \tag{28}$$

where w^* is the number of singly-internal nodes for T^* and ρ_{PRUNE} is the probability of choosing the PRUNE alteration.
The resulting transition ratio is:

$$\frac{\mathbb{P}(T^* \to T)}{\mathbb{P}(T \to T^*)} = \frac{\rho_{PRUNE}}{\rho_{GROW}}\frac{b}{w^*}p(\eta)n_j(\eta) \tag{29}$$

Let us now consider the likelihood ratio for the GROW alteration. Considering the model (**??**), the likelihood is determined by how the responses fall into the b_t terminal nodes of the tree structure. Thus,

$$\mathbb{P}(R_1, ..., R_n|T, \sigma) \overset{ind}{=} \prod_{i=1}^{b} \mathbb{P}(R_{i_1}, ..., R_{i_{n_i}}|\sigma), \tag{30}$$

where the R_i's denote the residuals in the ith terminal node, and n_i is the number of observations in the ith terminal node such that $n = \sum_{i=1}^{b} n_i$. Given that $R_{i_1}, ..., R_{i_{n_i}}|\mu_i, \sigma \overset{iid}{\sim} \mathcal{N}(\mu_i, \sigma^2)$, we integrate out μ_i for the reasons described by *Chipman et al. (page 275)*[**?**]. Remembering that a priori $\mu \sim \mathcal{N}(0, \sigma_\mu^2)$,

$$\begin{aligned}\mathbb{P}(R_{i_1}, ..., R_{i_{n_i}}|\sigma) &= \int_{\mathbb{R}} \mathbb{P}(R_{i_1}, ..., R_{i_{n_1}}, d\mu_i|\sigma) \\ &= \int_{\mathbb{R}} \mathbb{P}(R_{i_1}, ..., R_{i_{n_1}}|\mu_i, \sigma)\pi(\mu_i|\sigma)d\mu_i\end{aligned} \tag{31}$$

We can prove via completion of the square that the resulting likelihood is:

$$\begin{aligned}\mathbb{P}(R_{i_1}, ..., R_{i_{n_i}}|\sigma) =& \frac{1}{(2\pi\sigma^2)^{n_i/2}} \sqrt{\frac{\sigma^2}{\sigma^2 + n_i\sigma_\mu^2}} \times \\ & \exp\left\{ -\frac{1}{2\sigma^2} \left(\sum_{i=1}^{n_i} (R_{i_1} - \bar{R}_i)^2 - \frac{\bar{R_i}^2 n_i^2}{n_i + \frac{\sigma^2}{\sigma_\mu^2}} + n_i \bar{R_i}^2 \right) \right\}\end{aligned} \tag{32}$$

The new proposed tree T^* differs from T only by the number of terminal nodes and the grown node. We denote with i_L and i_R the new terminal nodes on the left and right respectively. The resulting likelihood ratio is:

$$\begin{aligned}&\frac{\mathbb{P}(\underline{R}|T^*, \sigma)}{\mathbb{P}(\underline{R}|T, \sigma)} = \frac{\mathbb{P}(R_{i_{L,1}}, ..., R_{i_{L,n_{i,L}}}|\sigma)\mathbb{P}(R_{i_{R,1}}, ..., R_{i_{R,n_{i,R}}}|\sigma)}{\mathbb{P}(R_{i_1}, ..., R_{i_{n_i}}|\sigma)} \\ =& \sqrt{\frac{\sigma^2(\sigma^2 + n_i\sigma_\mu^2)}{(\sigma^2 + n_{i,L}\sigma_\mu^2)(\sigma^2 + n_{i,R}\sigma_\mu^2)}} \exp\left\{ \frac{\sigma_\mu^2}{2\sigma^2} \left(\frac{\left(\sum_{j=1}^{n_{i,L}} R_{i_{L,j}}\right)^2}{(\sigma^2 + n_{i,L}\sigma_\mu^2)} + \frac{\left(\sum_{j=1}^{n_{i,R}} R_{i_{R,j}}\right)^2}{(\sigma^2 + n_{i,R}\sigma_\mu^2)} - \frac{\left(\sum_{j=1}^{n_i} R_{i_j}\right)^2}{(\sigma^2 + n_i\sigma_\mu^2)} \right) \right\}\end{aligned} \tag{33}$$

Finally, for the structure ratio, we have the following:

$$\pi(T) = \prod_{\eta \in H_T} (1 - \mathbb{P}_{SPLIT}(\eta)) \prod_{\eta \in H_I} \mathbb{P}_{SPLIT}(\eta) \prod_{\eta \in H_I} \mathbb{P}_{RULE}(\eta), \tag{34}$$

where H_T is the set of terminal nodes, and H_I is the set of internal nodes. The probability of splitting is $\mathbb{P}_{SPLIT}(\eta) = \alpha/(1 + d_\eta)^\beta$, and the probability of picking a rule (as per the transition ratio) is $\mathbb{P}_{RULE}(\eta) = 1/p(\eta) \times 1/n_j(\eta)$ (probability of picking a covariate $\times$ probability of picking a value for the covariate).
The original tree differs from the new tree by the grown node η and the two new terminal nodes: $H_T^* = H_T \cup \{\eta_L, \eta_R\} \setminus \{\eta\}$, $H_I^* = H_I \cup \{\eta\}$, thus the ratio is:

$$\begin{aligned}\frac{\pi(T^*)}{\pi(T)} =& \frac{\prod_{\eta \in H_T^*}(1 - \mathbb{P}_{SPLIT}(\eta)) \prod_{\eta \in H_I^*} \mathbb{P}_{SPLIT}(\eta) \prod_{\eta \in H_I^*} \mathbb{P}_{RULE}(\eta)}{\prod_{\eta \in H_T}(1 - \mathbb{P}_{SPLIT}(\eta)) \prod_{\eta \in H_I} \mathbb{P}_{SPLIT}(\eta) \prod_{\eta \in H_I} \mathbb{P}_{RULE}(\eta)} \\ =& \frac{(1 - \mathbb{P}_{SPLIT}(\eta_L))(1 - \mathbb{P}_{SPLIT}(\eta_R))\mathbb{P}_{SPLIT}(\eta)\mathbb{P}_{RULE}(\eta)}{(1 - \mathbb{P}_{SPLIT}(\eta))} \\ =& \frac{\left(1 - \frac{\alpha}{(1+d_{\eta_L})^\beta}\right)\left(1 - \frac{\alpha}{(1+d_{\eta_R})^\beta}\right)\frac{\alpha}{(1+d_\eta)^\beta}\frac{1}{p(\eta)}\frac{1}{n_j(\eta)}}{\left(1 - \frac{\alpha}{(1+d_\eta)^\beta}\right)} \\ =& \alpha \frac{\left(1 - \frac{\alpha}{(2+d_\eta)^\beta}\right)^2}{((1 + d_\eta)^\beta - \alpha)\, p(\eta) n_j(\eta)}\end{aligned} \tag{35}$$

The last step can be derived considering that $d_{\eta_L} = d_{\eta_R} = d_\eta + 1$, this is because the new terminal nodes will be at the depth of the parent (the grown node η), plus one.

Thus, the full GROW alteration proposal ratio is given by:

$$\begin{aligned} r_{GROW} =& \frac{\rho_{PRUNE}}{\rho_{GROW}} \frac{b}{w^*} p(\eta) n_j(\eta) \times \\ & \sqrt{\frac{\sigma^2(\sigma^2 + n_i\sigma_\mu^2)}{(\sigma^2 + n_{i,L}\sigma_\mu^2)(\sigma^2 + n_{i,R}\sigma_\mu^2)}} \times \\ & \exp\left\{ \frac{\sigma_\mu^2}{2\sigma^2} \left(\frac{\left(\sum_{j=1}^{n_{i,L}} R_{i_{L,j}}\right)^2}{(\sigma^2 + n_{i,L}\sigma_\mu^2)} + \frac{\left(\sum_{j=1}^{n_{i,R}} R_{i_{R,j}}\right)^2}{(\sigma^2 + n_{i,R}\sigma_\mu^2)} - \frac{\left(\sum_{j=1}^{n_i} R_{i_j}\right)^2}{(\sigma^2 + n_i\sigma_\mu^2)} \right) \right\} \times \\ & \alpha \frac{\left(1 - \frac{\alpha}{(2+d_\eta)^\beta}\right)^2}{\left((1+d_\eta)^\beta - \alpha\right) p(\eta) n_j(\eta)} \end{aligned} \tag{36}$$

Note that the probability of picking a covariate and one of its values can be simplified.

A.2. PRUNE alteration

The PRUNE alteration represents the opposite alteration to GROW. PRUNE selects a singly-internal node, removes both children and transforms the node itself into a terminal node.
As for the GROW alteration, let's start with the transition ratio:

$$\mathbb{P}(T \to T^*) = \mathbb{P}(PRUNE)\mathbb{P}(\text{select } \eta \text{ to prune}) = \rho_{PRUNE} \frac{1}{w}, \tag{37}$$

where w is the number of singly-internal nodes for the old tree structure. The opposite operation is equivalent to a GROW alteration as described for equation (**??**). The difference in this transition is that the old tree has $b-1$ terminal nodes, thus the transition probability is:

$$\mathbb{P}(T \to T^*) = \rho_{GROW} \frac{1}{b-1} \frac{1}{p(\eta^*)} \frac{1}{n_j(\eta^*)}. \tag{38}$$

The transition ratio is:

$$\frac{\mathbb{P}(T^* \to T)}{\mathbb{P}(T \to T^*)} = \frac{\rho_{GROW}}{\rho_{PRUNE}} \frac{w}{b-1} \frac{1}{p(\eta^*) n_j(\eta^*)} \tag{39}$$

The likelihood ratio is nothing more than the inverse of the likelihood ratio for the GROW alteration in equation (**??**).

$$\begin{aligned} \frac{\mathbb{P}(\underline{R}|T^*, \sigma)}{\mathbb{P}(\underline{R}|T, \sigma)} =& \frac{\mathbb{P}(R_{i_1}, ..., R_{i_{n_i}}|\sigma)}{\mathbb{P}(R_{i_{L,1}}, ..., R_{i_{L,n_{i,L}}}|\sigma)\mathbb{P}(R_{i_{R,1}}, ..., R_{i_{R,n_{i,R}}}|\sigma)} \\ =& \sqrt{\frac{(\sigma^2 + n_{i,L}\sigma_\mu^2)(\sigma^2 + n_{i,R}\sigma_\mu^2)}{\sigma^2(\sigma^2 + n_i\sigma_\mu^2)}} \exp\left\{ -\frac{\sigma_\mu^2}{2\sigma^2} \left(\frac{\left(\sum_{j=1}^{n_{i,L}} R_{i_{L,j}}\right)^2}{(\sigma^2 + n_{i,L}\sigma_\mu^2)} + \frac{\left(\sum_{j=1}^{n_{i,R}} R_{i_{R,j}}\right)^2}{(\sigma^2 + n_{i,R}\sigma_\mu^2)} - \frac{\left(\sum_{j=1}^{n_i} R_{i_j}\right)^2}{(\sigma^2 + n_i\sigma_\mu^2)} \right) \right\} \end{aligned} \tag{40}$$

This is also true for the structure ratio:

$$\frac{\pi(T^*)}{\pi(T)} = \frac{1}{\alpha} \frac{\left((1+d_\eta)^\beta - \alpha\right) p(\eta^*) n_j(\eta^*)}{\left(1 - \frac{\alpha}{(2+d_\eta)^\beta}\right)^2} \tag{41}$$

The complete ratio is thus given by:

$$\begin{aligned} r_{PRUNE} =& \frac{\rho_{GROW}}{\rho_{PRUNE}} \frac{w}{b-1} \frac{1}{p(\eta^*) n_j(\eta^*)} \times \\ & \sqrt{\frac{(\sigma^2 + n_{i,L}\sigma_\mu^2)(\sigma^2 + n_{i,R}\sigma_\mu^2)}{\sigma^2(\sigma^2 + n_i\sigma_\mu^2)}} \times \\ & \exp\left\{ -\frac{\sigma_\mu^2}{2\sigma^2} \left(\frac{\left(\sum_{j=1}^{n_{i,L}} R_{i_{L,j}}\right)^2}{(\sigma^2 + n_{i,L}\sigma_\mu^2)} + \frac{\left(\sum_{j=1}^{n_{i,R}} R_{i_{R,j}}\right)^2}{(\sigma^2 + n_{i,R}\sigma_\mu^2)} - \frac{\left(\sum_{j=1}^{n_i} R_{i_j}\right)^2}{(\sigma^2 + n_i\sigma_\mu^2)} \right) \right\} \times \\ & \frac{1}{\alpha} \frac{\left((1+d_\eta)^\beta - \alpha\right) p(\eta^*) n_j(\eta^*)}{\left(1 - \frac{\alpha}{(2+d_\eta)^\beta}\right)^2} \end{aligned} \tag{42}$$

Once again, the probability of picking a covariate and one of its values can be simplified.

A.3. CHANGE alteration

The CHANGE alteration picks an internal node and changes its rule by picking a new predictor and a new value among the available ones for the chosen predictor. In this particular case, we consider only the singly-internal nodes, although it can easily be extended to all internal nodes.
The transition probability is as follows:

$$\begin{aligned}\mathbb{P}(T \to T^*) =& \mathbb{P}(CHANGE)\mathbb{P}(\text{select node } \eta \text{ to change})\times \\ & \mathbb{P}(\text{select new covariate as splitting variable})\times \\ & \mathbb{P}(\text{select value for covariate as splitting constant}) \\ =& \rho_{CHANGE}\frac{1}{w}\frac{1}{p(\eta)n_j(\eta)}\end{aligned} \tag{43}$$

Similarly, the opposite is:

$$\mathbb{P}(T^* \to T) = \rho_{CHANGE}\frac{1}{w^*}\frac{1}{p(\eta^*)n_j(\eta^*)} \tag{44}$$

However, the number of singly-internal nodes and available covariates remains unchanged, therefore the transition ratio is simply:

$$\frac{\mathbb{P}(T^* \to T)}{\mathbb{P}(T \to T^*)} = \frac{n_j(\eta)}{n_j(\eta^*)} \tag{45}$$

For the likelihood ratio, the new tree differs from the original in the child nodes and the split rule. However, the likelihood is effected only by the new child nodes, thus:

$$\begin{aligned}\frac{\mathbb{P}(\underline{R}|T^*,\sigma)}{\mathbb{P}(\underline{R}|T,\sigma)} =& \frac{\mathbb{P}(R_{1^*,1},...,R_{1^*,n_{1^*}}|\sigma)\mathbb{P}(R_{2^*,1},...,R_{2^*,n_{2^*}}|\sigma)}{\mathbb{P}(R_{1,1},...,R_{1,n_1}|\sigma)\mathbb{P}(R_{2,1},...,R_{2,n_2}|\sigma)} \\ =& \sqrt{\frac{(\sigma^2+n_1\sigma_\mu^2)(\sigma^2+n_2\sigma_\mu^2)}{(\sigma^2+n_1^*\sigma_\mu^2)(\sigma^2+n_2^*\sigma_\mu^2)}}\times \\ & \exp\left\{\frac{\sigma_\mu^2}{2\sigma^2}\left(\frac{(\sum_{j=1}^{n_1^*}R_{1^*,j})^2}{n_1^*\sigma_\mu^2+\sigma^2}+\frac{(\sum_{j=1}^{n_2^*}R_{2^*,j})^2}{n_2^*\sigma_\mu^2+\sigma^2}-\frac{(\sum_{j=1}^{n_1}R_{1,j})^2}{n_1\sigma_\mu^2+\sigma^2}-\frac{(\sum_{j=1}^{n_2}R_{2,j})^2}{n_2\sigma_\mu^2+\sigma^2}\right)\right\},\end{aligned} \tag{46}$$

where we denote with $R_1,\cdot$ the residuals in the first child node and with $R_2,\cdot$ the residuals in the second child node. Similarly, we denote with n_1 and n_2 the number of observations in the two nodes. Lastly, the tree structure ratio is given by:

$$\begin{aligned}\frac{\pi(T^*)}{\pi(T)} =& \frac{(1-\mathbb{P}_{SPLIT}(\eta_1^*))(1-\mathbb{P}_{SPLIT}(\eta_2^*))\mathbb{P}_{SPLIT}(\eta^*)\mathbb{P}_{RULE}(\eta^*)}{(1-\mathbb{P}_{SPLIT}(\eta_1))(1-\mathbb{P}_{SPLIT}(\eta_2))\mathbb{P}_{SPLIT}(\eta)\mathbb{P}_{RULE}(\eta)} \\ =& \frac{n_j(\eta)}{n_j(\eta^*)},\end{aligned} \tag{47}$$

this is because the depth of the tree does not change. Also, note that the structure ratio is the opposite of the transition ratio, thus we are left with only the likelihood ratio:

$$\begin{aligned}r_{CHANGE} =& \sqrt{\frac{(\sigma^2+n_1\sigma_\mu^2)(\sigma^2+n_2\sigma_\mu^2)}{(\sigma^2+n_1^*\sigma_\mu^2)(\sigma^2+n_2^*\sigma_\mu^2)}}\times \\ & \exp\left\{\frac{\sigma_\mu^2}{2\sigma^2}\left(\frac{(\sum_{j=1}^{n_1^*}R_{1^*,j})^2}{n_1^*\sigma_\mu^2+\sigma^2}+\frac{(\sum_{j=1}^{n_2^*}R_{2^*,j})^2}{n_2^*\sigma_\mu^2+\sigma^2}-\frac{(\sum_{j=1}^{n_1}R_{1,j})^2}{n_1\sigma_\mu^2+\sigma^2}-\frac{(\sum_{j=1}^{n_2}R_{2,j})^2}{n_2\sigma_\mu^2+\sigma^2}\right)\right\}\end{aligned} \tag{48}$$

B. Computation of the wind adjacency matrix

The Structurally Informed Adjacency Matrix requires a particular adjacency matrix based on wind direction. In the paper, Kim proposed a couple of ways to calculate the matrix and we decided to use the same one he used in the example at the end of the paper.
Specifically, we consider two municipalities adjacent if their centroids are within a certain angle parameter to the left and right of the prevailing wind direction. Following is the algorithm we developed to calculate the wind adjacency matrix.

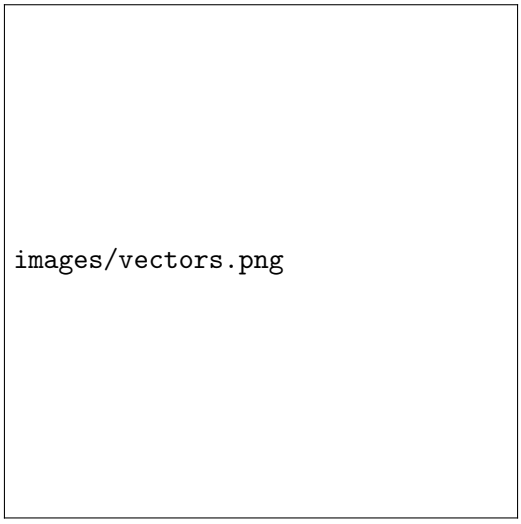


Figure 25: Diagram of the vectors used in the algorithm

Let $\mathbf{c} = (c_x, c_y)$ be a point on the map, $\mathbf{w} = (w_x, w_y)$ the prevailing wind vector in that point and θ the angle parameter. We can define two other vectors $\mathbf{u}$ and $\mathbf{v}$ (displayed in *Figure* **??**) in this way:

$$\mathbf{u} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} (-\mathbf{w}) = \begin{pmatrix} -\cos\theta & \sin\theta \\ -\sin\theta & -\cos\theta \end{pmatrix} \mathbf{w} = \begin{pmatrix} -w_x\cos\theta + w_y\sin\theta \\ -w_x\sin\theta - w_y\cos\theta \end{pmatrix} \tag{49}$$

$$\mathbf{v} = \begin{pmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{pmatrix} (-\mathbf{w}) = \begin{pmatrix} -\cos\theta & -\sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix} \mathbf{w} = \begin{pmatrix} -w_x\cos\theta - w_y\sin\theta \\ +w_x\sin\theta - w_y\cos\theta \end{pmatrix} \tag{50}$$

Note that the matrix $\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ is the *transformation matrix* that rotates the points in the plane counterclockwise through an angle of θ about the origin.

With these two vectors ($\mathbf{u}$ and $\mathbf{v}$) we can define the points in the shaded area shown in orange in *Figure* **??**. A point $\mathbf{d}$ is adjacent to $\mathbf{c}$ (is in the shaded area) if there exist two **non-negative** real numbers s and t such that:

$$\mathbf{d} = \mathbf{c} + s\mathbf{u} + t\mathbf{v} \tag{51}$$

If we define $\mathbf{d}' = \mathbf{d} - \mathbf{c} = (x, y)$, we can rewrite (**??**) as a 2×2 linear system.

$$\begin{pmatrix} -w_x\cos\theta + w_y\sin\theta & -w_x\cos\theta - w_y\sin\theta \\ -w_x\sin\theta - w_y\cos\theta & +w_x\sin\theta - w_y\cos\theta \end{pmatrix} \begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} \tag{52}$$

Notice that as long as θ is between 0 and $\pi/2$ both excluded, then u and v are linearly independent and so the system admits a unique solution.

$$\begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} -w_x\cos\theta + w_y\sin\theta & -w_x\cos\theta - w_y\sin\theta \\ -w_x\sin\theta - w_y\cos\theta & +w_x\sin\theta - w_y\cos\theta \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \end{pmatrix} \tag{53}$$

The matrix is always invertible and so we can expand (**??**) in a more explicit form.

$$\begin{pmatrix} s \\ t \end{pmatrix} = \frac{1}{-(w_x^2 + w_y^2)\sin(2\theta)} \begin{pmatrix} w_x\sin\theta - w_y\cos\theta & w_x\cos\theta + w_y\sin\theta \\ w_x\sin\theta + w_y\cos\theta & w_y\sin\theta - w_x\cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{54}$$

Finally, we can check if the components of the solution (s and t) are all non negative. In that case, $\mathbf{d}$ is in the shaded region and therefore considered adjacent to $\mathbf{c}$, otherwise the two points are not adjacent.

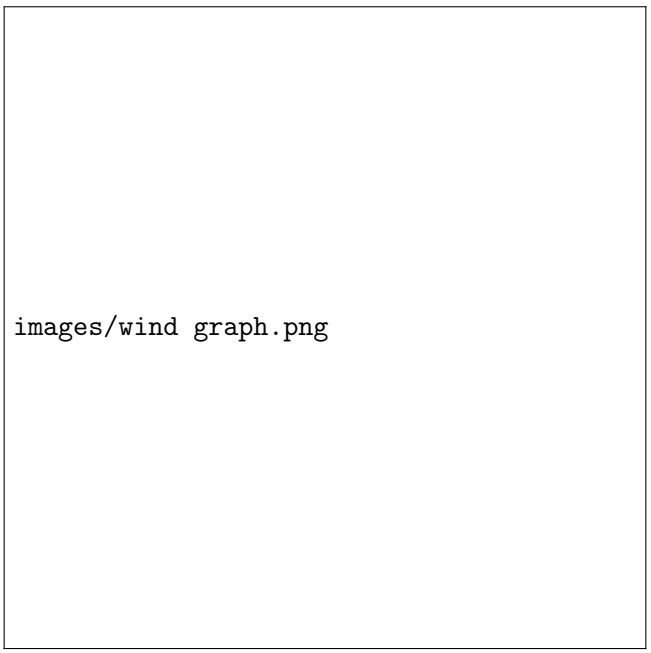


Figure 26: Example: here all the municipalities that are considered adjacent to Cremona are colored in red.

C. AgrImOnIA dataset

C.1. AGC_Dataset_v_3_0_0

Column	Description
Latitude	Latitude of the measurement.
Longitude	Longitude of the measurement.
Time	Time of the measurement.
Altitude	Altitude of the measurement.
WE_temp_2m	Temperature at 2 meters.
WE_wind_speed_10m_mean	Wind speed at 10 meters.
WE_wind_speed_10m_max	Maximum wind speed at 10 meters.
WE_mode_wind_direction_10m	Mode wind direction at 10 meters.
WE_tot_precipitation	Total precipitation.
WE_precipitation_t	Precipitation type.
WE_surface_pressure	Surface pressure.
WE_solar_radiation	Solar radiation.
WE_rh_min	Minimum relative humidity.
WE_rh_mean	Mean relative humidity.
WE_rh_max	Maximum relative humidity.
WE_wind_speed_100m_mean	Wind speed at 100 meters.
WE_wind_speed_100m_max	Maximum wind speed at 100 meters.
WE_mode_wind_direction_100m	Mode wind direction at 100 meters.
WE_blh_layer_max	Maximum boundary layer height.
WE_blh_layer_min	Minimum boundary layer height.
EM_nh3_livestock_mm	Ammonia emissions from livestock.
EM_nh3_agr_soils	Ammonia emissions from agricultural soils.
EM_nh3_agr_waste_burn	Ammonia emissions from agricultural waste burning.
EM_nh3_sum	Total ammonia emissions.
EM_nox_traffic	Nitrogen oxides emissions from traffic.
EM_nox_sum	Total nitrogen oxides emissions.
EM_so2_sum	Total sulfur dioxide emissions.
LI_pigs	Number of pigs.
LI_bovine	Number of bovines.
LA_hvi	One-half of the total green leaf area for high vegetation type.
LA_lvi	One-half of the total green leaf area for low vegetation type.
LA_land_use	Land use.

Table 1: Columns of the AgrImOnIA dataset (AGC_Dataset_v_3_0_0.csv).

C.2. Agrimonia_Dataset_v_3_0_0.csv

Column	Description
IDStations	Station ID.
Latitude	Latitude of the station.
Longitude	Longitude of the station.
Time	Time of the measurement.
Altitude	Altitude of the station.
AQ_pm10	Particulate matter 10.
AQ_pm25	Particulate matter 2.5.
AQ_co	Carbon monoxide.
AQ_nh3	Ammonia.
AQ_nox	Nitrogen oxides.
AQ_no2	Nitrogen dioxide.
AQ_so2	Sulfur dioxide.
WE_temp_2m	Temperature at 2 meters.
WE_wind_speed_10m_mean	Wind speed at 10 meters.
WE_wind_speed_10m_max	Maximum wind speed at 10 meters.
WE_mode_wind_direction_10m	Mode wind direction at 10 meters.
WE_tot_precipitation	Total precipitation.
WE_precipitation_t	Precipitation type.
WE_surface_pressure	Surface pressure.
WE_solar_radiation	Solar radiation.
WE_rh_min	Minimum relative humidity.
WE_rh_mean	Mean relative humidity.
WE_rh_max	Maximum relative humidity.
WE_wind_speed_100m_mean	Wind speed at 100 meters.
WE_wind_speed_100m_max	Maximum wind speed at 100 meters.
WE_mode_wind_direction_100m	Mode wind direction at 100 meters.
WE_blh_layer_max	Maximum boundary layer height.
WE_blh_layer_min	Minimum boundary layer height.
EM_nh3_livestock_mm	Ammonia emissions from livestock.
EM_nh3_agr_soils	Ammonia emissions from agricultural soils.
EM_nh3_agr_waste_burn	Ammonia emissions from agricultural waste burning.
EM_nh3_sum	Total ammonia emissions.
EM_nox_traffic	Nitrogen oxides emissions from traffic.
EM_nox_sum	Total nitrogen oxides emissions.
EM_so2_sum	Total sulfur dioxide emissions.
LI_pigs	Number of pigs.
LI_bovine	Number of bovines.
LI_pigs_v2	Number of pigs.
LI_bovine_v2	Number of bovines.
LA_hvi	One-half of the total green leaf area for high vegetation type.
LA_lvi	One-half of the total green leaf area for low vegetation type.
LA_land_use	Land use.
LA_soil_use	Soil use.

Table 2: Columns of the AgrImOnIA dataset (Agrimonia_Dataset_v_3_0_0.csv).