Cross-Validation for spatial data

Cristina Chavez-Chong

Chapter 1: Cross-Validation for spatial data

Cristina Olimpia Chavez Chong

June 2024

In this chapter, we delve into the topic of spatial cross-validation, a well known method for model assessment and parameter selection. We begin by providing key definitions of classical cross-validation (CV) using the regression framework, setting the foundation for understanding the subsequent discussions. We also conduct a literature review to explore the main spatial cross-validation methods proposed in the field. Furthermore, we highlight the main drawbacks associated with each method, shedding light on the challenges and considerations researchers face when implementing spatial cross-validation techniques and serving as motivation and starting point for the rest of the thesis.

1 Statistical framework

Let us recall some basics of cross-validation. We consider some sample $\mathcal{D}_n = \{(\mathbf{X}_i, Y_i),\ i = 1, \ldots, n\}$, with $(\mathbf{X}_i, Y_i)$ i.i.d., where for $i = 1$ to n, $\mathbf{X}_i \in \mathbb{R}^p$ $(X_{1,i}, \ldots, X_{p,i})$ and $Y_i \in \mathbb{R}$ is the variable of interest. Under the general regression setting we write ,

$$Y_i = f(\mathbf{X}_i) + \varepsilon_i \tag{1}$$

Here f is an unknown regression function and ε_i is the error term. The goal is to estimate the function f and assess the goodness of fit; if we are interested in forecasting, we wonder how reliable are the predictions and it is essential to assess the risk of the estimator leading to these predictions. This is achieved by minimizing the predictive risk of f given a certain loss function (or cost function) l,

$$\mathcal{R}(f) = \mathbb{E}[l(f(\mathbf{X}), \mathbf{Y}^{(1)})] \tag{2}$$

where $\mathbf{Y}^{(1)}$ is an independent copy of $\mathbf{Y}$.

From now, let us consider the framework of linear regression and write again (1) as

$$Y_i = \mathbf{X}_i\boldsymbol{\beta} + \varepsilon_i,\ i = 1, \ldots, n \tag{3}$$

where $\boldsymbol{\beta}$ is a $(p+1)$-dimensional vector of parameters to be estimated. Using matrix expression, equation (3) can be written as,

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{4}$$

where X_i is a vector of p covariates and the constant, $\mathbf{Y} = (Y_1, \ldots, Y_n)^{\mathrm{T}}$, we will denote by A^{T} the transpose of a matrix from now on, $\mathbf{X}$ is a $n \times (p+1)$ matrix and $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^{\mathrm{T}}$.

Typically l is the quadratic loss function and the risk (2) is defined by

$$\mathcal{R}(\boldsymbol{\beta}) = \mathbb{E}[\|\mathbf{Y}^{(1)} - \mathbf{X}\boldsymbol{\beta}\|^2] \tag{5}$$

and for an estimator $\hat{\boldsymbol{\beta}}$, the associated risk is

$$\mathcal{R}(\hat{\boldsymbol{\beta}}) = \mathbb{E}[\|\mathbf{Y}^{(1)} - \mathbf{X}\hat{\boldsymbol{\beta}}\|^2]. \tag{6}$$

Let us also define the empirical risk

$$\mathcal{R}_n(\hat{\boldsymbol{\beta}}) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \mathbf{X}_i\hat{\boldsymbol{\beta}})^2. \tag{7}$$

It is interpreted as the mean prediction error on the sample $\mathcal{D}_n$. The empirical risk is an unbiased estimator of the true risk, meaning that for the estimator $\hat{\boldsymbol{\beta}}$, the expected value of the empirical risk is equal to the true risk:

$$\mathbb{E}[\mathcal{R}_n(\hat{\boldsymbol{\beta}})] = \mathcal{R}(\hat{\boldsymbol{\beta}}) \tag{8}$$

The objective in model selection problems is to find the parameter that minimizes the empirical risk, whereas in model assessment, the goal is to accurately estimate this risk. Among the various techniques for risk estimation, cross-validation is particularly favored for its universality in data-splitting. This method fundamentally assumes that the data are identically distributed and that the training and validation sets are independent. Therefore, cross-validation's versatility makes it suitable for nearly any model assessment scenario. In the following section, we will explore the main cross-validation methods and their associated risk estimates.

1.1 Classical Cross-validation

We start by the so-called hold-out procedure. Let us divide the sample set $\mathcal{D}_n$ into two separated independent non-empty subsets $\mathcal{D}_n^t$ and $\mathcal{D}_n^v$ such that $\mathcal{D}_n^t \cup \mathcal{D}_n^v = \mathcal{D}_n$ and $\mathcal{D}_n^t \cap \mathcal{D}_n^v = \emptyset$. $\mathcal{D}_n^t$ is called the *training set* and is used to estimate; $\mathcal{D}_n^v$ is called the validation set and is used to estimate the risk.

We estimate $\boldsymbol{\beta}$ by $\hat{\boldsymbol{\beta}}^t$ using observations lying in $\mathcal{D}_n^t$, typically by minimizing

$$\frac{1}{\text{Card } \mathcal{D}_n^t}\sum_{(\mathbf{X}_i, Y_i) \in \mathcal{D}_n^t}(Y_i - \mathbf{X}_i\boldsymbol{\beta})^2.$$

Note that we could we write $\hat{\boldsymbol{\beta}}_n^t$ instead of $\hat{\boldsymbol{\beta}}^t$ to maintain an easier notation. Then we test the model's prediction ability induced by $\hat{\boldsymbol{\beta}}^t$ by computing the hold-out empirical risk, defined by

$$\mathcal{R}_n^v(\hat{\boldsymbol{\beta}}^t) = \frac{1}{\text{Card } \mathcal{D}_n^v}\sum_{(\mathbf{X}_i, Y_i) \in \mathcal{D}_n^v}(Y_i - \mathbf{X}_i\hat{\boldsymbol{\beta}}^t)^2, \tag{9}$$

We note here that the hold-out empirical risk depends on how the data has been split into the training and validation sets.

The K-fold cross-validation algorithm is closely related to the hold-out method. It partitions $\mathcal{D}_n$ into K subsets $\{\mathcal{D}_1, \ldots, \mathcal{D}_K\}$, the observations being randomly assigned to each subset, but ensuring that the cardinals of the subsets are more or less equal. For $m = 1, \ldots, K$, successively, the hold-out risk (9) is calculated on $D_n^v = \mathcal{D}_m$, $\boldsymbol{\beta}$ being estimated by $\hat{\boldsymbol{\beta}}^{t_m}$ on $D_n^t = \bigcup_{i \neq m} \mathcal{D}_i$. The K-fold cross-validation risk is the average of the hold-out risks:

$$\mathcal{R}_n^{KCV}(\hat{\boldsymbol{\beta}}) = \frac{1}{K} \sum_{m=1}^{K} \mathcal{R}_n^{v_m}(\hat{\boldsymbol{\beta}}^{t_m}). \tag{10}$$

K-fold cross validation is usually preferred to hold-out because it gives the opportunity to train the model on K training-validation subsets splits. This provides better indication of how well the model behaves on unknown data. Moreover it is shown that the estimates obtained by minimizing (10) improve in terms of both bias and variance compared to the ones obtained via simple hold-out (see for instance [1]).

When $K = n$, K-fold is called leave-one-out (LOO). In each iteration of LOO, one observation becomes the validation set and the remaining $n-1$ observations are used for training the model. Then we obtain, for $m \in \{1, \ldots n\}$, $\mathcal{D}_n^{t_m} = \{(\mathbf{X}_m, Y_m)\}^c$ the complementary subset of the singleton $\mathcal{D}_n^{v_m} = \{(\mathbf{X}_m, Y_m)\}$ and

$$\mathcal{R}_n^{LOO}(\hat{\boldsymbol{\beta}}) = \frac{1}{n} \sum_{m=1}^{n} \mathcal{R}_n^{v_m}(\hat{\boldsymbol{\beta}}^{t_m}) = \frac{1}{n} \sum_{m=1}^{n} (Y_m - \mathbf{X}_m \hat{\boldsymbol{\beta}}^{t_m})^2. \tag{11}$$

1.2 What can be difficult for spatial data?

As stated in the first law of geography and fundamental principle in geostatistical analysis according to [2], "Everything is related to everything else, but near things are more related than distant things".

There are two main drawbacks when adjusting a model using data with internal dependence structures:

- Non-independence of the residuals in the context of regression, which violates the critical assumption present in many models and methods, see [3].
- Overfitting to the dependence structure. It refers to a scenario where a model inappropriately captures the residual variation due to structural dependencies, rather than the actual signal. This occurs when the model confounds the effects of covariates with the spatial dependencies present in the residuals. In essence, the model might attribute variations that are due to inherent dependencies in the data to one or more explanatory variables. This misattribution can obscure true relationships and mask the non-independence of residuals, leading to a misunderstanding of the underlying

model's adequacy. Additionally, this type of overfitting complicates the detection of model misspecification. Since the dependence structure is absorbed into the covariate effects, standard diagnostic tests that check for the independence of residuals may not reveal any issues, thereby hiding the underlying problems in model specification.

The standard statistical approach to account for spatial dependence in data is to use parametric models that incorporate the corresponding dependence structure, as exemplified in [4]. However, in practice, the use of such models can suffer from specification error and structural overfitting, leading to poor performance in model evaluation. Therefore, to address these issues, there is a need for robust methods for validation, selection, and assessment of predictive accuracy in models that involve dependent data.
As previously discussed, classical cross-validation involves repeatedly dividing the data into independent training and validation sets. This partitioning is typically performed at random, which becomes problematic in the presence of dependence structures among the data. Such scenarios often lead to models that appear to perform well, but yield overly optimistic predictions for the users.

[5] recently showed that CV with random selection of the validation set can provide less biased estimates of the root mean squared error (RMSE) than Spatial Leave-One-Out and Spatial Blocked Cross-Validation, which are cross-validation procedures adapted to spatial data to be defined later on, when the sample data adequately represent the prediction area, for example, when samples are uniformly distributed across the prediction area. However, ensuring that the available data accurately represent the prediction locations can be challenging in practice. This difficulty is compounded by the tendency for samples to exhibit spatial clustering, for example [6] note that air quality observations are usually clustered in urban regions. This phenomenon of clustered sampling is especially prevalent on a global scale as mentioned by [7]. When data are clustered, the regions from which samples are collected tend to be overly represented. This leads to an under-representation of other regions in the prediction locations as shown by [8]. This is a commonly observed feature in spatial data; examples can be found in various fields, like ecology (see for instance [9]), air quality research ([10]), or soil data analysis ([11]).

Also, a good prediction model will likely be needed to be applied to new locations, this means that it need to be able to extrapolate. In our work the objective is not to test the extrapolation ability of models, for example we will not consider kriging.

Several approaches have been proposed to circumvent the limitations inherent in classical cross-validation when applied to spatial data. Before defining them precisely, let us introduce the basic principle of spatial cross-validation.

Let us consider a spatial domain $\mathcal{S}_n = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\} \subset \mathbb{R}^d$, where our sample data $\mathcal{D}_n = \{(\mathbf{X}(\mathbf{s}_i), Y(\mathbf{s}_i)) \mid \mathbf{s}_i \in \mathcal{S}_n\}$ is collected. Spatial CV needs to redefine the partitions within this domain.

The core modifications revolve around the principle of "point separation",

that is, strategically removing observations from the training set to achieve 'independence' between the training and validation sets. We do this by considering three spatial domains, the validation domain, the training domain and one that separates them enough to achieve (more or less) independence between them. Let us note h the necessary distance between the validation domain and the training domain that satisfies the independence assumptions. Specifically, for a chosen validation domain $\mathcal{S}_n^v \subset \mathcal{S}_n$, we delineate a new subset of sites $\mathcal{S}_{n,h}^D$ defined by $\mathcal{S}_{n,h}^D = \{\mathbf{s}_j \notin \mathcal{S}_n^v | \, \exists \, \mathbf{s}_i \in \mathcal{S}_n^v : d(\mathbf{s}_j, \mathbf{s}_i) \leq h\}$. We shall call this domain the "dead zone". The remaining sites compose the training domain $\mathcal{S}_{n,h}^t$. This is the set of all sites not in either $\mathcal{S}_n^v$ or $\mathcal{S}_{n,h}^D$, symbolized as $\mathcal{S}_{n,h}^t = \mathcal{S}_n \setminus (\mathcal{S}_n^v \cup \mathcal{S}_{n,h}^D)$. Let us note that the distance h is pivotal therefore its determination is one of the objectives of this dissertation.

The process can be algorithmically outlined as follows. Considering we have determinated h , we proceed with the m-th iteration of the cross-validation procedure in three steps:

1. **Select the validation set:** We first select the validation domain $\mathcal{S}_n^{v_m}$ for this iteration, and thus we define the validation set $\mathcal{D}_n^{v_m} = \{(\mathbf{X}(\mathbf{s}_i), Y(\mathbf{s}_i)) \mid s_i \in \mathcal{S}_n^{v_m}\}$.
2. **Establish the "dead zone":** $\mathcal{S}_{n,h}^{D_m}$.
3. **Define the training set:** The corresponding training set is then $\mathcal{D}_n^{t_m} = \{(\mathbf{X}(\mathbf{s}_i), Y(\mathbf{s}_i)) \mid \mathbf{s}_i \in \mathcal{S}_{n,h}^{t_m}\}$, where $\mathcal{S}_{n,h}^{t_m}$ is the training domain for this iteration.

It is important to note that the specific structure of the validation domain is not defined here. However, once the distance h is determined, the process for selecting the "dead zone" and the training domain remains consistent. This implies that the various adaptations of principal cross-validation methods can be distinguished by how they define the validation domain. The framework described before is particularly tailored for the isotropic case, wherein distances and separations are uniformly defined in all directions. Consequently, a single value of h suffices to delineate the minimal separation requisite between the validation and training sets to guarantee their independence. In the context of anisotropic scenarios, where the strength of the spatial dependence vary with direction, a simplistic approach with a single value of h may not suffice. Instead, a more nuanced approach is required, potentially involving two or more values of h to accurately capture the differences in spatial relationships. Figure1, illustrates the "dead zone" taken around one point in the case of anisotropy in horizontal and vertical directions and in the isotropic case. On the left, the elliptical blue contour delimits a "dead zone" surrounding a central point under anisotropic conditions. We assume that the dependence is expressed differently in the vertical and horizontal directions, therefore one has to determine both h_1 and h_2. In contrast, the "dead zone" under isotropic condition is delimited by the red circular contour, necessitating only the radius h.

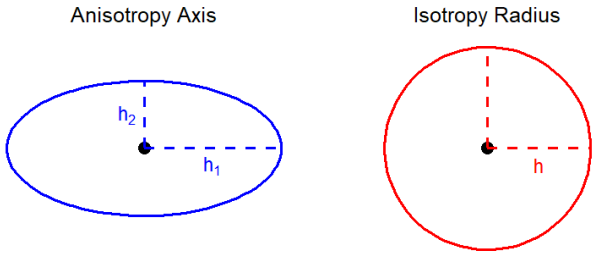

Figure 1: Example of "dead zone" under anisotropic and isotropic conditions.

We present hereafter the adaptations of main CV procedures to the spatial context.

2 Spatial cross-validation methods

2.1 Spatial leave-one-out

We start by presenting Spatial leave-one-out (SLOO), which will be the main method that we shall use later in the dissertation. [12] presented this spatial cross-validation procedure for model selection purposes. Back in 1994, [13] introduced a modification of leave-one-out cross-validation to deal with general stationary data, they called it h-block cross-validation and it contains the basic ideas behind spatial leave-one-out. As said before, the main idea of SLOO is to remove optimistic bias from the training set due to spatial dependece by omitting data points "close" to the one held out for validation from the training set, see figure 2.

Let us precise the Spatial leave-one-out procedure in our regression framework and under the assumption of isotropy. Let us assume that we the know the distance h required to satisfy independence between validation and training sets. At the $m-$th iteration of SLOO only one site is selected as the validation domain $\mathcal{S}_n^{v_m} = \{\mathbf{s}_m\}$ and $\mathcal{D}_n^{v_m} = \{(\mathbf{X}(\mathbf{s}_m), Y(\mathbf{s}_m))\}$ is the validation set. We define the dead zone around $\{\mathbf{s}_m\}$ as a neighbourhood of radius h, $\mathcal{S}_{n,h}^{D_m} = \{\mathbf{s}_j \in \mathcal{S}_n | j \neq m \text{ and } d(\mathbf{s}_j, \mathbf{s}_m) \leq h)\}$; the remaining points form the training domain, $\mathcal{S}_{n,h}^{t_m} = \{\mathbf{s}_j \in \mathcal{S}_n : d(\mathbf{s}_j, \mathbf{s}_m) > h\}$ and then the corresponding training set is $\mathcal{D}_n^{t_m} = \{(\mathbf{X}(\mathbf{s}_j), Y(\mathbf{s}_j)) | \mathbf{s}_j \in \mathcal{S}_{n,h}^{t_m}\}$.

The predictive empirical risk is given by

$$\mathcal{R}_{n,h}^{SLOO}(\hat{\boldsymbol{\beta}}) = \frac{1}{n}\sum_{m=1}^{n} \mathcal{R}_n^{v_m}(\hat{\boldsymbol{\beta}}_h^{t_m}) = \frac{1}{n}\sum_{m=1}^{n} (Y(\mathbf{s}_m) - \mathbf{X}(\mathbf{s}_m)\hat{\boldsymbol{\beta}}_h^{t_m})^2 \qquad (12)$$

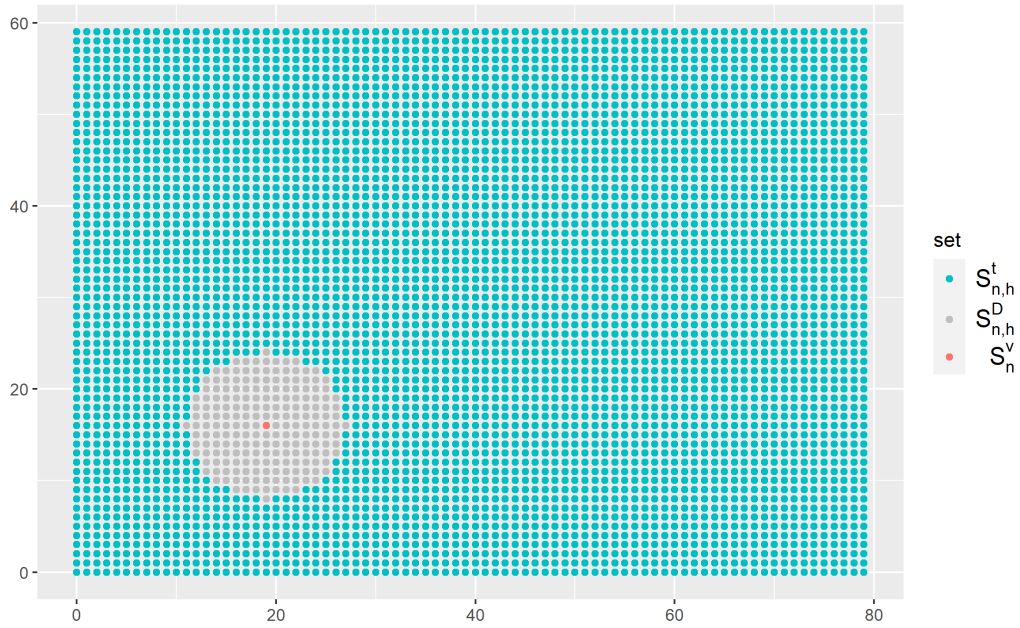

Figure 2: Dataset partition for one iteration of spatial leave-one-out (SLOO) with buffer of radius $h = 8$

where we note $\hat{\boldsymbol{\beta}}_h^m$ the estimator computed on the training set $\mathcal{D}_{n,h}^{t_m}$, whose expression is,

$$\hat{\boldsymbol{\beta}}_h^m = \left(\mathbf{X}_{m,h}^{\mathrm{T}}\mathbf{X}_{m,h}\right)^{-1}\mathbf{X}_{m,h}^{\mathrm{T}}\mathbf{Y}_{m,h}, \tag{13}$$

with $\mathbf{X}_{m,h}$ and $\mathbf{Y}_{m,h}$ denoting respectively the matrix $\mathbf{X}$ and the vector $\mathbf{Y}$ whose elements related to $\mathcal{S}_n^{v_m} \cup \mathcal{S}_{n,h}^{D_m}$ have been deleted.

The spatial leave-one-out cross-validation method is widely used in practice and has been shown to produce reliable estimates of predictive error. Many works have provided evidence of its superiority to ordinary LOO cross-validation in the spatial context. For example, [14] found that SLOO provided the most unbiased estimates of predictive error across a range of sample sizes when compared with ordinary K-fold CV and LOO cross-validation. [15] also provided evidence of the superiority of SLOO when compared to LOO when modelling mapping seabed sediments using random forest.

Given its popularity, several extensions and modifications have been proposed to improve its performance and applicability in different settings. [15] proposed a method called spatially resampled leave-one-out cross-validation which extends the SLOO approach by considering a spatial buffering procedure not only around the validation domain but also from each point on the training domain, this ensures that no adjacent points are used for model fitting or validation. This modification enhances the method's ability to account for spatial auto-

correlation and avoid biased estimates of model performance. [16] proposed a new method for map validation called Nearest Neighbour Distance Matching (NNDM) LOO CV. The NNDM algorithm refines the LOO cross-validation process for irregularly distributed locations. At each iterations they consider the distance of the validation point to its nearest neighbour.

2.1.1 Limitations

SLOO inherits two limitations from the classical leave-one-out method. Firstly, it can be computationally expensive, especially when dealing with large datasets or complex training models and learning algorithms. The process of iteratively training and validate the model for each individual point in the dataset can require significant computational resources.
Furthermore, SLOO tends to exhibit high variance in the hold-out estimations. This is because the validation set in LOO and SLOO consists of only one data point, leading to increased variability in the performance metrics, compared to methods that use larger validation sets.
As underlined before, the efficacy of critically depends on the accurate selection of the spatial buffer parameter, h. Selecting an optimal h value requires a careful balance: it should be large enough to maintain sufficient observations in the training set for effective model calibration, yet adequately expansive to achieve the necessary independence from the validation set. Several researchers have offered different strategies for setting this spatial parameter. roberts2017cross recommend determining the separation distance by the range at which residual spatial autocorrelation is nullified. le2014spatial and telford2009evaluation propose to set h equal to the range value of the variogram calculated from the model's residuals. Meanwhile, karasiak2022spatial estimate the spatial dependence at distance h using Moran's Index:

$$I(h) = \frac{n}{S_0} \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j}(Y(\mathbf{s}_i) - \overline{Y})(Y(\mathbf{s}_j) - \overline{Y})}{\sum_{i=1}^{n}(Y(\mathbf{s}_i) - \overline{Y})^2} \tag{14}$$

where $w_{i,j} = 1$ if $d(\mathbf{s}_i, \mathbf{s}_j) = h$ and equals 0 otherwise, and $S_0 = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j}$. They plot Moran's I as a function of the distance h and they advise to choose the threshold at which the index is not significant to ascertain the appropriate separation distance. This method can be very time-consuming due to the necessity of computing pairwise distances for all data points and to repeatedly update the binary weight matrix W for different values of h. Additionally, each calculation of Moran's I requires complex aggregation and summation across large datasets, and the need for iterative computation to identify non-significant distances. Statistical significance testing of Moran's I, which may include permutation tests, further increases the computational demand, making the process particularly challenging for large spatial datasets.

Overall, while SLOO offers a spatially aware approach to cross-validation, it is important to consider its computational demands, the increased variance of hold-out estimations, and the careful selection of the spatial bandwidth parameter to ensure its effectiveness in model evaluation and prediction.

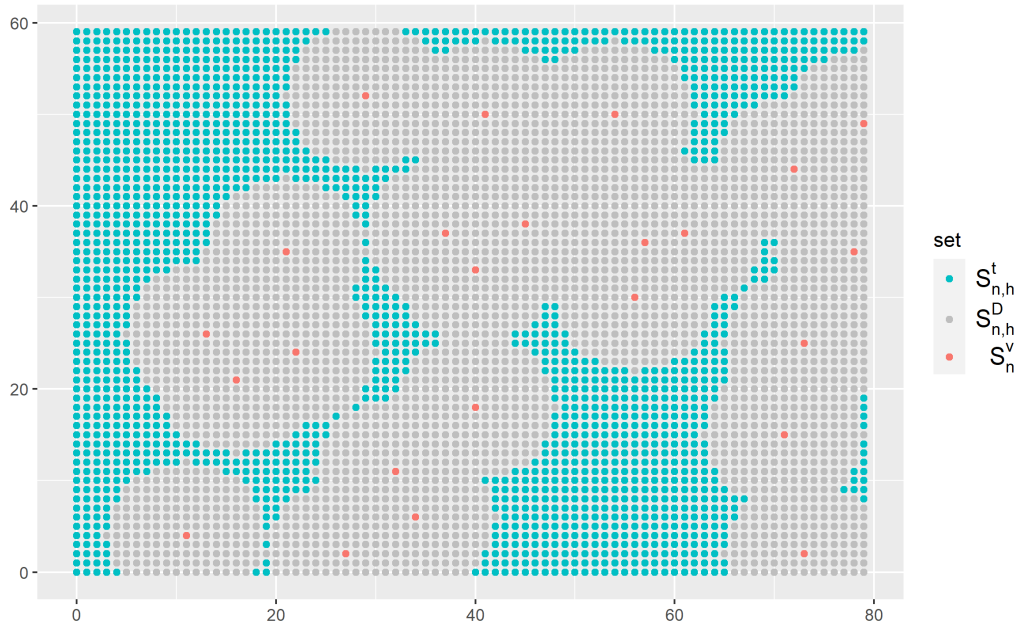

Figure 3: Dataset partition for one iteration of Spatial K-fold cross-validation with buffer radius $h = 8$ and $K = 200$

2.2 Spatial K-fold cross-validation

Spatial K-fold cross-validation (SKVC), is a modified K-fold CV method proposed by [17],

We follow the notations of section 1.2. Let K be the number of folds, the set $\{\mathcal{S}_n^{v_1}, \ldots, \mathcal{S}_n^{v_K}\}$ is the set of CV folds defined such that for each $m : 1 \leq m \leq K$, the locations in $\mathcal{S}_n^{v_m}$ are selected randomly, Card $(\mathcal{D}_n^{v_m}) = n_m, [n/K] \leq n_m \leq [n/K] + 1)$, $\cap_m \mathcal{S}_n^{v_m} = \emptyset$ and $\cup_m \mathcal{S}_n^{v_m} = \mathcal{S}_n$.

In the $m-$th step of the SKCV algorithm, given validation domain $\mathcal{S}_n^{v_m}$, the set of dead points is $\mathcal{S}_{n,h}^{D_m} = \{\mathbf{s}_j \notin \mathcal{S}_n^{v_m} | \exists\ \mathbf{s}_i \in \mathcal{S}_n^{v_m} : d(\mathbf{s}_j, \mathbf{s}_i) \leq h\}$ and the training domain is $\mathcal{S}_{n,h}^{t_m} = \{\mathbf{s} \in \mathcal{S}_n | \mathbf{s} \notin \mathcal{S}_n^{v_m} \cup \mathcal{S}_{n,h}^{D_m}\}$.

The SKCV predictive empirical risk is given by:

$$\mathcal{R}_{n,h}^{SKCV}(\hat{\boldsymbol{\beta}}) = \frac{1}{K} \sum_{m=1}^{K} \mathcal{R}_n^{v_m}(\hat{\boldsymbol{\beta}}_h^{t_m}). \tag{15}$$

where we $\hat{\boldsymbol{\beta}}_h^{t_m}$ is the estimator computed on the training set $\mathcal{D}_{n,h}^{t_m}$. Figure 3 illustrates an example of SKCV. $\mathcal{S}_n$ is a lattice of size $n = 80 \times 60$, we consider $K = 200$ folds of cardinal 24 and $h = 8$

Significant research related to SKCV has focused on modifying the type of distance used to define the "dead zone". For instance, [18] considers spatially

disjoint partitioning for image data, which accounts for topological constraints arising from the spatial arrangement of features.

Furthermore, [19] argue that the Euclidean distance may not be the most suitable measure for determining the deadzone in urban environments. As an alternative, they recommend calculating the "dead zone" based on road distance and travel time, providing a relevant approach for the complex spatial dynamics of urban areas.

2.2.1 Limitations

When utilizing spatial K-fold cross-validation, one potential drawback is the removal of a large number of training data points. This can introduce an additional pessimistic bias in the prediction performance. To investigate this bias, [17] conducted a study where they randomly removed the same number of points as those removed during SKCV and compared the results. Their findings indicate that random removal of training data points has a negligible impact on prediction performance compared to spatial-based data removal.
In addition to the issue of data removal and similarly to SLOO, the performance of SKCV is also influenced by the selection of the buffer radius.
Furthermore, the number of folds used in the SKCV procedure can exacerbate the challenges mentioned above. Depending on the size and spatial distribution of the data, it is possible that a significant portion of the training data is removed due to the combined effect of dead zones. Therefore, selecting an appropriate number of folds is crucial to strike a balance between capturing spatial dependencies and maintaining an adequate sample size for model training. To optimize the performance of SKCV, careful consideration should be given to both the selection of buffer radius and the number of folds.

2.3 Blocked cross-validation

A well known spatial cross-validation technique is blocked cross-validation. The technique is similar to SKCV but the locations in each fold are not selected randomly but chosen following a block pattern. Partitioning the spatial domain into blocks for cross-validation is particularly advantageous when the objective is to identify causal predictors or to forecast within novel dependence frameworks. This is because each block can encapsulate a unique segment of the spatial domain, characterized by potentially distinct features. [20] conducted a survey on blocked cross-validation for dependent data. They concluded that besides the common objectives of spatial cross-validation, block cross-validation performs well to measure the ability of a model to extrapolate to independent data.
This spatial cross-validation method can be considered with and without spatial buffering, in what follows we will refer to Spatial blocked cross-validation (SBCV) when considering $h > 0$ and blocked cross-validation otherwise (BCV). Let B be the number of blocks, we can keep the same notation as for SKCV with $K = B$. The difference is that the random configuration of the folds is replaced

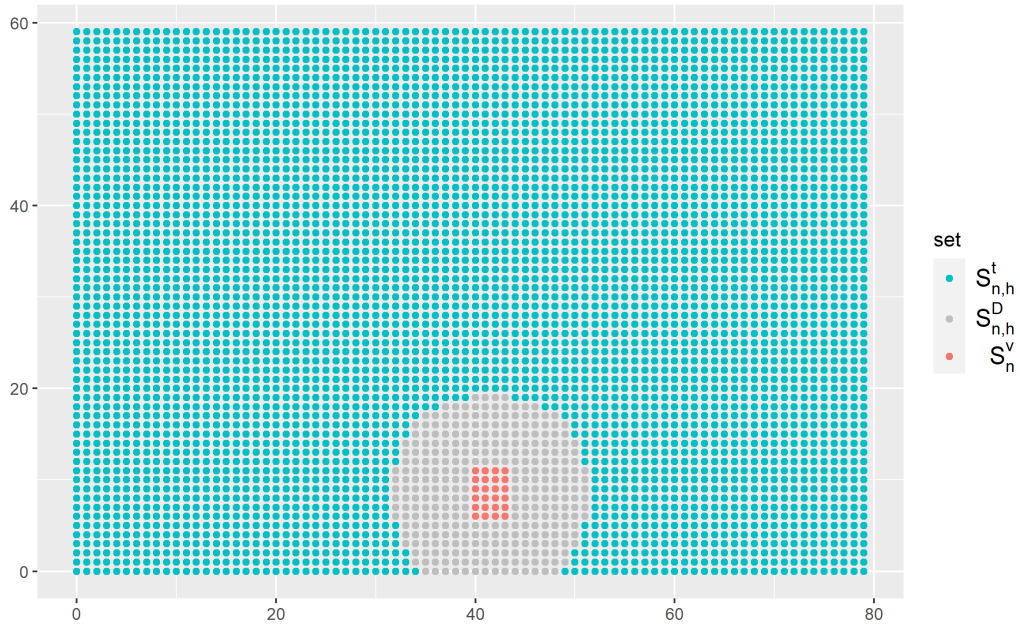

Figure 4: Dataset partition for one iteration of Spatial blocked cross-validation with buffer of radius $h = 8$ and square block of size 4×6.

by blocks consisting of neighbour locations. Figure 3 illustrates an example of SBCV. $\mathcal{S}_n$ is a lattice of size $n = 80 \times 60$, we consider $B = 200$ blocks of size 6×4 and $h = 8$. At this point it is worth mentioning that although, the number of locations included in the validation set in figures 3 and 4 is the same, the cardinal of $\mathcal{S}_{n,h}^D$ is 310 for SBCV and 3074 for SKCV.

The risk estimate for SBCV is given by:

$$\mathcal{R}_{n,h}^{SBCV}(\hat{\boldsymbol{\beta}}) = \frac{1}{B}\sum_{m=1}^{B}\mathcal{R}_n^{v_m}(\hat{\boldsymbol{\beta}}_h^{t_m}). \tag{16}$$

It is important to note that unlike SBCV, the risk estimate for BCV, $\mathcal{R}_n^{BCV}(\hat{\boldsymbol{\beta}})$, does not depend on h, because there is no "dead zone".

In the context of SBCV, the structuring of partitions ensures that the quantity of points encompassed within the 'dead zone' is considerably less than that observed in SKCV. If we compare figures 3 and 4 this difference becomes obvious, though the validation domain in both methods include an identical number of locations (24). This distinction underscores the methodological advantage of SBCV in minimizing the impact of the "dead zone", thereby enhancing the integrity of the validation process. Blocked cross-validation can be seen as a large class of cross-validation methods due to the variety of ways to define blocks. [20] discusses oblong blocks as well as a checker board structure for a hold out kind of spatial cross-validation. [21] proposes a blocked cross-validation with and without spatial buffering where the blocks are constructed using the K-means algorithm. Another variant is to consider more than one block at each iteration of the algorithm.

2.3.1 Limitations

When employing blocking techniques, it is important to consider the potential implications on the characteristics of the data. Grouping similar structural units together through blocking may inadvertently remove unique features and introduce extrapolation issues, especially in situations where the models are intended to interpolate, see for example [22].

An additional critical consideration in SBCV involves determining the optimal block size for regular blocks. roberts2017cross recommend estimating a variogram of the dependent variable and adopting its range as the block size. Conversely, in the `blockCV` package, valavi2019blockcv suggest selecting the block size based on the median of the ranges obtained from the variograms computed on the covariates.

Furthermore, defining effective regular blocks can be challenging when dealing with data clusters resulting from irregular sampling patterns. In such scenarios, alternative approaches can be considered. One solution involves using irregularly arranged but similarly sized blocks, which can help capture the spatial structure while accounting for the data distribution. Another approach is to employ irregularly shaped blocks, tailored to match the spatial clustering of the

data, providing a more accurate representation of the underlying spatial patterns.
In summary, when applying blocking techniques, it is crucial to balance the size and the structure of the blocks to minimize the loss of unique characteristics, utilize as much data as possible for training, and consider alternative strategies when dealing with irregular data clusters.

2.4 Comparison between spatial cross-validation strategies

We conduct a simulation study to compare the different cross-validation estimates of the empirical risk. Let $\mathcal{S}_n$ be a rectangular lattice of size 80×60, with $n = 4800$ sites. We consider a non stationary trend, linked to the spatial location on the domain. We consider the regression model

$$Y(\mathbf{s}) = \mathbf{X}(\mathbf{s})^{\mathrm{T}}\boldsymbol{\beta} + \varepsilon(\mathbf{s}), \tag{17}$$

$\boldsymbol{\varepsilon}$ is generated according to a zero-mean Gaussian spatial process model with isotropic (in both vertical and horizontal directions) exponential covariance function defined by $C(h) = \sigma^2 \exp\left(-\frac{h}{\theta}\right)$. We choose this covariance functions because it is a well known model. We consider $\sigma^2 = 1$, and different values of $\theta = (3, 6, 9, 12, 15, 20)$ which determines the strength of the spatial dependence. For the trend we consider the following design; denoting $\mathbf{s} = (s_1; s_2) \in \mathcal{S}_n$, we define

$$\mathbf{X}(\mathbf{s})^{\mathrm{T}} = (1; s_1 - 40; s_2 - 30)$$

We calibrate $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)^{\mathrm{T}} = (0.9, 1/32, 1/24)$, β_1, β_2 are chosen following the procedure described by [23]. It ensures that the deterministic term exhibits a comparable empirical variance to the one of the error term. The value of β_0 is selected to obtain a similar coefficient of variation as β_1 and β_2. Let us note that in our model, though the expectation of $Y(\mathbf{s})$ is not constant, the covariance between $Y(\mathbf{s}_i)$ and $Y(\mathbf{s}_j)$ depends only on $d(\mathbf{s}_i, \mathbf{s}_j)$. Figure 5 shows a realization of Y with $\theta = 9$.

Our estimation scenario ignores the spatial dependence structure; we estimate the empirical risk associated to the ordinary least squares (OLS) estimator by the different cross-validation methods.

We consider the following cross-validation methods:

- **Ordinary Leave-one-out**
- **Ordinary K-Fold Cross Validation**: The dataset is divided into $K = 200$ folds.
- **Spatial Leave-one-out**: Here we consider different buffer sizes of $h \in \{5, 10, 15, 20\}$.
- **Spatial K-Fold Cross Validation (SKCV)**: We keep $K = 200$ and $h \in \{5, 10, 15, 20\}$.

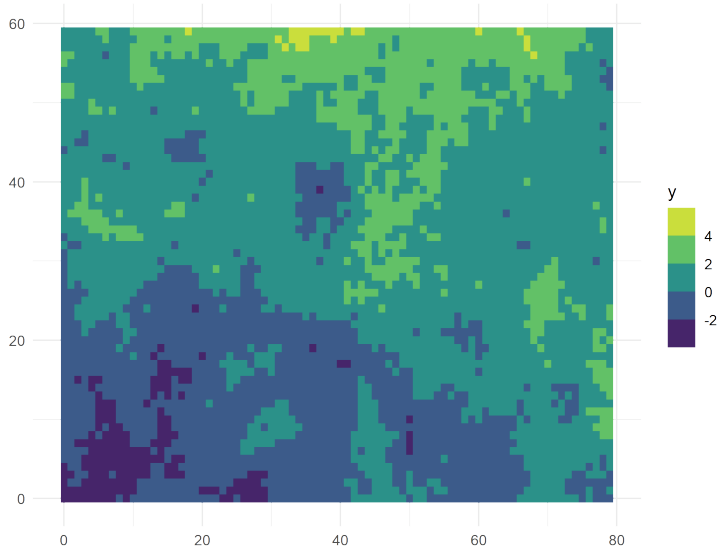

Figure 5: Simulated data generated following model 17 with $\theta = 9$

- **Blocked Cross Validation (without spatial buffering)**: Here we consider $B = 200$ blocks of size 6×4 which aligns with the number of folds and the size of the validation set in each iteration with those used in KCV. And later also consider block sizes 8×12, 15×16 and 20×20.
- **Spatial Block Cross Validation**: We add spatial buffering around each block with $h \in \{5, 10, 15, 20\}$.

The cross-validation estimates of the risk are compared to an *ideal* risk. Given $M = 100$ independent realizations $Y^{(1)}, \cdots, Y^{(M)}$ of Y, let $Y^{(l)}$ be the $l-$th copy. We define the empirical *ideal* predictive risk for $Y^{(l)}$ as

$$\mathcal{R}_n^{I(l)} = \frac{1}{M-1} \sum_{\substack{j=1 \\ j \neq l}}^{M} \frac{1}{n} \sum_{\mathbf{s} \in \mathcal{S}_n} (\hat{Y}^{(l)}(\mathbf{s}) - Y^{(j)}(\mathbf{s}))^2, \tag{18}$$

where $\hat{Y}^{(l)}(\mathbf{s}) = \mathbf{X}(\mathbf{s})\hat{\boldsymbol{\beta}}^{(l)}$ with $\hat{\boldsymbol{\beta}}^{(l)} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}^{(l)}$. This ideal error is obtained by the principle of predicting onto the other independent realizations.

Let's first compare SBCV and SKCV with respect to the size of the "dead zone" $\mathcal{S}_h^D$ and the training domain $\mathcal{S}_h^t$ for a fixed size of the validation domain $\mathrm{Card}(\mathcal{S}_n^v) = 24$. Table 1 gives the average of the number of points and the associated percentage with respect to $n = 4800$ for both subsets and for different buffer sizes. We observe that on average, for SKCV a staggering 93.61% of the spatial domain is categorized as the "dead zone" for $h = 15$. This occasionally

leads to scenarios where the training set is entirely vacated. Even at $h = 10$, the model is trained on merely an average of 25% of $\mathcal{S}_n$, a stark contrast to SBCV's utilization of approximately 91.15% of $\mathcal{S}_n$ for training, despite $\mathcal{S}_n^v$ being equivalently sized as in SKCV. This disparity highlights the potential limitations of SKCV in spatial contexts and accentuates the efficiency of SBCV in leveraging available data for model training.

Table 1: Comparison of sizes of dead zone and training domain for SBCV and SKCV for $\text{Card}(\mathcal{S}_n^v) = 24$, showing percentages of spatial domain $n = 4800$

Method	h	ave(Card($\mathcal{S}_n^D$)) (%)	ave(Card($\mathcal{S}_n^t$)) (%)
BCV	0	0 (0%)	4776 (99.5%)
KCV		0 (0%)	4776 (99.5%)
SBCV	5	143.7 (2.99%)	4632.3 (96.51%)
SKCV		1502 (31.29%)	3272 (68.19%)
SBCV	10	400.96 (8.35%)	4375 (91.15%)
SKCV		3588 (74.5%)	1188 (24.75%)
SBCV	15	746.72 (15.56%)	4029.28 (83.94%)
SKCV		4493.14 (93.61%)	282.87 (5.86%)
SBCV	20	1157.04 (24.11%)	3618.96 (75.39%)
SKCV		4713.56 (98.20%)	62.45 (1.3%)

Figure 6 showcases the risk estimates from various estimation methods and the *ideal* risk for the different spatial dependence strengths indicated by θ. We considered different buffer sizes h as said before, $h = 0$ corresponds to the methods without spatial buffering.
Across all examined values of θ, it is evident that the Leave-One-Out, K-fold Cross-Validation and Blocked Cross-Validation without spatial buffering methods yield the most optimistic risk assessments. This consistent optimism in risk estimates may suggest a potential underestimation of true risk in non-spatial cross-validation methods.

As θ increases, we see a corresponding rise in both mean and variance of the *ideal* risk, signaling the mounting challenges in modeling the data with linear regression using OLS estimation amidst stronger spatial dependencies. Analogously, the risks estimated by (spatial and non-spatial) cross-validation methods increase in variance with θ.

As expected, the need of a spatial aware cross-validation method becomes more obvious as θ increases as well as the necessity of a larger buffer size. Particularly at higher values of θ, we observe that none of the estimation methods manage to meet the *ideal* risk, underscoring a collective limitation in the face of high spatial dependence. Indeed we could not consider larger buffers because we would lose too many points in the "dead zone" and consequently keep not enough points for accurate training. In particular, we note the computational limitation encountered with Spatial K-Fold Cross-Validation at higher values of h, specifically $h = \{15, 20\}$, where the training set was rendered empty post

the exclusion of validation and dead zone areas.

Upon examining the spatial cross-validation methods more closely, we note an upward trend in risk estimates as h increases. Additionally, the variability of these estimates grows with h, indicating greater predictive uncertainty at larger spatial separations for a constant θ. Comparing SLOO and SBCV, the results are generally similar, with SLOO typically providing marginally lower risk estimates than SBCV for the same h, but exhibiting slightly reduced variability. This resemblance suggests that both methods react similarly to the spatial structure of the data for the given block size. These findings are supported by Figure 7, which illustrates the accuracy of risk estimates for SLOO and SBCV relative to the ideal risk, computed as follows:

$$\delta(h) = \left(\frac{1}{M} \sum_{l=1}^{M} (\mathcal{R}_{n,h}^{(l)} - \mathcal{R}_{n}^{I(l)}) \right)^2 \tag{19}$$

The variance of these estimates in relation to their mean risk is also analyzed, calculated by:

$$\sigma^2_{\mathcal{R}_{n,h}}(h) = \frac{1}{M} \sum_{l=1}^{M} \left(\mathcal{R}_{n,h}^{(l)} - \bar{\mathcal{R}}_{n,h} \right)^2 \tag{20}$$

where $\bar{\mathcal{R}}_{n,h} = \frac{1}{M} \sum_{l=1}^{M} \mathcal{R}_{n,h}^{(l)}$. These two measures are presented in the spirit of a bias-variance decomposition, although $\delta(h)$ is not formally a bias. We still observe the increase of variance with h and the necessity of introducing spatial buffering in cross-validation, especially for large values of θ. Also, SBCV has generally higher variance and lower accuracy than SLOO.

To further explore the influence of block size, we examine four distinct configurations: block sizes of 4×6, 8×12, 16×15, and 20×20, which respectively comprise 1%, 2%, 5%, and 8% of the total sample size. Figure **??** displays the risk estimates from SBCV for these block sizes alongside the *ideal* risk, categorized by varying levels of spatial dependence denoted by θ. A consistent trend is observed where risk estimates increase with the enlargement of block size. Notably, in scenarios with minimal spatial dependence ($\theta = 3$), the risk estimates for the largest block size (20×20) are excessively pessimistic. Furthermore, the variance of these estimates tends to increase with block size. Finally, the necessity of using spatial buffering remains present, especially as the strength of spatial dependence intensifies, even for the largest block size considered.

In summary, SKCV should not be considered due to the loss of observations. SBCV and SLOO emerge as superior methods for risk estimation, producing results that align more closely with the ideal risk. When selecting between these two, it is crucial to recognize the importance of appropriately determining h. A first point of distinction lies in computational efficiency; SBCV offers a notable reduction in computation time compared to SLOO, requiring only 200 iterations against SLOO's 4800, which translates to a significant decrease in computational demand. However, for SBCV, one needs to determine the appropriate shape and block size, which serves as an additional parameter and needs to be selected.

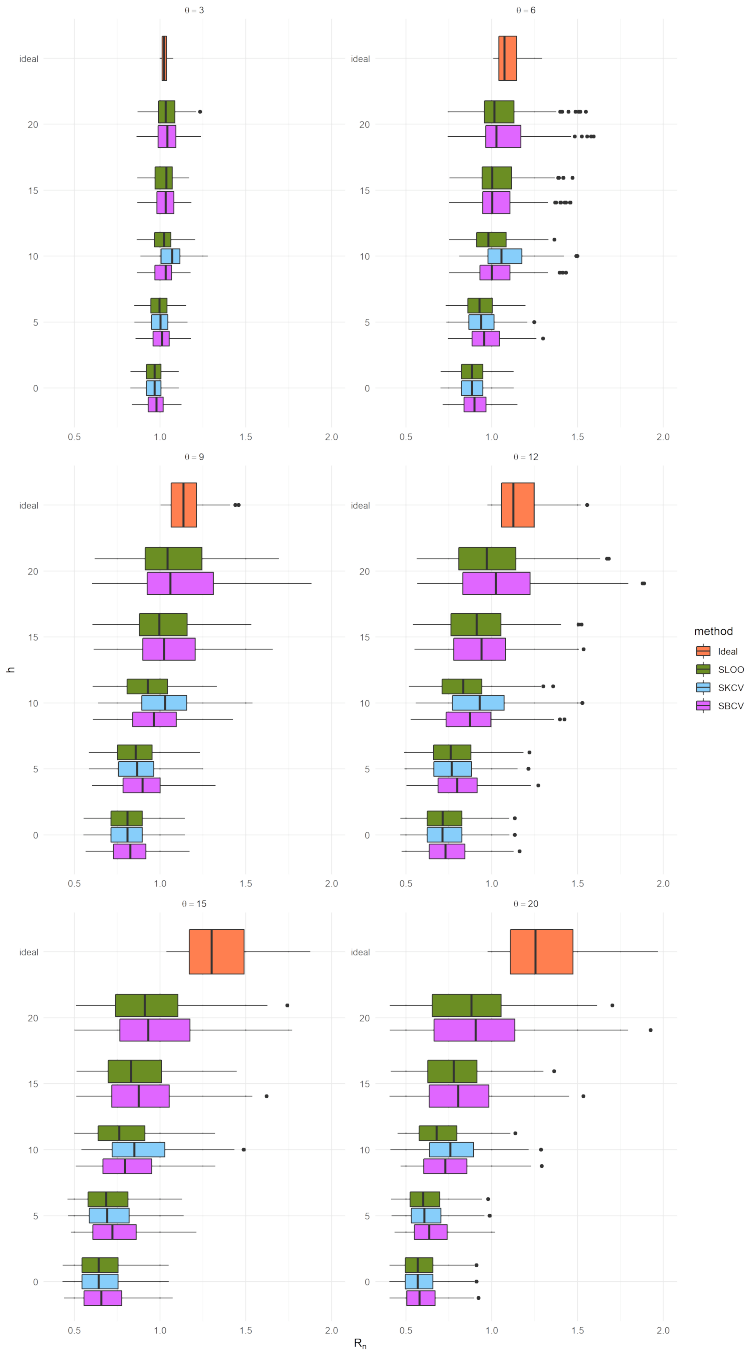

Figure 6: Risk estimates derived from 100 replications of the response variable Y, compared across different cross-validation methods. The results are grouped according to the value of the parameter θ, of the exponential covariance function in the simulation. For $h = 0$ SLOO, SKCV and SBCV become LOO, K-fold cross-validation and block cross-validation.

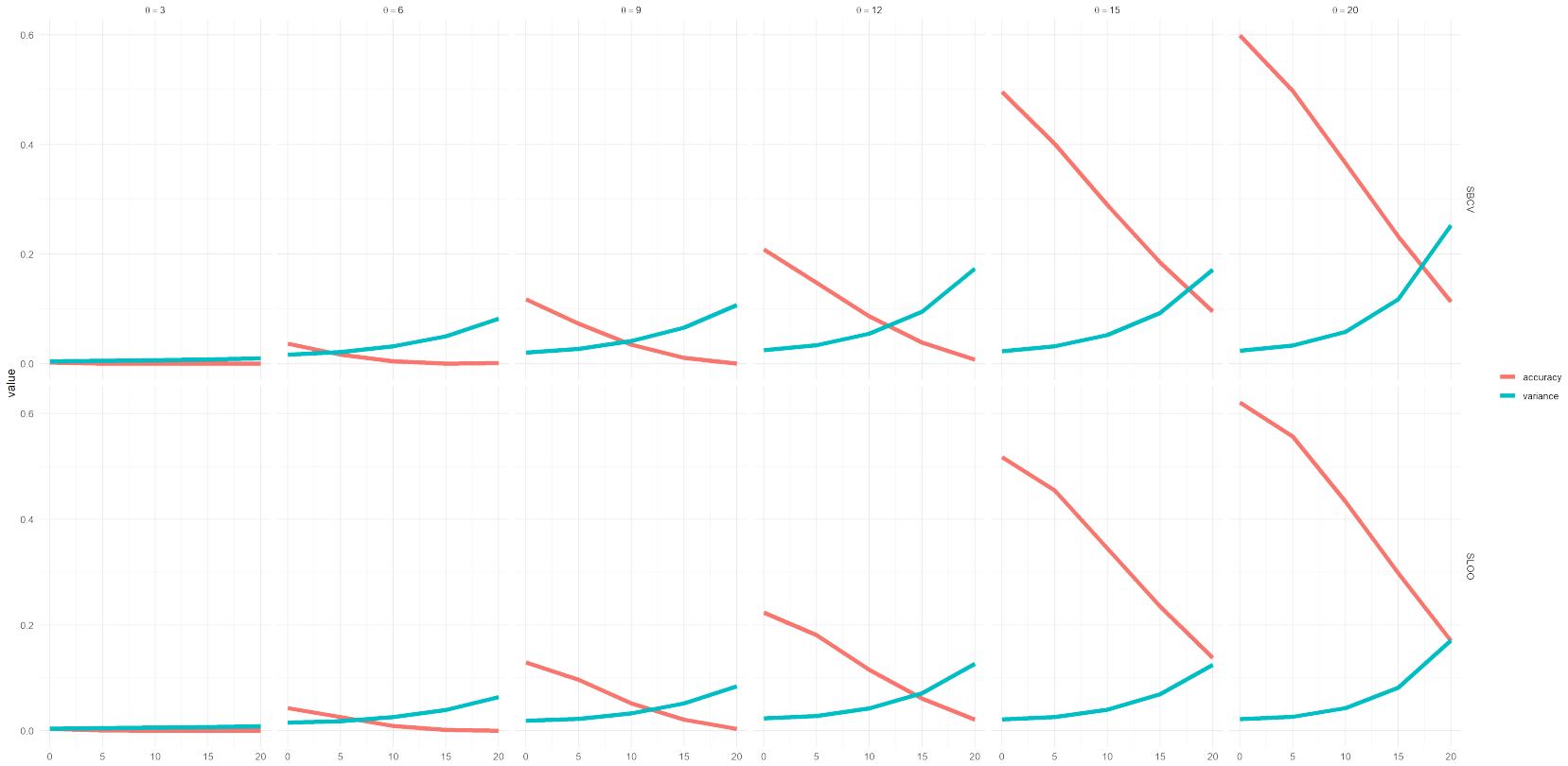

Figure 7: Comparative accuracy of risk estimates for SLOO and SBCV relative to the ideal risk across varying values of θ. The plot also illustrates the variance of these estimates in relation to their mean risk.

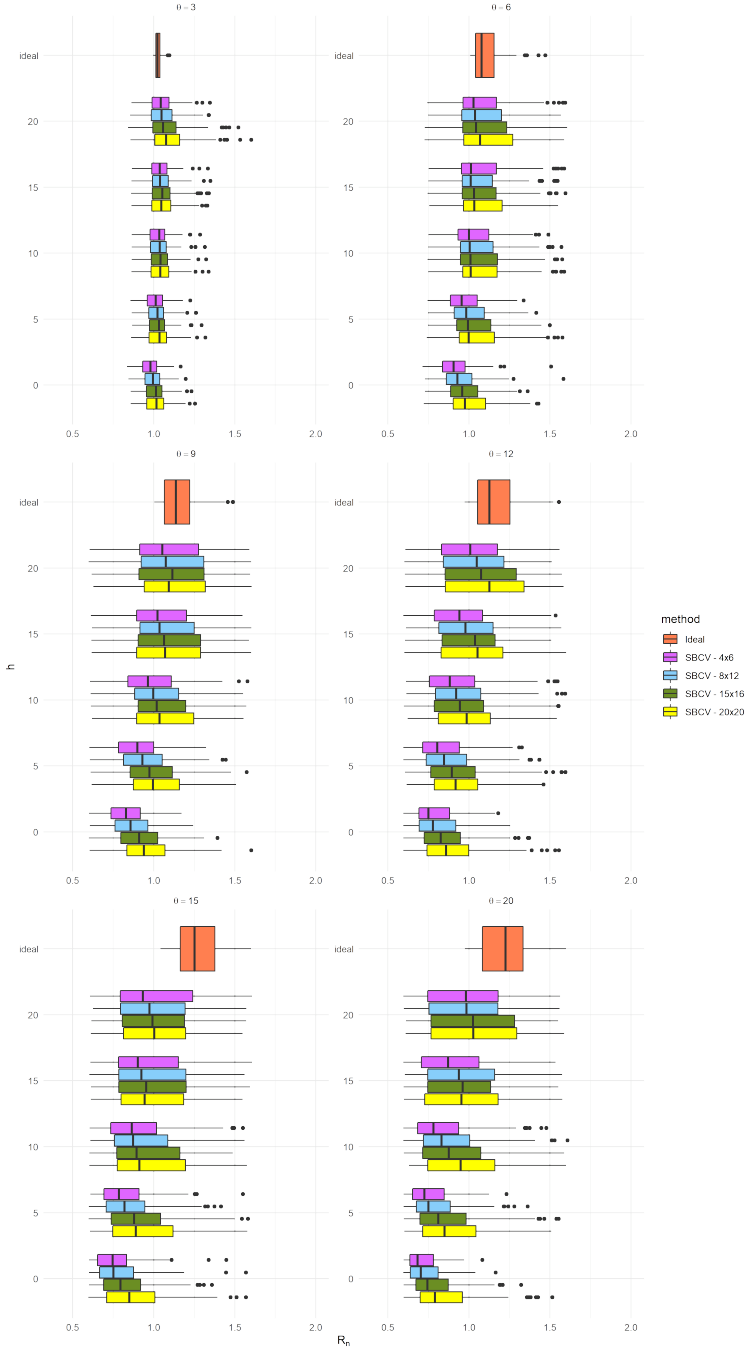

Figure 8: Risk estimates from SBCV based on 100 replications of the response variable Y, compared across various block sizes and against the Ideal risk. Estimates are grouped by the parameter θ from the exponential covariance function used in the simulation.

References

[1] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[2] Waldo R Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, 46(sup1):234–240, 1970.

[3] Pierre Legendre. Spatial autocorrelation: trouble or new paradigm? *Ecology*, 74(6):1659–1673, 1993.

[4] Carsten F. Dormann, Jana M. McPherson, Miguel B. Araújo, Roger Bivand, Janine Bolliger, Gudrun Carl, Richard G. Davies, Alexandre Hirzel, Walter Jetz, W Daniel Kissling, et al. Methods to account for spatial autocorrelation in the analysis of species distributional data: a review. *Ecography*, 30(5):609–628, 2007.

[5] Alexandre M.J.-C. Wadoux, Gerard B.M. Heuvelink, Sytze de Bruin, and Dick J. Brus. Spatial cross-validation is not the right way to evaluate map accuracy. *Ecological Modelling*, 457:109692, 2021.

[6] Tongwen Li, Huanfeng Shen, Chao Zeng, and Qiangqiang Yuan. A validation approach considering the uneven distribution of ground stations for satellite-based pm2.5 estimation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:1312–1321, 2020.

[7] Hanna Meyer and Edzer Pebesma. Machine learning-based global maps of ecological variables and the challenge of assessing them. *Nature Communications*, 13(1):2208, 2022.

[8] Sytze de Bruin, Dick J. Brus, Gerard B.M. Heuvelink, Tom van Ebbenhorst Tengbergen, and Alexandre M.J-C. Wadoux. Dealing with clustered samples for assessing map accuracy by cross-validation. *Ecological Informatics*, 69:101665, 2022.

[9] Pierre Ploton, Frédéric Mortier, Maxime Réjou-Méchain, Nicolas Barbier, Nicolas Picard, Vivien Rossi, Carsten Dormann, Guillaume Cornu, Gaëlle Viennois, Nicolas Bayol, et al. Spatial validation reveals poor predictive performance of large-scale ecological mapping models. *Nature communications*, 11(1):4540, 2020.

[10] Qingyang Xiao, Howard H. Chang, Guannan Geng, and Yang Liu. An ensemble machine-learning model to predict historical pm2.5 concentrations in china from satellite data. *Environmental Science & Technology*, 52(22):13260–13269, 2018. PMID: 30354085.

[11] Tomislav Hengl, Gerard B. M. Heuvelink, Bas Kempen, Johan G. B. Leenaars, Markus G. Walsh, Keith D. Shepherd, Andrew Sila, Robert A.

MacMillan, Jorge Mendes de Jesus, Lulseged Tamene, and Jérôme E. Tondoh. Mapping soil properties of africa at 250 m resolution: Random forests significantly improve current predictions. *PLOS ONE*, 10(6):1–26, 06 2015.

[12] K. Le Rest, D. Pinaud, and V. Bretagnolle. Accounting for spatial autocorrelation from model selection to statistical inference: application to a national survey of a diurnal raptor. *Ecological informatics*, 14:17–24, 2013.

[13] Prabir Burman, Edmond Chow, and Deborah Nolan. A cross-validatory method for dependent data. *Biometrika*, 81(2):351–358, 1994.

[14] Nicolas Karasiak, J-F Dejoux, Claude Monteil, and David Sheeren. Spatial dependence between training and test sets: another pitfall of classification accuracy assessment in remote sensing. *Machine Learning*, 111(7):2715–2740, 2022.

[15] Benjamin Misiuk, Markus Diesing, Alec Aitken, Craig J Brown, Evan N Edinger, and Trevor Bell. A spatially explicit comparison of quantitative and categorical modelling approaches for mapping seabed sediments using random forest. *Geosciences*, 9(6):254, 2019.

[16] Carles Mila, Jorge Mateu, Edzer Pebesma, and Hanna Meyer. Nearest neighbour distance matching leave-one-out cross-validation for map validation. *Methods in Ecology and Evolution*, 13(6):1304–1316, 2022.

[17] J. Pohjankukka, T. Pahikkala, P. Nevalainen, and J. Heikkonen. Estimating the prediction performance of spatial models via spatial k-fold cross validation. *International Journal of Geographical Information Science*, 31:2001–2019, 2017.

[18] Christian Geiß, Patrick Aravena Pelizari, Henrik Schrade, Alexander Brenning, and Hannes Taubenböck. On the effect of spatially non-disjoint training and test samples on estimated model generalization capabilities in supervised classification with spatial features. *IEEE Geoscience and Remote Sensing Letters*, 14(11):2008–2012, 2017.

[19] Theodoros Damoulas Henry Crosby and Stephen A. Jarvis. Road and travel time cross-validation for urban modelling. *International Journal of Geographical Information Science*, 34(1):98–118, 2020.

[20] D. R. Roberts, V. Bahn, S. Ciuti, M. S. Boyce, J. Elith, G. Guillera-Arroita, S. Hauenstein, J. J. Lahoz-Monfort, B. Schröder, W. Thuiller, et al. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40:913–929, 2017.

[21] A. Brenning. Spatial cross-validation and bootstrap for the assessment of prediction rules in remote sensing: The r package sperrorest, 2012. Paper presented at the 2012 IEEE international geoscience and remote sensing symposium, IEEE, Munich, 5372–5375 July 2012.

[22] Júlio Hoffimann, Maciel Zortea, Breno de Carvalho, and Bianca Zadrozny. Geostatistical learning: Challenges and opportunities. *Frontiers in Applied Mathematics and Statistics*, 7, 2021.

[23] Jeremy Aldworth and Noel Cressie. Sampling designs and prediction methods for gaussian spatial processes. In *Multivariate analysis, design of experiments, and survey sampling*, pages 25–78. CRC Press, 1999.