Université Paris Nanterre

200 av. de la République
92001 Nanterre Cedex
www.parisnanterre.fr

ICIMAF

Membre de l'université Paris Lumières
École doctorale 139 : Connaissance, langage, modélisation
MODAL'X UMR 9023

Cristina Olimpia CHÁVEZ CHONG

Méthodes de validation croisée pour des données spatialement dépendantes

Thèse présentée et soutenue publiquement le 11/04/2025

en vue de l'obtention du doctorat de Mathématiques appliquées et applications des mathématiques de l'Université Paris Nanterre

sous la direction de Mme Cécile HARDOUIN (Université Paris Nanterre),

de Mme. Ana Karina FERMIN RODRIGUEZ (Université Paris Nanterre)

et de M. Jesús Eladio SÁNCHEZ GARCÍA (ICIMAF)

Jury * **:**

Rapporteur·e :	Mme. Anne GÉGOUT-PETIT	Prof., Université de Lorraine, Institut Elie Cartan de Lorraine
Rapporteur·e :	M. Éric MATZNER-LØBER	Prof., Centre de Recherche en Economie et Statistique (CREST), ENSAE
Membre du jury :	Mme. Liliane BEL	Prof., AgroParisTech/INRAE UMR 518 Math. Info. Appli.
Membre du jury :	M. Thierry DUMONT	MCF, Université Paris Nanterre
Président du jury :	M. Jean-Marc BARDET	Prof., Université Paris 1 Panthéon Sorbonne SAMM (UR 4543)

Abstract

Data with spatial structure have become increasingly prominent in fields ranging from environmental science and ecology to urban planning, driven by advances in technology and open-data initiatives. However, conventional cross-validation techniques often underestimate predictive risk when observations are spatially correlated, owing to an implicit assumption of independence. In this thesis, we develop and evaluate spatially adapted cross-validation methods that maintain sufficient separation between training and validation sets, thus providing more accurate estimates of model risk.

We begin by reviewing various spatial cross-validation techniques, including spatial blocked cross-validation and spatial leave-one-out, under scenarios of low to strong spatial dependence. We then propose a practical framework for determining an optimal "buffer size" that reduces autocorrelation between training and validation subsets. This framework is further enhanced by a parametric bootstrap approach designed to approximate the true (or "ideal") risk in single-realization settings. Simulation experiments confirm that these methods effectively capture the underlying spatial structure, leading to more reliable risk estimation.

Next, we address multicollinearity in spatial autoregressive models by introducing a Ridge-type regularization procedure, where the tuning parameter is selected through a dedicated spatial cross-validation strategy. Our approach preserves the spatial correlation assumptions while mitigating parameter instability. Through simulation studies and an application to COVID-19 data in France, we demonstrate the flexibility and robustness of this new method for real-world spatial processes.

Overall, the thesis highlights the importance of adapting cross-validation to spatial data, both for straightforward risk estimation in linear models and for regularized spatial autoregressive frameworks. The proposed techniques offer practical tools for researchers and practitioners, enabling them to obtain more reliable predictive assessments when analyzing spatially dependent datasets.

Résumé

Les données à structure spatiale occupent une place de plus en plus importante dans des domaines aussi variés que les sciences de l'environnement, l'écologie ou l'urbanisme, stimulées par les avancées technologiques et les initiatives de données ouvertes. Cependant, les techniques conventionnelles de validation croisée sous-estiment souvent le risque prédictif lorsque les observations sont spatialement corrélées, en raison d'une hypothèse implicite d'indépendance. Dans cette thèse, nous développons et évaluons des méthodes de validation croisée adaptées au contexte spatial, qui assurent une séparation suffisante entre les ensembles d'entraînement et de validation, permettant ainsi d'obtenir des estimations du risque du modèle plus précises.

Nous commençons par examiner diverses techniques de validation croisée spatiale, telles que la validation croisée par blocs spatiaux et la validation croisée de type leave-one-out, dans des scénarios allant d'une dépendance spatiale faible à forte. Nous proposons ensuite un cadre pratique permettant de déterminer une taille de zone tampon optimale, laquelle réduit l'autocorrélation entre les sous-ensembles d'entraînement et de validation. Ce cadre est par ailleurs renforcé par une approche de bootstrap paramétrique conçue pour approcher le risque réel (ou idéal) dans des contextes de réalisation unique. Des expériences de simulation confirment que ces méthodes capturent efficacement la structure spatiale sous-jacente, conduisant ainsi à une estimation du risque plus fiable.

Nous abordons ensuite la question de la multicolinéarité dans les modèles autorégressifs spatiaux en introduisant une procédure de régularisation de type Ridge, où le paramètre de réglage est sélectionné au moyen d'une stratégie dédiée de validation croisée spatiale. Notre approche préserve les hypothèses de corrélation spatiale tout en atténuant l'instabilité des paramètres. Grâce à des études de simulation et à une application aux données de la COVID-19 en France, nous démontrons la flexibilité et la robustesse de cette nouvelle méthode pour les processus spatiaux en conditions réelles.

Dans l'ensemble, cette thèse souligne l'importance d'adapter la validation croisée aux données spatiales, que ce soit pour une estimation directe du risque dans des modèles linéaires ou pour des cadres autorégressifs spatiaux régularisés. Les techniques proposées

offrent ainsi des outils pratiques aux chercheurs et aux praticiens, leur permettant d'obtenir des évaluations prédictives plus fiables lors de l'analyse de jeux de données présentant une dépendance spatiale.

Acknowledgements

The work contained in this document wouldn't exist without the support, collaboration, and generosity of many people and institutions. Before presenting the results of this research, I would like to express my heartfelt gratitude to everyone who has accompanied me on this long and transformative path.

I would like to begin by extending my sincere thanks to Josué for encouraging me to pursue a PhD in France, you made such a compelling case that here I am today. I am also deeply grateful to Stéphane Mischler for forwarding my introductory email to potential advisors, and to Cécile Hardouin and Ana Karina Fermin, the two intrepid advisors who not only accepted me as a student but also guided me with unwavering commitment throughout this journey. I will be forever thankful for your trust and support.

My gratitude also goes to Thierry Dumont, Emilie Lebarbier, Liliane Bel, and Fabrice Tricou for serving as members of my "'comité de suivi" over the years, offering invaluable insights that have enriched my perspective. I would also like to thank Lisandro Fermin for the thought-provoking conversations about the subjects of my thesis.

I am equally grateful to El Mehddi, Karolina, Ellie and Théotime for the smiles and engaging conversations during our coffee breaks. I also wish to thank Richard, my flatmate during the pandemic, whose support and empathy were essential during those uncertain times. To Charly and Marian, thank you for the aperitifs, walks around Paris, and all the invitations, each one has meant the world to me. And to the community of Cuban mathematicians in France, spending time with you always felt like bathing in the warm Cuban sun.

I am grateful to Jesús E. Sánchez for his indispensable guidance, to Erwan LePennec for the his generous hospitality, and to Patrice Bertail for his readiness to offer help and advice.

My deepest gratitude to my family, my grandmother, Fernando, my brother, for your unwavering support, belief in me, and for being my safe haven. To Chiqui and Beló, I carried you in my thoughts every step of the way. And to my mom, whose presence is a constant source of comfort and inspiration, it's no wonder my brother once said that if he

were lost in a forest, he'd sit and wait for you to come pick him up, I am so fortunate to have you.

To France, the country that welcomed me and taught me new ways of perceiving both the world and academia; the Embassy of France in Cuba; Labex MME-DII (ANR11-LBX-0023-01); TELECOM Paris Tech; Projet MATH AmSud 51754ZK - SMILE; the Laboratoire SAMM at Université Paris 1 Panthéon-Sorbonne; and the Réseau Thématique MAIAGES (CNRS 2179), whose financial support made many aspects of this research possible. I also thank the Université Paris Nanterre for becoming my intellectual home and providing me with a solid platform for my professional development, as well as Modal'X-UMR 9023 for opening its doors, fostering my academic growth, and supporting my stays and research materials.

I would like to express my sincere gratitude to all the members of the jury for agreeing to serve on this panel. A special thank you goes to the rapporteurs, Prof. Anne Gégout-Petit and Prof. Éric Matzner-Løber for their willingness to participate in this evaluation. I also appreciate the commitment of the examiners, Dr. Thierry Dumont, Prof. Liliane Bel and Prof. Jean-Marc Bardet for accepting this responsibility.

Each of you has contributed to shaping this project, which has been enriched by a multicultural experience and consolidated by the generosity of those who shared their time, knowledge, and affection in support of my academic journey. Thank you for believing in me, for uplifting me during difficult times, and for standing by me in every achievement. Without your presence, this thesis, born of several years of work between two countries—would not have been possible.

Thank you all, infinitely!

Contents

Abstract vi

Acknowledgements xii

Introduction 2

1 Cross-Validation for spatial data 7

1.1 Preliminaries . . . 7

1.1.1 Regression problem in fixed design setting . . . 7

1.1.2 Classical cross-validation . . . 9

1.1.3 Basic principle of spatial cross-validation . . . 10

1.2 Spatial cross-validation methods . . . 14

1.2.1 Spatial Leave-One-Out . . . 14

1.2.2 Spatial K-fold cross-validation . . . 17

1.2.3 Blocked cross-validation . . . 20

1.2.4 Comparison of cross-validation strategies . . . 25

2 Determination of optimal buffer size 40

2.1 Ideal risk and spatial cross-validation risk discrepancy . . . 40

2.2 "Ideal" buffer size . . . 41

2.3 A naive choice of h . . . 45

2.3.1 The variogram-range method . . . 45

2.3.2 The Moran's I Method . . . 46

2.4 The SAC-0.1 method . . . 46

2.5 Comparison of the SAC-0.1 method with other approaches 69
2.5.1 SLOO Case 70
2.5.2 SBCV Case 72

3 Refined optimal buffer determination **77**
3.1 Sample spatial leave-one-out (SSLOO) 77
3.2 Parametric Bootstrap 80
3.3 Ideal bootstrap buffer size 83
3.3.1 Sample spatial leave-one-out with bootstrap 83
3.3.2 Spatial blocked cross-validation with bootstrap 85
3.4 Application 88
3.5 Conclusions 92

4 Ridge regularization for spatial autoregressive models **95**
4.1 Introduction 96
4.2 Spatial autoregressive models 98
4.3 Estimation 100
4.3.1 Spatial autoregressive models estimation 100
4.3.2 Ridge regression for spatial autoregressive models 101
4.3.3 Spatial cross-validation for selecting the Ridge parameter 103
4.4 Importance of the explanatory variables 104
4.5 Simulation experiments 106
4.6 Application 110
4.6.1 Data description 111
4.6.2 Exploratory analysis 112
4.6.3 Results 114
4.7 Conclusions 118

Conclusions and perspectives **120**

Introduction

The abundance of data with spatial structure has emerged as a valuable resource across diverse fields such as environmental science, ecology, epidemiology, and urban planning. Thanks to technological advancements, governmental policies (Cornes et al., 2018), and open data initiatives, new and rich sources of spatial data are increasingly accessible. However, the analysis, learning, and accurate prediction of such data pose unique challenges that conventional statistical techniques struggle to address (Reichstein et al., 2019). While the theory of statistical learning and its applications have undergone extensive investigation (Massart, 2007; Hastie et al., 2009; Bach, 2020), their extension to dependent processes, particularly in the spatial context, remains limited.

Numerous tasks in statistical learning require the estimation of the risk of an estimator (Vapni, 2000). From model selection to model assessment, having reliable estimators of the risk is fundamental for guiding decisions and ensuring the robustness of predictive models. A well-known approach for risk estimation is cross-validation (Bates et al., 2021), a resampling method that estimates the prediction error of a model by partitioning the data into subsets, fitting the model to some of these subsets (training data), and evaluating its performance on the remaining subsets (validation data).

Cross-validation is widely used in statistical learning for risk estimation and for hyper-parameter or model selection (Arlot et al., 2010). In the context of independent and identically distributed (i.i.d.) data, cross-validation methods have well-understood properties. For example, Burman (1989) demonstrated that, in the regression framework, the leave-one-out cross-validation estimator asymptotically exhibits minimal bias and variance. Moreover, Breiman and Spector (1992) showed that cross-validation provides consistent risk estimation in regression models.

However, these desirable properties rely on the assumption that data observations are independent. In the presence of spatial, temporal, or other forms of structured dependence, this assumption does not hold. In particular, when observations are spatially dependent, standard cross-validation can lead to overly optimistic risk estimates because the training and validation sets may share spatially correlated information, violating the independence

assumption (Roberts et al., 2017).

According to Tobler's First Law of Geography, a fundamental principle in geostatistical analysis, "everything is related to everything else, but near things are more related than distant things" (Tobler, 1970). As discussed before, cross-validation works by dividing the data repeatedly into training and test set, frequently this partitioning is carried out randomly, which is problematical in the presence of dependence structures, random data splitting does not provide independent validation. This leads to models performing apparently well and users trusting their prediction capabilities based on the cross-validation results. Thus, cross-validation needs to be modified to deal with data exhibiting this characteristics.

In recent years, there has been growing interest in developing cross-validation methods adapted to spatial settings, the review by Jemeljanova et al. (2024) found over 26 different variants of spatial cross-validation techniques. These methods aim to provide accurate risk estimation in spatially dependent contexts, accounting for the challenges posed by non-independent observations. For example, Roberts et al. (2017) proposed spatial blocked cross-validation methods, which involve partitioning the data into spatially contiguous blocks to reduce spatial autocorrelation between training and validation sets. Similarly, Valavi et al. (2019) developed the `blockCV` package in `R` for spatial and environmental data, providing tools for spatial cross-validation and model evaluation.

Applied studies across various disciplines have started to adopt spatial cross-validation techniques. In agricultural and food sciences, Killeen et al. (2024) used k-means based spatial cross-validation to identify over-optimistic yield prediction models. In environmental science, Melton et al. (2022) used spatial cross-validation to estimate the model error of a spatially continuous global map of peatland fractional coverage. In epidemiology, Dhingra et al. (2016) utilized spatial cross-validation to assess the extrapolation accuracy of a global suitability model of highly pathogenic avian influenza (HPAI) H5N1.

Despite these advancements, classical cross-validation techniques are used in spatial contexts, sometimes leading to misleading conclusions and there is still debate around the which method cross-validation method to use, depending on the objectives of the work. For instance, Wadoux et al. (2021) indicated that for map validation, cross-validation with random selection of the validation set can yield less biased estimates of the root mean squared error (RMSE) compared to specialized techniques like spatial leave-one-out and spatial blocked cross-validation, which are tailored for spatial data. The affirmation by Wadoux et al. (2021) is true if we know that sample data adequately represent the prediction area, particularly when samples are uniformly distributed across the area.

In this thesis, we focus on cross-validation methods adapted to spatial settings. The goal of this work is to develop and evaluate methods that provide accurate risk estimation

in spatially dependent contexts, accounting for the challenges posed by non-independent observations.

The remainder of the thesis is organized as follows. Chapter 1 clarifies the statistical setting that we will use in Chapters 1, 2 and 3. It presents a review of the literature on cross-validation methods in spatial contexts, focusing on recent developments specifically designed to address spatial dependence. The main spatial cross-validation methods are then compared through a simulation study considering data generated over a rectangular lattice $\mathcal{S}_n$ as follows,

$$Y(s) = \mathbf{X}(s)\boldsymbol{\beta} + \varepsilon(\mathbf{s}), \quad \mathbf{s} = (x_1, x_2) \in \mathcal{S}_n, \tag{0.0.1}$$

where $\boldsymbol{\varepsilon}$ is generated according to a zero-mean Gaussian spatial process model with an isotropic exponential covariance function defined by $C(h) = \sigma^2 \exp\left(-\frac{h}{\theta}\right)$. We consider different scenarios of spatial dependence: low, medium, and high.

Building on the findings from the previous chapter, Chapter 2 focuses on a key aspect of spatial cross-validation, the concept of a buffer zone, a spatial separation imposed between the training and validation sets to reduce their dependence. Within this framework, the optimal buffer size is then defined as the distance that minimizes the discrepancy between the spatial cross-validation risk estimate and the empirical "ideal" risk. The chapter then explores three key methods for determining this optimal buffer zone's size:

1. A *variogram-range* approach from Le Rest et al. (2014), which identifies the distance at which spatial dependence effectively disappears, assuming a particular variogram model.
2. A *Moran's Index* method from Karasiak et al. (2022), which leverages a statistically rigorous criterion to determine when autocorrelation becomes negligible.
3. The *SAC-0.1 method*, proposed by us, which sets a fixed spatial auto-correlation (SAC) threshold 0.1, which is close to the optimal buffer size.

All three methods estimate the buffer size by analyzing the spatial dependence in the model's residuals. Through numerical experiments, Chapter 2 compares their performance in terms of the discrepancy between the cross-validation risk and the "ideal" risk, highlighting scenarios in which these methods excel or encounter limitations. While each approach can be effective under certain conditions, they remain relatively "naive" because they do not directly optimize for the buffer size that best aligns the cross-validation risk with the ideal risk, an issue addressed in the subsequent chapter.

Chapter 3 addresses this gap by introducing a parametric bootstrap method designed to approximate the ideal risk, and the corresponding buffer size, directly. Fitting a spatial

covariance model to the residuals and generating multiple bootstrap replicates circumvents the practical limitations of having only one realization of a spatial process. The procedure is validated through simulation studies, demonstrating its ability to approximate the "ideal" buffer size consistently across diverse settings. Additionally, strategies for reducing computation time, such as subsampling schemes, are proposed and shown to maintain accuracy at a fraction of the usual cost. The chapter concludes by applying these methods to a benchmark simulation scheme inspired by a model for species abundances, illustrating the robustness of this new approach even when the underlying linear model is partially misspecified.

The conventional statistical approach to address spatial dependence in data involves using parametric models that incorporate corresponding dependence structures, as exemplified in the works by Anselin (1988) and Diggle and Ribeiro (2007). Chapter 4 proposes a new method for building an explanatory spatial autoregressive model in a multicollinearity context. We consider the models,

$$\mathbf{Y} = \rho W \mathbf{Y} + \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_n), \tag{0.0.2}$$

and

$$\begin{aligned} \mathbf{Y} &= \mathbf{X}\boldsymbol{\beta} + \mathbf{u}, \\ \mathbf{u} &= \lambda W \mathbf{u} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_n). \end{aligned} \tag{0.0.3}$$

where the spatial weights matrix $W = (w_{ij})_{1 \leq i,j \leq n}$ reflects an underlying neighbourhood graph. We consider the framework of highly colinear covariates and we use Ridge regularization to address this issue. We present new estimation algorithms that allow for the estimation of the regression coefficients as well as the spatial dependence parameter. A spatial cross-validation procedure is used to tune the regularization parameter. Variable importance is assessed by permutation tests since classical tests are not valid after Ridge regularization. We assess the performance of our methodology through numerical experiments conducted on simulated data. Finally, we apply our method to a real dataset to evaluate the impact of some socio-economic variables on the COVID-19 intensity in France.

Finally, we conclude the thesis with a summary of the main findings, we discuss the implications of the results, and outline potential avenues for future research. We also reflect on the limitations of the current work and suggest possible extensions to further enhance the robustness of risk estimation in spatially dependent settings.

Chapter 1

Cross-Validation for spatial data

1.1 Preliminaries

1.1.1 Regression problem in fixed design setting

The classical regression problem involves estimating an unknown function f based on observed data $\mathcal{D}_n = \{(\mathbf{X}_i, Y_i), i = 1, \ldots, n\}$. This problem can be considered under two different frameworks: the *fixed design* setting and the *random design* setting.

In the fixed design setting, the $\mathbf{X}_i$ are considered fixed and non-random, and the randomness arises solely from the noise in the observations. The model is defined as,

$$Y_i = f(\mathbf{X}_i) + \varepsilon_i, \quad i = 1, \ldots, n, \tag{1.1.1}$$

where $\mathbf{X}_i \in \mathbb{R}^{p+1}$ is the covariate vector for the i-th observation, which includes a constant term (intercept) and p fixed covariates, $Y_i \in \mathbb{R}$ are the realizations of the dependent variable, ε_i are independent and identically distributed (i.i.d.) noise terms with $\mathbb{E}[\varepsilon_i] = 0$ and $\mathbb{E}[\varepsilon_i^2] = \sigma^2$.

In contrast, the random design setting treats the covariates $\mathbf{X}_i$ as random variables drawn from a probability distribution P_X. Bach (2020) discusses empirical risk minimization for regression problems for both random and fixed design. In this work, we focus on the fixed design setting. We are particularly interested in the case where f is linear in some parameters $\boldsymbol{\beta}$, and of the form $f(\mathbf{X}) = \mathbf{X}\boldsymbol{\beta}$, where $\boldsymbol{\beta} \in \mathbb{R}^{p+1}$ is a vector of coefficients to be estimated.

The linear regression model in equation (1.1.1) can be rewritten in matrix form,

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1.1.2}$$

where $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_n)^\top \in \mathbb{R}^n$ is the vector of responses, $\mathbf{X} \in \mathbb{R}^{n\times(p+1)}$ is the design matrix, with the i-th row equal to $\mathbf{X}_i$ and $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)^\top$ is the vector of noise terms.

Under these assumptions, we address the regression problem by obtaining an estimator $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ by minimizing the fixed design *risk*,

$$\mathcal{R}(\boldsymbol{\beta}) = \mathbb{E}_{\mathbf{Y}}\left[\frac{1}{n}\sum_{i=1}^{n}\ell(Y_i, \mathbf{X}_i\boldsymbol{\beta})\right] \tag{1.1.3}$$

where $\ell : \mathbb{R}\times\mathbb{R} \longleftarrow \mathbb{R}$ is called the *loss function*. In what follows we will consider the quadratic loss function $\ell(\mathbf{Y}, \mathbf{Y}') = (\mathbf{Y} - \mathbf{Y}')^2$. In the regression framework, the quadratic loss is particularly useful for several reasons. The quadratic loss is convex and differentiable, which simplifies the optimization problem and allows for its derivation of explicit solutions.

In this work, we are interested in estimating the predictive risk associated with an estimator $\hat{\boldsymbol{\beta}}$ obtained from the sample set $\mathcal{D}_n = \{(\mathbf{X}_i, Y_i), i = 1, \ldots, n\}$, focusing particularly on its performance in predicting new observations. This problem has been extensively studied in the statistical literature, with significant contributions by Breiman (1992) in the fixed design setting and more recently by Rosset and Tibshirani (2020) in the random design context.

Suppose that $\mathbf{Y}^{\text{new}} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}^{\text{new}}$ represents a new independent realization of the dependent variable, with the same fixed design matrix $\mathbf{X}$ and a new independent noise $\boldsymbol{\varepsilon}^{\text{new}}$, where $\boldsymbol{\varepsilon}^{\text{new}}$ has the same distribution as $\boldsymbol{\varepsilon}$ and is independent of it. The *predictive risk* associated with an estimator $\hat{\boldsymbol{\beta}}$ obtained from the data $\mathcal{D}_n$ is then defined as,

$$\mathcal{R}_{\text{pred}}(\hat{\boldsymbol{\beta}}) = \mathbb{E}_{\mathbf{Y},\mathbf{Y}^{\text{new}}}\left[\frac{1}{n}\left\|\mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{Y}^{\text{new}}\right\|^2\right], \tag{1.1.4}$$

where the subscripts on the expectation indicate the random variables over which the expectation is taken.

However, in practice, since the true parameter $\boldsymbol{\beta}$ and the noise terms are unknown, and we typically do not have access to new independent data $\mathbf{Y}^{\text{new}}$, we cannot compute $\mathcal{R}_{\text{pred}}(\hat{\boldsymbol{\beta}})$ directly. Instead, we often estimate the predictive risk using an empirical approximation based on the observed data.

$$\mathcal{R}_n(\hat{\boldsymbol{\beta}}) = \frac{1}{n}\left\|\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right\|^2. \tag{1.1.5}$$

The empirical risk $\mathcal{R}_n(\hat{\boldsymbol{\beta}})$ defined in equation (1.1.5) measures the average squared error between the observed responses $\mathbf{Y}$ and the predictions $\mathbf{X}\hat{\boldsymbol{\beta}}$ on the available data $\mathcal{D}_n$. From now on we will write $\mathcal{R}_n$ instead of $\mathcal{R}_n(\hat{\boldsymbol{\beta}})$ to simplify notations.

One common way to estimate the predictive risk is through cross-validation. Cross-validation involves partitioning the observed data into training and validation sets multiple times, effectively mimicking the process of predicting new independent observations. By training the model on one subset of the data and validating it on another, we can obtain an unbiased estimate of the model's predictive performance.

In the following section, we explore classical cross-validation techniques used for risk estimation in regression analysis.

1.1.2 Classical cross-validation

As stated by Arlot et al. (2010) the universality of the data splitting heuristics, attracts a major interest to cross-validation since this method only assumes that data are identically distributed, and training and validation samples are independent.

We start with the so-called *hold-out* (HO) procedure. Let us divide the sample set $\mathcal{D}_n = \{(\mathbf{X}_i, Y_i), i = 1, \ldots, n\}$ into two separate independent non-empty subsets $\mathcal{D}_n^{\text{train}}$ and $\mathcal{D}_n^{\text{val}}$ such that $\mathcal{D}_n^{\text{train}} \cup \mathcal{D}_n^{\text{val}} = \mathcal{D}_n$ and $\mathcal{D}_n^{\text{train}} \cap \mathcal{D}_n^{\text{val}} = \emptyset$. $\mathcal{D}_n^{\text{train}}$ is called the training set and is used to estimate $\boldsymbol{\beta}$, while $\mathcal{D}_n^{\text{val}}$ is called the validation set and is used to estimate the risk. From now, we write $\mathcal{D}_n^{\text{t}}$ for $\mathcal{D}_n^{\text{train}}$, $\mathcal{D}_n^{\text{v}}$ for $\mathcal{D}_n^{\text{val}}$, and $\hat{\boldsymbol{\beta}}^{\text{t}}$ instead of $\hat{\boldsymbol{\beta}}^{\text{train}}$ to simplify notations.

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

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

Then we test the model's prediction ability induced by $\hat{\boldsymbol{\beta}}^{\text{t}}$ by computing the hold-out empirical risk:

$$\mathcal{R}_n^{\text{HO}} = \mathcal{R}_n^{\text{V}} = \frac{1}{\text{Card}(\mathcal{D}_n^{\text{v}})} \sum_{(\mathbf{X}_i, Y_i) \in \mathcal{D}_n^{\text{v}}} (Y_i - \mathbf{X}_i \hat{\boldsymbol{\beta}}^{\text{t}})^2. \tag{1.1.6}$$

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* (KCV) algorithm is closely related to the hold-out method. It partitions $\mathcal{D}_n$ into K subsets $\{\mathcal{D}_1, \ldots, \mathcal{D}_K\}$, with observations randomly assigned to each subset, but ensuring that the sizes of the subsets are approximately equal, that is $\text{Card}(\mathcal{D}_m) = n_m$, with $\left\lfloor \frac{n}{K} \right\rfloor \leq n_m \leq \left\lceil \frac{n}{K} \right\rceil$. For $m = 1, \ldots, K$, successively, the hold-out risk $\mathcal{R}_n^{\text{V}_m}$ defined in (1.1.6) is calculated on $D_n^{\text{v}} = \mathcal{D}_m$, with $\boldsymbol{\beta}$ estimated by $\hat{\boldsymbol{\beta}}^{\text{t}_m}$ using $D_n^{\text{t}} = \bigcup_{i \neq m} \mathcal{D}_i$. The K-fold cross-validation empirical risk is the average of the K hold-out

empirical risks:

$$\mathcal{R}_n^{\mathrm{KCV}} = \frac{1}{K}\sum_{m=1}^{K} \mathcal{R}_n^{\mathrm{v}_m}. \tag{1.1.7}$$

K-fold cross validation is usually preferred over the hold-out method because it trains the model on K different training-validation subsets splits. This provides a better indication of how well the model performs on unseen data. Moreover, it has been shown that the estimates obtained by minimizing (1.1.7) improve in terms of both bias and variance compared to those obtained via simple hold-out (see for instance James et al. (2013)).

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 to train the model. For $m \in \{1, \dots n\}$, we have $\mathcal{D}_n^{\mathrm{t}_m} = \{(\mathbf{X}_m, Y_m)\}^c$, the complement of the singleton $\mathcal{D}_n^{\mathrm{v}_m} = \{(\mathbf{X}_m, Y_m)\}$. The leave-one-out cross-validation risk is then given by:

$$\mathcal{R}_n^{\mathrm{LOO}} = \frac{1}{n}\sum_{m=1}^{n} \mathcal{R}_n^{\mathrm{v}_m} = \frac{1}{n}\sum_{m=1}^{n} (Y_m - \mathbf{X}_m \hat{\boldsymbol{\beta}}^{\mathrm{t}_m})^2, \tag{1.1.8}$$

$\hat{\boldsymbol{\beta}}^{\mathrm{t}_m}$ being estimated in $\mathcal{D}_n^{\mathrm{t}_m}$.

1.1.3 Basic principle of spatial cross-validation

Classical cross-validation assumes that data are identically distributed and that training and validation samples are independent. However, in the context of spatial data, as noted by Tobler (1970), the behavior of a variable at a certain location is influenced by its neighbors due to spatial autocorrelation. Consequently, randomly splitting the data does not guarantee independence between training and validation sets, which can lead to overly optimistic risk estimates and inflated assessments of model performance.

Our illustration in Figure 1.1 displays the empirical risks under the following framework. We consider the general linear model $Y(\mathbf{s}) = \mathbf{X}(\mathbf{s})\boldsymbol{\beta} + \varepsilon(\mathbf{s})$ where $\mathbf{X}(\mathbf{s})\boldsymbol{\beta}$ represents the large-scale, deterministic mean structure of the process, and $\varepsilon(\mathbf{s})$ is the small-scale stochastic component modeling the spatial dependence among the data. We assume $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. We generate 100 datasets and estimate $\boldsymbol{\beta}$ using the ordinary least squares procedure. Then we compute $\mathcal{R}_n^{\mathrm{KCV}}$ and its spatial version $\mathcal{R}_n^{\mathrm{SKCV}}$ with $K = 200$ and display the 100 results in boxplots diagrams.

Let us denote $\|\boldsymbol{\alpha}\|^2 = \boldsymbol{\alpha}^\top \boldsymbol{\alpha} = \sum_{i=1}^n \alpha_i^2$ for an $n-$dimensional vector $\boldsymbol{\alpha}$. The *ordinary least squares* (OLS) estimator is defined by:

$$\hat{\boldsymbol{\beta}}_{\mathrm{O}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\operatorname{argmin}} \left(\frac{1}{n} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 \right) = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{Y}, \tag{1.1.9}$$

provided that $\mathbf{X}^\top\mathbf{X}$ is invertible. When the noise terms are assumed to be normally distributed, minimizing the quadratic loss corresponds to maximum likelihood estimation.

The estimates are derived from 100 simulated datasets generated under the model $Y(\mathbf{s}) = \mathbf{X}(\mathbf{s})\boldsymbol{\beta}+\varepsilon(\mathbf{s})$, where the error terms $\boldsymbol{\varepsilon}$ are generated according to a zero-mean Gaussian spatial process with an isotropic exponential covariance function $C(h) = \sigma^2 \exp\left(-\frac{h}{\theta}\right)$. More details about the simulation design are provided in Section 1.2.4.

The left panel of Figure 1.1 displays results for $\theta = 3$, representing low spatial dependence, while the right panel shows results for $\theta = 12$, indicating high spatial dependence. In both cases, we observe that the risk estimates obtained using traditional K-fold cross-validation are lower than those obtained using Spatial K-fold cross-validation. This suggests that spatial cross-validation methods provide less optimistic risk estimates compared to traditional methods. However, at this stage, the validity of these spatial cross-validation estimates remains to be determined.

These initial observations motivate the need to further investigate the effectiveness of spatial cross-validation methods in providing accurate and reliable risk estimates in the presence of spatial autocorrelation. Throughout this chapter, we will explore these methods and evaluate their performance through simulation studies, aiming to establish their validity and practical utility in spatial data analysis.

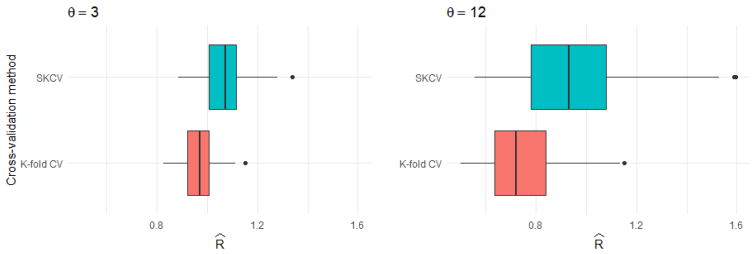


Figure 1.1: Risk estimates associated with the OLS estimator obtained using K-fold CV and Spatial K-fold CV (SKCV) methods, with $K = 200$.

Let us consider a spatial domain $\mathcal{S}_n = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\} \subset \mathbb{R}^d$, where our sample data $\mathcal{D}_n = \{(\mathbf{X}(\mathbf{s}_i), Y(\mathbf{s}_i)) \,|\, \mathbf{s}_i \in \mathcal{S}_n\}$ are collected. We consider the following regression model:

$$Y(\mathbf{s}) = \mathbf{X}(\mathbf{s})\boldsymbol{\beta} + \varepsilon(\mathbf{s}), \tag{1.1.10}$$

where $\mathbf{X}(\mathbf{s})\boldsymbol{\beta}$ represents the large-scale, deterministic mean structure of the process, and $\varepsilon(\mathbf{s})$ is the small-scale stochastic component modeling the spatial dependence among the data. Typically, we assume $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Let us note that this general formulation allows for non-stationarity in the process, but it also encompasses the stationary case when $\mathbf{X}(\mathbf{s}) = \text{const}$.

Suppose we wish to estimate the predictive risk of an estimator of $\boldsymbol{\beta}$ obtained from our sample data $\mathcal{D}_n$. When the errors exhibit correlation, the Ordinary Least Squares (OLS) estimator remains unbiased but is no longer efficient, and standard error estimates may be inconsistent, leading to invalid inference. To account for the covariance structure of the errors, we can use the *Generalized Least Squares* (GLS) estimator. Assuming that the covariance matrix $\boldsymbol{\Sigma}$ is known, the GLS estimator is defined as:

$$\hat{\boldsymbol{\beta}}_{\text{GLS}} = \left(\mathbf{X}^\top \boldsymbol{\Sigma}^{-1} \mathbf{X}\right)^{-1} \mathbf{X}^\top \boldsymbol{\Sigma}^{-1} \mathbf{Y}. \tag{1.1.11}$$

In practice, $\boldsymbol{\Sigma}$ is often unknown and must be estimated from the data, leading to the *Feasible GLS* (FGLS) estimator where $\hat{\boldsymbol{\Sigma}}$ is a consistent estimate of the true covariance matrix,

$$\hat{\boldsymbol{\beta}}_{\text{FGLS}} = \left(\mathbf{X}^\top \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{X}\right)^{-1} \mathbf{X}^\top \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{Y}. \tag{1.1.12}$$

However, even with GLS, the issue of spatial dependence affects the evaluation of predictive performance through cross-validation. Spatial cross-validation involves redefining how we partition the data within the spatial domain $\mathcal{S}_n$. The key modification revolves around the principle of "point separation", which involves strategically removing observations from the training set to reduce dependence between the training and validation sets. This is achieved by delineating three spatial domains: the validation domain, the training domain, and a separating domain called the "buffer" or "dead zone", that provides a buffer zone to mitigate spatial dependence.

Let us consider the isotropic case, where spatial independence between observations can be achieved by establishing a buffer zone at the same distance in all directions. Let us denote h the necessary distance between the validation domain and the training domain that satisfies the independence assumption. For a chosen validation domain $\mathcal{S}_n^{\text{v}} \subset \mathcal{S}_n$, we define a new subset of sites $\mathcal{S}_{n,h}^{\text{D}}$. This set consists of all sites that are not part of the validation domain but are within a distance h of at least one site in the validation domain. Formally, the dead zone or *buffer* is defined as,

$$\mathcal{S}_{n,h}^{\text{D}} = \{\mathbf{s}_j \notin \mathcal{S}_n^{\text{v}} : \ \exists\, \mathbf{s}_i \in \mathcal{S}_n^{\text{v}} : d(\mathbf{s}_j, \mathbf{s}_i) \leq h\}.$$

The training domain $\mathcal{S}_{n,h}^{\text{t}}$ then comprises all sites not in $\mathcal{S}_n^{\text{v}}$ or $\mathcal{S}_{n,h}^{\text{D}}$, and is represented as

$$\mathcal{S}_{n,h}^{\text{t}} = \mathcal{S}_n \setminus (\mathcal{S}_n^{\text{v}} \cup \mathcal{S}_{n,h}^{\text{D}}).$$

It is important to note that the choice of distance h is crucial as it determines the effectiveness of achieving independence between the training and validation domains. Therefore, determining h accurately is one of the primary objectives of this dissertation.

The process can be outlined algorithmically as follows. Assuming h has been determined, we proceed with the m-th iteration of the cross-validation procedure in three steps:

1. **Select the validation set:** We choose the validation domain $\mathcal{S}_n^{\mathrm{v}m}$ for this iteration, and thus define the validation set $\mathcal{D}_n^{\mathrm{v}m} = \{(\mathbf{X}(\mathbf{s}_i), Y(\mathbf{s}_i)) : s_i \in \mathcal{S}_n^{\mathrm{v}m}\}$.
2. **Establish the "dead zone":** We define $\mathcal{S}_{n,h}^{\mathrm{D}m}$.
3. **Define the training set:** The corresponding training set is $\mathcal{D}_{n,h}^{\mathrm{t}m} = \{(\mathbf{X}(\mathbf{s}_i), Y(\mathbf{s}_i)) : \mathbf{s}_i \in \mathcal{S}_{n,h}^{\mathrm{t}m}\}$, where $\mathcal{S}_{n,h}^{\mathrm{t}m}$ represents the training domain for this iteration.

It is important to note that while the validation domain's specific structure is not detailed here, once h is determined, the process for selecting the dead zone and the training domain remains consistent. Then, different 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, where distances and separations are uniformly defined in all directions. Consequently, a single value of h is sufficient to delineate the minimal separation needed between the validation and training sets to ensure their independence.

However, in anisotropic scenarios where spatial dependence varies with direction, a simplistic approach with a single value of h may not be adequate. Instead, a more nuanced approach involving two or more values of h may be required to accurately capture spatial relationships. Figure 1.2 illustrates the dead zone taken around one point under anisotropic and isotropic conditions. On the left, the elliptical blue contour shows varying separation distances h_1 and h_2 in vertical and horizontal directions due to differing spatial dependencies in these directions. In contrast on the right, under isotropic conditions (depicted by the red circular contour), only a single radius h is necessary.

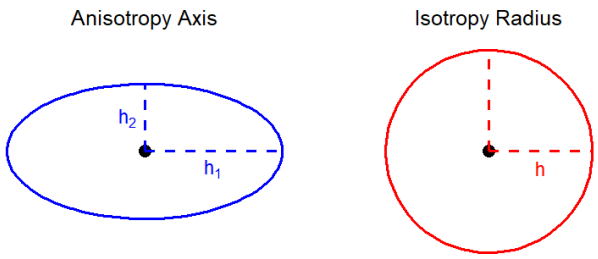


Figure 1.2: Examples of dead zone under anisotropic and isotropic conditions.

We now present adaptations of the main cross-validation procedures to the spatial context. To ease the presentation, we consider the isotropic framework in what follows.

1.2 Spatial cross-validation methods

1.2.1 Spatial Leave-One-Out

We begin by introducing Spatial Leave-One-Out (SLOO). This cross-validation procedure was initially presented by Le Rest et al. (2013) for model selection purposes. Back in 1994, Burman et al. (1994) introduced a modification of leave-one-out cross-validation tailored for general stationary data, known as h-block cross-validation, which laid the groundwork for Spatial Leave-One-Out. The primary goal of SLOO is to mitigate optimistic bias in the training set caused by spatial dependence. This is achieved by excluding data points "close" to the one held out for validation from the training set, as depicted in Figure 1.3. In this figure we represent $\mathcal{S}_n$, a lattice of size 80×60 with $n = 4800$ sites and its partition into the three domains for one iteration of SLOO. The validation set is represented by a unique red point, a dead zone of radius 8 is removed around the validation domain, and the remaining locations colored in blue conform the training domain.

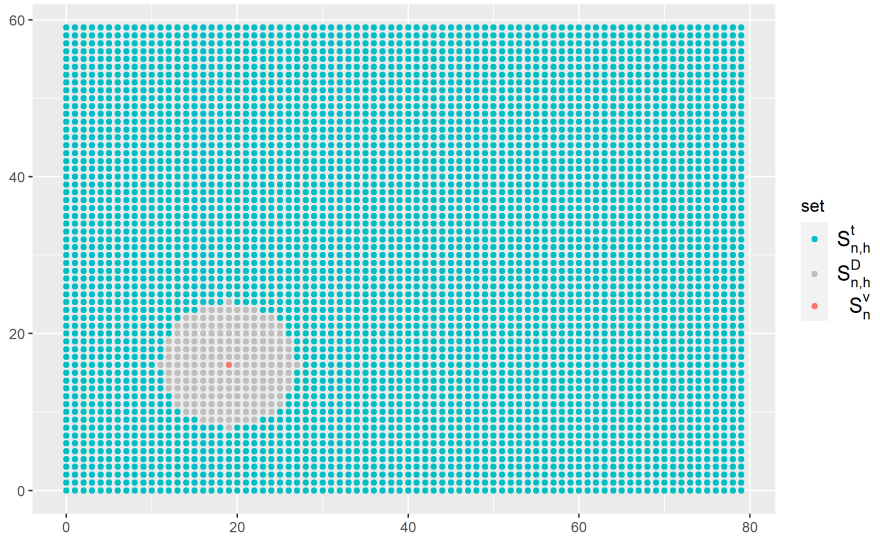


Figure 1.3: Dataset partition for one iteration of SLOO with a buffer of radius $h = 8$.

Let us precise the Spatial Leave-One-Out procedure within the regression framework, and assuming isotropy. We suppose that we know the distance h required to ensure independence between validation and training sets. At the $m-$th iteration of SLOO, one site is selected as the validation domain

$$\mathcal{S}_n^{\mathrm{v}m} = \{\mathbf{s}_m\},$$

and

$$\mathcal{D}_n^{\mathrm{v}m} = \{(\mathbf{X}(\mathbf{s}_m), Y(\mathbf{s}_m))\}$$

forms the validation set. The "dead zone" around $\{\mathbf{s}_m\}$ is defined as its neighbourhood of radius h,

$$\mathcal{S}_{n,h}^{\mathrm{D}_m} = \{\mathbf{s}_j \in \mathcal{S}_n : j \neq m \text{ and } d(\mathbf{s}_j, \mathbf{s}_m) \leq h)\}.$$

The remaining points constitute the training domain,

$$\mathcal{S}_{n,h}^{\mathrm{t}_m} = \{\mathbf{s}_j \in \mathcal{S}_n : d(\mathbf{s}_j, \mathbf{s}_m) > h\},$$

and the corresponding training set is

$$\mathcal{D}_n^{\mathrm{t}_m} = \{(\mathbf{X}(\mathbf{s}_j), Y(\mathbf{s}_j)) | \mathbf{s}_j \in \mathcal{S}_{n,h}^{\mathrm{t}_m}\}.$$

The predictive empirical risk is given by

$$\mathcal{R}_{n,h}^{\mathrm{SLOO}} = \frac{1}{n}\sum_{m=1}^{n} \mathcal{R}_{n,h}^{\mathrm{v}_m} = \frac{1}{n}\sum_{m=1}^{n} (Y(\mathbf{s}_m) - \mathbf{X}(\mathbf{s}_m)\hat{\boldsymbol{\beta}}_h^{\mathrm{t}_m})^2, \tag{1.2.1}$$

where we denote $\hat{\boldsymbol{\beta}}_h^{\mathrm{t}_m}$ the estimator computed on the training set $\mathcal{D}_{n,h}^{\mathrm{t}_m}$. The subscript h highlights the buffer or "dead zone" of radius that now needs to be considered.

For example, if we chose the OLS estimator, $\hat{\boldsymbol{\beta}}_h^{\mathrm{t}m}$ is defined by,

$$\hat{\boldsymbol{\beta}}_h^{\mathrm{t}m} = \left(\mathbf{X}_{m,h}^{\mathrm{T}}\mathbf{X}_{m,h}\right)^{-1}\mathbf{X}_{m,h}^{\mathrm{T}}\mathbf{Y}_{m,h}, \tag{1.2.2}$$

with $\mathbf{X}_{m,h}$ and $\mathbf{Y}_{m,h}$ representing the matrix $\mathbf{X}$ and vector $\mathbf{Y}$ whose elements related to $\mathcal{S}_n^{\mathrm{v}_m} \cup \mathcal{S}_{n,h}^{\mathrm{D}_m}$ have been deleted.

Spatial leave-one-out has proven effective in providing less biased estimates of predictive error. Research has shown its superiority over ordinary leave-one-out (LOO) cross-validation, particularly in spatial contexts. For example, Karasiak et al. (2022) demonstrated that SLOO yielded less biased predictive error estimates across various sample sizes when compared to ordinary K-fold cross-validation and LOO. They trained a random forest model to classify two forest classes using images from the French department (area) "Herault". Subsequently, they compared the predictive error estimates derived from SLOO, LOO, and K-fold cross-validation against those obtained from a separate, independent dataset of images from two other French areas, which are sufficiently distant from the Herault department. Their findings indicated that the predictive error estimated by SLOO closely matched the error estimated using the independent data.

Similarly, Misiuk et al. (2019) provided evidence of SLOO's advantages over LOO in estimating seabed sediment mapping using random forests. They compared the root mean square error, mean absolute error, and R^2 values in both interpolation and extrapolation areas, as estimated by SLOO and LOO, against the true values. Their results revealed that LOO tended to provide more optimistic estimates of performance metrics than SLOO.

Limitations

Spatial Leave-One-Out inherits two key limitations from the classical leave-one-out method. Firstly, it can be computationally expensive, particularly with large datasets or complex training models and learning algorithms. The iterative process of training and validating the model for each individual point in the dataset can demand substantial computational resources.

Secondly, SLOO tends to exhibit high variance in its hold-out estimations. This arises because the validation set in LOO and SLOO consists of only one data point, which leads to increased variability in performance metrics, compared to methods using larger validation sets.

As previously emphasized, the effectiveness of SLOO heavily relies on the accurate selection of the spatial buffer parameter h. Choosing an optimal h value requires a delicate balance: it must be small enough to retain sufficient observations in the training set for effective model calibration, yet expansive enough to ensure the necessary independence from the validation set.

Extensions

Due to its popularity, several extensions and modifications have been proposed to enhance its performance and applicability across different scenarios. Misiuk et al. (2019) introduced Spatially Resampled Leave-One-Out Cross-Validation (SR-LOO), which extends SLOO by incorporating spatial buffering procedures around both validation and training domains to prevent adjacent points from influencing model fitting or validation. This approach improves performance in handling spatial autocorrelation. The strategy was introduced to deal with random forests and builds upon the ideas from the variable scale selection procedure proposed by Holland et al. (2004) and the fact that separate "forests" can be combined and treated as a single model to make predictions (Liaw and Wiener (2002)). For each iteration of SR-LOO, besides the classical buffer around the point left out for validation, each training point is also spatially buffered so that no adjacent points are used for model fitting. To address the potential issue of significant data loss due to this strategy, they resample the training set multiple times and fit a small number of trees for each subsample. The final "forest" is constructed by combining the trees from the separate subsamples. As mentioned before, this method is particularly designed for random forests in scenarios where we aim to avoid overfitting on the training data because we wish to train a model that is good for extrapolation.

Mila et al. (2022) proposed the *Nearest Neighbour Distance Matching Leave-One-Out Cross-Validation* method for map validation. This method uses nearest neighbor distance

distribution functions to ensure that the spatial structure between training and validation sets matches that of the target area.

For the prediction target sites, a nearest neighbor distance distribution function is used to describe the distance between each prediction site and its nearest neighbor from the sample set. Specifically, the cross nearest neighbor cumulative distribution function $G_{i,j}(r)$ is defined as,

$$G_{i,j}(r) = P\left(d_{i,j} \leq r\right) = \frac{1}{N_i} \sum_{k=1}^{N_i} I\left(d_k^{(i,j)} \leq r\right),$$

where N_i is the number of prediction sites in dataset i, $d_k^{(i,j)}$ is the distance from the k-th prediction site in dataset i to its nearest neighbor in the sample set j and $I(\cdot)$ is the indicator function. This nearest neighbor distribution is replicated between the training and test sets using rejection sampling. The goal is to ensure that the distribution of distances from test sites to their nearest training sites matches the overall spatial structure of the target area.

This method only considers the spatial distribution of the data and does not take response variable or covariates into consideration. It refines SLOO by considering distances to the nearest neighbor at each iteration, which is especially useful for irregularly distributed locations. At each iteration, the distance of the validation point to its nearest neighbor is considered. If there is minimal or no spatial autocorelation, only one site is placed in the test set and the method behaves like SLOO. If the nearest-neighbor distance is larger than the spatial autocorrelation range, the method behaves like spatial blocked CV, see section 1.2.3. One drawback is that the algorithm may eliminate too many data points from the training set when there is strong clustering of the sites.

In summary, while SLOO provides a spatially informed approach to cross-validation, it is essential to consider its computational challenges, the increased variance of hold-out estimations, and the meticulous selection of the spatial bandwidth parameter to ensure its efficacy in model evaluation and prediction.

1.2.2 Spatial K-fold cross-validation

Spatial K-fold cross-validation (SKVC) is a variant of K-fold cross-validation introduced by Pohjankukka et al. (2017).

We follow the notations of Section 1.1.3. In Spatial K-fold cross-validation, the domain is divided into K folds, denoted by $\{\mathcal{S}_n^{\mathrm{v}1}, \ldots, \mathcal{S}_n^{\mathrm{v}K}\}$, where each fold $\mathcal{S}_n^{\mathrm{v}m}$ is defined such that its sites are selected randomly, and its cardinality satisfies $\mathrm{Card}(\mathcal{S}_n^{\mathrm{v}m}) = n_m$, with $\left\lfloor \frac{n}{K} \right\rfloor \leq n_m \leq \left\lceil \frac{n}{K} \right\rceil$. It is important to note that here we refer to the cardinality of the spatial domains $\mathcal{S}_n^{\mathrm{v}m}$ rather than the cardinality of data subsets, as it is typically

done in classical K-fold cross-validation. However, since each site corresponds to a data point, the cardinality of each validation domain $\mathcal{S}_n^{\mathrm{v}m}$ is equal to the cardinality of the corresponding validation data subset $\mathcal{D}_n^{\mathrm{v}m}$. This means that $\mathrm{Card}(\mathcal{S}_n^{\mathrm{v}m}) = \mathrm{Card}(\mathcal{D}_n^{\mathrm{v}m})$, ensuring consistency between the spatial partitioning and the data used in validation. Besides, the partition satisfies $\cap_m \mathcal{S}_n^{\mathrm{v}m} = \emptyset$ and $\cup_m \mathcal{S}_n^{\mathrm{v}m} = \mathcal{S}_n$.
During the $m-$th step of SKCV, given the validation domain $\mathcal{S}_n^{\mathrm{v}m}$, the set of dead points is defined as

$$\mathcal{S}_{n,h}^{\mathrm{D}m} = \{\mathbf{s}_j \notin \mathcal{S}_n^{\mathrm{v}m} : \exists\ \mathbf{s}_i \in \mathcal{S}_n^{\mathrm{v}m} : d(\mathbf{s}_j, \mathbf{s}_i) \leq h\}$$

and the training domain is

$$\mathcal{S}_{n,h}^{\mathrm{t}m} = \{\mathbf{s} \in \mathcal{S}_n : \mathbf{s} \notin \mathcal{S}_n^{\mathrm{v}m} \cup \mathcal{S}_{n,h}^{\mathrm{D}m}\}.$$

The SKCV predictive empirical risk is given by,

$$\mathcal{R}_{n,h}^{\mathrm{SKCV}} = \frac{1}{K} \sum_{m=1}^{K} \mathcal{R}_{n,h}^{\mathrm{v}m}. \tag{1.2.3}$$

Figure 1.4 illustrates an example of SKCV applied to a lattice $\mathcal{S}_n$ of size $n = 80 \times 60$, with $K = 200$ folds of cardinality 24, and a dead zone of radius $h = 8$ around each point of the validation domain.

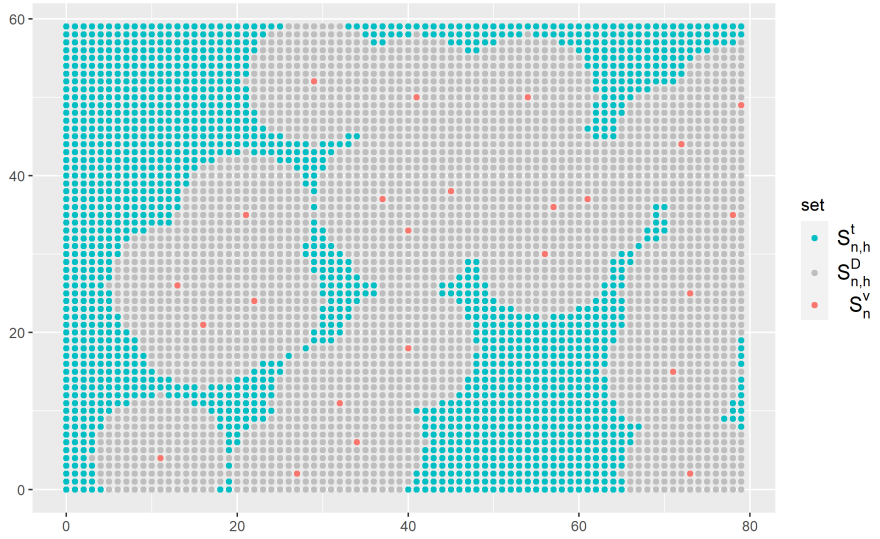


Figure 1.4: Dataset partition for one iteration of Spatial K-fold cross-validation with buffer radius $h = 8$ and $K = 200$.

Limitations

When employing spatial K-fold cross-validation, one potential drawback arises from the removal of a substantial number of training data points, which can introduce an additional pessimistic bias in prediction performance. To investigate this bias, Pohjankukka et al. (2017) conducted a study where they randomly removed an equivalent number of points as those excluded during SKCV and compared the outcomes. Their findings indicate that random removal of training data points has minimal impact on prediction performance compared to spatial-based data exclusion.

Furthermore, the number of folds used in the SKCV procedure can exacerbate the aforementioned challenges. Depending on the size and spatial distribution of the data, a significant portion of the training data may be removed due to the cumulated effect of dead zones. Therefore, choosing an optimal number of folds is crucial to balance capturing spatial dependencies and maintaining an adequate sample size for robust model training.

Finally, similar to SLOO, the effectiveness of SKCV depends on the selection of an appropriate buffer radius. Pohjankukka et al. (2017) noted that the estimated RMSE by SKCV increased as the size of the dead zone did so. However, no criteria for the choice of the optimal buffer was given. Henry Crosby and Jarvis (2020) proposed a heuristic for determining the optimal buffer radius. In line with common recommendations in the literature, this heuristic suggests using the range of the variogram of the response variable as an upper bound for the buffer radius in the general case. Specifically, it defines an interval of potential values for the buffer radius, ranging from zero to the variogram range $d_{\max}$, depending on the proximity between the prediction locations and the training points. The criteria for the buffer radius h are as follows,

$$h = \begin{cases} 0, & \text{if } \frac{\mu'_{tt}}{\mu'_{tr}} \leq 1, \quad \text{(interpolation scenario)} \\ \mu'_{tt}, & \text{if } \frac{\mu'_{tt}}{\mu'_{tr}} > 1 \text{ and } \mu'_{tt} < d_{\max}, \quad \text{(mixture scenario)} \\ d_{\max}, & \text{in any other case,} \quad \text{(extrapolation scenario),} \end{cases} \tag{1.2.4}$$

where μ'_{tt} is the average distance between each point in the training set and the corresponding prediction location, μ'_{tr} is the average distance within the training set and $d_{\max}$ is the range of the variogram of the response variable.

This method leverages information about the prediction locations (the new points where the model will be applied), making it adaptable to various spatial settings. In the pure interpolation scenario, where the prediction locations are well represented by the training set, the heuristic suggests using classical K-fold cross-validation with $h = 0$. For scenarios with both interpolation and extrapolation, the mixture scenario, it proposes a buffer radius equal to the average distance μ'_{tt}, provided that this distance is less than $d_{\max}$. In the

extrapolation scenario, where prediction points are relatively far from the training data, the buffer radius defaults to the variogram range $d_{\max}$.

It is important to note that while this heuristic provides a systematic approach for buffer selection, the paper does not present any evidence for the optimality of the proposed buffer values.

In conclusion, although SKCV is designed to account for spatial autocorrelation by introducing buffer zones, it often results in the removal of an excessive number of data points. This significant reduction in the effective sample size can adversely affect model performance, leading to decreased statistical power and less reliable predictions. Consequently, SKCV may not always be the most efficient method for spatial cross-validation due to its tendency to discard more data than necessary. It is crucial to carefully evaluate the selection of the buffer radius and the number of folds, or consider alternative cross-validation strategies, to balance the need for spatial independence with the retention of sufficient data for robust spatial analysis.

1.2.3 Blocked cross-validation

Blocked cross-validation (BCV) is a well-established spatial cross-validation technique that differs from SKCV by using a structured block pattern instead of random location selection when determining the folds. This method is particularly advantageous when the goal is to identify causal predictors or forecast within unique spatial frameworks, as each block can encapsulate distinct spatial features. According to Roberts et al. (2017), blocked cross-validation performs well in assessing a model's ability to extrapolate to independent data, making it a valuable tool in spatial analysis.

This method can be considered with and without spatial buffering around the block; in what follows we will refer to Spatial blocked cross-validation (SBCV) when considering buffering around the block (with buffer's width $h > 0$) and blocked cross-validation otherwise, with no buffer ($h = 0$).

Figure 1.4 illustrates an example of SBCV. Here, $\mathcal{S}_n$ 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 1.4 and 1.5 is the same, the cardinal of the dead zone $\mathcal{S}^{\mathrm{D}}_{n,h}$ is 324 for SBCV and 3074 for SKCV.

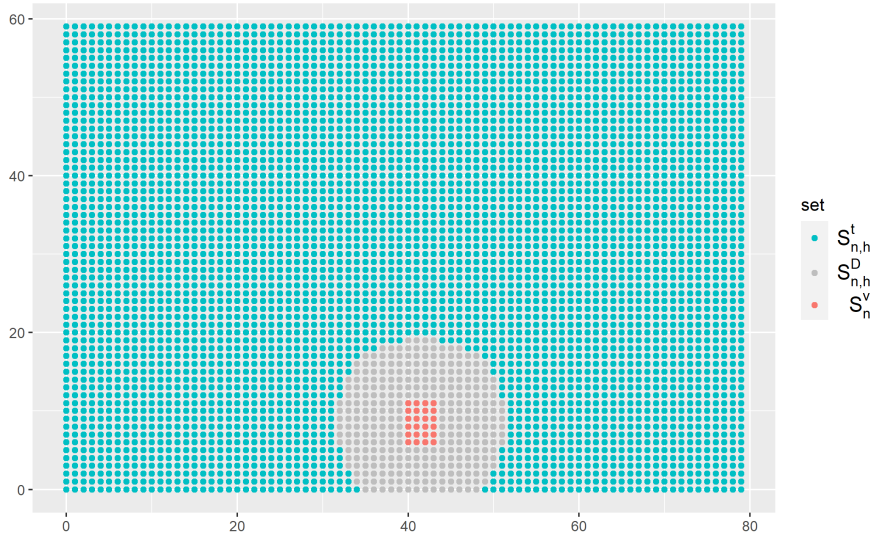


Figure 1.5: Dataset partition for one iteration of SBCV with buffer of radius $h = 8$ and square block of size 4×6.

Let B be the number of blocks, analogously to K in SKCV, but with blocks defined by neighboring locations rather than random selection. The risk estimate for SBCV is given by:

$$\mathcal{R}^{\text{SBCV}}_{n,h} = \frac{1}{B} \sum_{m=1}^{B} \mathcal{R}^{\text{v}_m}_{n,h}. \tag{1.2.5}$$

Here, each validation domain $\mathcal{S}^{V_m}_{n,h}$ is a block. It is important to note that unlike SBCV, the risk estimate $\mathcal{R}^{\text{BCV}}_n$ does not depend on h, because it does not utilize a dead zone:

$$\mathcal{R}^{\text{BCV}}_n = \frac{1}{B} \sum_{m=1}^{B} \mathcal{R}^{\text{v}_m}_n. \tag{1.2.6}$$

Comparing SBCV with SKCV, the structuring of blocks in SBCV ensures that fewer points fall within the dead zone compared to SKCV, as seen in figures 1.4 and 1.5. They illustrate how far the cardinality of the buffers differs, though the validation domain in both methods includes an identical number of locations (24 in this example). Compared to SKCV, SBCV minimizes the impact of the dead zone, enhancing the validation process's integrity. Overall, blocked cross-validation is the most widespread spatial cross-validation method.

The review paper by Jemeljanova et al. (2024) on machine learning methods in environmental studies found 20 different examples of the use of some variant of blocked cross-validation up to November 2023. Blocked cross-validation methods offer flexibility through various block definitions, such as oblong blocks or checkerboard structures, as discussed in Roberts et al. (2017). Variants include spatial blocking using clustering algorithms like

k-means, see Brenning (2012) and Urbazaev et al. (2018), or hierarchical clustering, see Stock et al. (2018) and Wang et al. (2023b).

The choice of block definition can significantly impact the performance of BCV, and different methods may be more appropriate depending on the characteristics of the data and the objectives of the study. The most common practice is to define the blocks by applying a clustering algorithm on the spatial coordinates as seen in the works by Brenning (2012), Urbazaev et al. (2018) and Stock et al. (2018). This approach can be particularly useful for irregularly spaced data, where regular grid-based blocks may not capture the spatial structure effectively.

However, for regularly spaced data, such as data collected on a lattice, clustering-based methods might result in irregularly shaped blocks, which may not be appropriate or may complicate the analysis. Figure 1.6 shows a partition defined by applying k-means clustering on the spatial coordinates in a 60×80 lattice. As can be seen, the resulting blocks are not regular, which may not align well with the underlying spatial structure of the data.

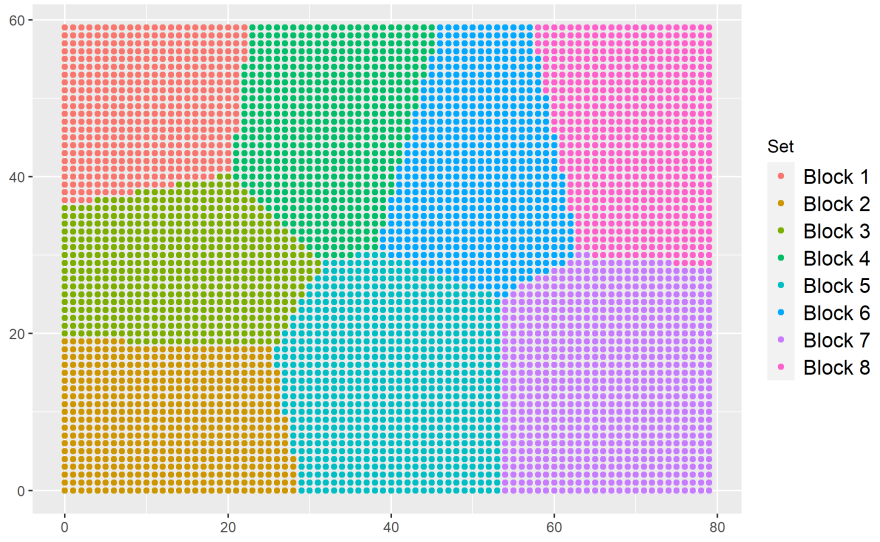


Figure 1.6: Dataset partition into blocks defined by k-means clustering on the spatial coordinates with $k = 8$.

Moreover, Wang et al. (2023b) argue that the data used for spatial prediction using machine learning models involve not only information about the spatial coordinates but also about the feature space, including the covariates and the response variable. Therefore, they suggest using a clustering approach that incorporates spatial coordinates, covariates, and the response variable to define the blocks. This method aims to create blocks that are more homogeneous in terms of both spatial location and feature space, potentially leading to better model validation.

The differences between these methods lie in how the blocks are defined and what aspects of the data are considered. Regular grid-based blocks may be appropriate for regularly spaced data and when the spatial structure aligns with the grid, offering simplicity and ease of implementation. Clustering-based methods may be more suitable for irregularly spaced data or when aiming to capture complex spatial dependencies. Including covariates and response variables in the clustering process, as suggested by Wang et al. (2023b), may further improve the effectiveness of BCV by accounting for feature space similarity as well as spatial proximity. However, it's important to note that these methods might not be appropriate for regularly spaced data if they result in irregularly shaped blocks, as this can lead to uneven representation of spatial areas and complicate the analysis. Moreover, there is also the question of the optimal number of clusters.

Mahoney et al. (2023b) explored three blocking techniques for spatial cross-validation: rectangular blocks, k-means clusters based on spatial locations, and the *Leave-one-disc-out* (LODO) method. LODO combines elements of SLOO and SBCV. The LODO method operates with as many iterations as there are points in the dataset, similarly to SLOO. However, instead of leaving out a single point for validation, LODO defines the validation set as a disc centered around each observation, including all points within a specified inclusion radius r. Additionally, it incorporates an exclusion radius h around the validation disc to ensure that nearby points are not included in the training set, thereby reducing spatial autocorrelation between training and validation sets.

Formally, for each observation $\mathbf{s}_m \in \mathcal{S}_n$, the validation set $\mathcal{S}_n^{\mathrm{v}_m}$ and training set $\mathcal{S}_n^{\mathrm{t}_m}$ are defined as,

$$
\begin{aligned}
\mathcal{S}_n^{\mathrm{v}_m} &= \{\mathbf{s}_i \in \mathcal{S}_n : \|\mathbf{s}_m - \mathbf{s}_i\| \leq r\}, \\
\mathcal{S}_n^{D_m} &= \{\mathbf{s}_i \in \mathcal{S}_n : r \geq \|\mathbf{s}_m - \mathbf{s}_i\| \leq h\}, \\
\mathcal{S}_n^{\mathrm{t}_m} &= \{\mathbf{s}_i \in \mathcal{S}_n : \|\mathbf{s}_m - \mathbf{s}_i\| > r + h\}.
\end{aligned}
$$

Leave-one-disc-out offers a flexible and localized approach to spatial cross-validation, combining features of both SLOO and SBCV. By adjusting the inclusion radius r and the exclusion radius (buffer size) h, practitioners can tailor the method to the specific spatial characteristics of their data. However, careful consideration of parameter selection and computational resources is necessary to effectively implement LODO. In particular, LODO needs to consider the selection of both h and r. In contrast, for SLOO, the primary consideration is determining the appropriate buffer size h. Moreover, it's important to note that LODO does not offer computational gains compared to SLOO.

Wang et al. (2023a) introduced a variation of SBCV that specifically addresses covariate shift during the data splitting process. Covariate shift refers to a situation where the

distribution of the covariates differs between the training data and the validation data, while the conditional distribution of the response variable given the covariates remains the same. Formally, let $p_{\text{train}}(\mathbf{X})$ and $p_{\text{val}}(\mathbf{X})$ denote the distributions of the covariates in the training and validation sets, respectively, and $p(\mathbf{Y} \mid \mathbf{X})$ denote the conditional distribution of the response variable given the covariates. Covariate shift occurs when,

$$p_{\text{train}}(\mathbf{X}) \neq p_{\text{val}}(\mathbf{X}), \quad \text{but} \quad p_{\text{train}}(\mathbf{Y} \mid \mathbf{X}) = p_{\text{val}}(\mathbf{Y} \mid \mathbf{X}).$$

This discrepancy means that while the relationship between $\mathbf{X}$ and $\mathbf{Y}$ remains consistent, the way $\mathbf{X}$ is distributed changes between the training and validation sets. Covariate shift can lead to biased estimates of model performance because the model is trained on a distribution of covariates that is different from the one it encounters during validation or testing. Here the covariates are random variables. This places their work within the random design framework, where both $\mathbf{X}$ and $\mathbf{Y}$ are considered random, and the variability in $\mathbf{X}$ is explicitly accounted for.

Limitations

Employing blocking techniques requires careful consideration of potential implications on data characteristics. Grouping similar units may inadvertently remove unique features critical for accurate interpolation, see for example Hoffimann et al. (2021).

An additional critical consideration when blocking involves determining the optimal block size for regular blocks. Roberts et al. (2017) recommend estimating the variogram of the dependent variable and using its range as the block size. This approach aims to capture the spatial autocorrelation present in the response variable, ensuring that observations within each block are sufficiently correlated.

Conversely, the `blockCV` package developed by Valavi et al. (2019), which offers a range of functions for generating training and testing folds for spatial cross-validation, but omits SBCV, suggests selecting the block size based on the median range obtained from the variograms of the covariates. This method focuses on the spatial structure of the predictors, under the assumption that accounting for the spatial autocorrelation in the covariates will lead to more reliable model validation.

Both recommendations apply to regular blocks in SBCV but differ in their emphasis on the dependent variable versus the covariates. The choice between these approaches may depend on the specific characteristics of the dataset and the objectives of the study. For instance, if the response variable exhibits strong spatial autocorrelation, using its variogram range may be more appropriate. On the other hand, if the covariates have significant spatial structure that influences the response, basing the block size on the covariates' variogram

ranges might provide better cross-validation results.

It is important to note that selecting an appropriate block size is crucial because it affects the balance between bias and variance in the cross-validation estimates. Blocks that are too small may not effectively mitigate spatial autocorrelation between training and validation sets, leading to overly optimistic performance estimates. Conversely, blocks that are too large may result in high variance and less reliable estimates due to insufficient data in each fold.

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, see Mahoney et al. (2023b). Choosing the most appropriate blocking method depends on the specific characteristics of the dataset and the goals of the study. Researchers should consider the spatial distribution of their data, the presence of spatial autocorrelation, and whether the inclusion of covariates and response variables in the blocking process would enhance model validation. Careful consideration of these factors will help ensure that the selected cross-validation method provides reliable and unbiased estimates of model performance.

1.2.4 Comparison of cross-validation strategies

Several studies have compared spatial cross-validation methods. In their seminal 2017 paper, Roberts et al. (2017) compared SBCV with SLOO and traditional cross-validation approaches within the framework of random forests. Their field of investigation is ecology, mainly species distribution Evans et al. (2011) and random forest are widely used in this field. More recently, Mahoney et al. (2023b) conducted a study using the same data generation process as Roberts et al. (2017), and compared different variants of SLOO and SBCV with classical cross-validation methods. Both studies benchmarked against an "ideal" root mean squared error (RMSE), with Roberts et al. employing visual comparisons of risk distributions, while Mahoney et al. focused on CV estimates falling within a specific percentile range of the ideal RMSE. Both concluded that classical cross-validation underestimates prediction errors in spatial contexts.

Ideal empirical risk

As in Roberts et al. (2017) and Mahoney et al. (2023b), we consider the **ideal empirical risk**. The motivation behind introducing the ideal risk comes from the need to have a

benchmark against which to compare the cross-validation risk estimates.

The predictive risk of an estimator $\hat{\boldsymbol{\beta}}$ is defined as the expected loss when predicting new observations at unseen locations, as given in Equation (1.1.4). To approximate the expected predictive performance, Roberts et al. (2017) propose averaging the prediction errors over multiple independent realizations of the spatial process.

Given M independent realizations $\mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \ldots, \mathbf{Y}^{(M)}$ of the spatial process $\mathbf{Y}$, we define the empirical ideal predictive risk for the l-th realization as,

$$\mathcal{R}_n^{I(l)} = \frac{1}{M-1} \sum_{\substack{j=1 \\ j \neq l}}^{M} \frac{1}{n} \sum_{\mathbf{s} \in \mathcal{S}_n} \left(\hat{Y}^{(l)}(\mathbf{s}) - Y^{(j)}(\mathbf{s}) \right)^2, \tag{1.2.7}$$

where,

- $\hat{Y}^{(l)}(\mathbf{s}) = \mathbf{X}(\mathbf{s})\hat{\boldsymbol{\beta}}^{(l)}$ is the predicted value at location $\mathbf{s}$ using the estimator $\hat{\boldsymbol{\beta}}^{(l)}$ obtained from the l-th realization,
- $Y^{(j)}(\mathbf{s})$ is the observed value at location $\mathbf{s}$ from the j-th realization.

This means that for each realization $Y^{(l)}(\mathbf{s})$ of the spatial process $\mathbf{Y}$, we obtain $\hat{\boldsymbol{\beta}}^{(l)}$ and then compute the ideal risk associated with this realization $\mathcal{R}_n^{I(l)}$ by considering the difference between the predicted values $\hat{Y}^{(l)}(\mathbf{s})$ and the observed values in the remaining $M-1$ independent realizations $Y^{(j)}(\mathbf{s})$ for all $j \neq l$.

This ideal risk measures the average prediction error of the model trained on $\mathbf{Y}^{(l)}$ when predicting independent realizations $\mathbf{Y}^{(j)}$ at the same locations. By averaging over multiple independent realizations, we obtain a more reliable estimate of the true predictive risk. The use of multiple independent realizations is a Monte Carlo approach, where we approximate expectations by averaging over a large number of random samples Rubinstein (1981).

The comparison with the ideal risk allows us to evaluate the effectiveness of different cross-validation strategies in estimating the true predictive performance of the model. It helps in identifying methods that provide less biased and reliable risk estimates in the presence of spatial dependence.

Simulation design

To evaluate and compare classical cross-validation and spatial cross-validation strategies, we now present a simulation study.

Let $\mathcal{S}_n$ be a rectangular lattice of size 80×60, with $n = 4800$ sites. We consider the regression model,

$$Y(\mathbf{s}) = \mathbf{X}(\mathbf{s})\boldsymbol{\beta} + \varepsilon(\mathbf{s}), \tag{1.2.8}$$

where $\boldsymbol{\varepsilon}$ is generated according to a zero-mean Gaussian spatial process with an isotropic exponential covariance function defined by,

$$C(h) = \sigma^2 \exp\left(-\frac{h}{\theta}\right).$$

We choose this covariance function because it is a well-known model for spatial dependence. We set $\sigma^2 = 1$ and consider different values of $\theta \in \{3, 6, 9, 12, 15, 20\}$ to control the strength of spatial dependence.

We introduce a non-stationary trend linked to the spatial location within the domain. We denote the spatial location as $\mathbf{s} = (s_1, s_2) \in \mathcal{S}_n$, where s_1 and s_2 represent the coordinates in the horizontal and vertical directions, respectively. We define the design vector at location $\mathbf{s}$ as,

$$\mathbf{X}(\mathbf{s})^\top = [1,\, X_1(\mathbf{s}),\, X_2(\mathbf{s})],$$

where $X_1(\mathbf{s}) = s_1 - 40$ and $X_2(\mathbf{s}) = s_2 - 30$ are centered spatial coordinates serving as covariates, and the first element corresponds to the intercept.

We set the parameter vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)^\top$. Following the proceedure described by Aldworth and Cressie (1999), β_1 and β_2 are chosen in such a way that the deterministic trend exhibits a variance comparable to the error term, achieving a signal-to-noise ratio of $\tau = \frac{\sigma^2_{\text{trend}}}{\sigma^2_Y} = \frac{1}{2}$, where σ^2_{trend} is the variance of the trend component and σ^2_Y is the total variance of $Y(\mathbf{s})$.

To ensure a balanced contribution from $\mathbf{X}_1$ and $\mathbf{X}_2$ to the variance of the trend component, we aim for their scaled variances to be equal

$$\beta_1^2 \operatorname{Var}(\mathbf{X}_1) = \beta_2^2 \operatorname{Var}(\mathbf{X}_2).$$

Here, $\mathbf{X}_1$ and $\mathbf{X}_2$ are deterministic covariates defined over the spatial domain $\mathcal{S}_n$. The terms $\operatorname{Var}(\mathbf{X}_1)$ and $\operatorname{Var}(\mathbf{X}_2)$ represent the sample variances of $\mathbf{X}_1$ and $\mathbf{X}_2$ across all spatial locations, calculated as,

$$\operatorname{Var}(\mathbf{X}_i) = \frac{1}{n}\sum_{k=1}^{n}\left(X_i(\mathbf{s}_k) - \overline{X}_i\right)^2, \quad \text{for } i = 1, 2,$$

where $\overline{X}_i = \frac{1}{n}\sum_{k=1}^{n} X_i(\mathbf{s}_k)$ is the mean of $\mathbf{X}_i$ over the spatial locations $\mathbf{s}_1, \ldots, \mathbf{s}_n$.

Given that the variance of the trend component is,

$$\sigma^2_{\text{trend}} = \beta_1^2 \operatorname{Var}(\mathbf{X}_1) + \beta_2^2 \operatorname{Var}(\mathbf{X}_2),$$

the total variance of $Y(\mathbf{s})$ is the sum of the variance of the trend and the variance of the error term $\varepsilon(\mathbf{s})$,

$$\sigma^2_Y = \sigma^2_{\text{trend}} + \sigma^2,$$

where $\sigma^2 = \operatorname{Var}(\varepsilon(\mathbf{s}))$ is the variance of the stochastic error term.

This procedure leads to set $\beta_1 = \dfrac{1}{32}$ and $\beta_2 = \dfrac{1}{24}$.

Finally, the intercept β_0 is chosen to ensure that the coefficients of variation of the estimators of β_0, β_1, and β_2 are similar, balancing the estimation precision across parameters. So we set the parameter vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)^\top = \left(0.9, \dfrac{1}{32}, \dfrac{1}{24}\right)$.

Note that in our model, although the expectation $\mathbb{E}[Y(\mathbf{s})]$ varies spatially due to the trend, the covariance between $Y(\mathbf{s}_i)$ and $Y(\mathbf{s}_j)$ depends only on the distance $h = \|\mathbf{s}_i - \mathbf{s}_j\|$, because the residuals $\varepsilon(\mathbf{s})$ are generated with an isotropic covariance function.

Figure 1.7 shows a realization of $\mathbf{Y}$ with $\theta = 9$, illustrating the spatial variation captured by the model.

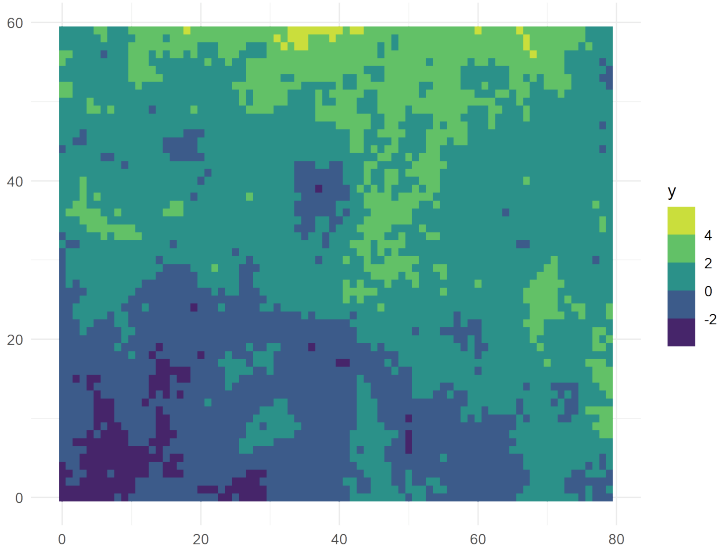


Figure 1.7: One simulated data generated following model of equation (1.2.8) with $\theta = 9$.

Our estimation scenario ignores the spatial dependence structure; we estimate the empirical risk associated with the ordinary least squares estimator using different cross-validation methods. Although Generalized Least Squares can account for spatial depen-

dence by incorporating the covariance structure of the errors, it requires accurate specification of the spatial correlation model, which can be complex and computationally intensive. In many practical situations, practitioners may prefer ordinary least squares due to its simplicity and ease of interpretation, even when spatial dependence is present.

Moreover, since our primary interest is to estimate the risk of an estimator, the estimator itself does not necessarily need to be the best possible one. Studying spatial cross-validation methods with ordinary least squares allows us to focus on the effectiveness of these validation techniques in estimating the predictive risk, regardless of whether the estimator fully accounts for spatial dependence. This approach provides valuable insights into how different cross-validation methods perform in practice, especially when the model does not explicitly model spatial autocorrelation.

Comparison of dead zone and training domain sizes for SBCV and SKCV

Let's first compare the SBCV and SKCV methods with respect to the sizes of the dead zone $\mathcal{S}_{n,h}^{\mathrm{D}}$ and the training domain $\mathcal{S}_{n,h}^{\mathrm{t}}$, while keeping the size of the validation domain fixed at $\mathrm{Card}(\mathcal{S}_n^{\mathrm{v}}) = 24$.

Table 1.1 presents the average number of points in the dead zone and the training domain for different buffer sizes h, along with the associated percentages relative to the total number of points $n = 4800$. The averages are calculated over all folds or blocks in the cross-validation procedure.

For example, for SKCV, the average size of the dead zone is computed as,

$$\mathrm{ave}(\mathrm{Card}(\mathcal{S}_n^{\mathrm{D}})) = \frac{1}{K}\sum_{m=1}^{K} \mathrm{Card}(\mathcal{S}_{n,h}^{\mathrm{D}_m}),$$

and the average size of the training domain is,

$$\mathrm{ave}(\mathrm{Card}(\mathcal{S}_n^{\mathrm{t}})) = \frac{1}{K}\sum_{m=1}^{K} \mathrm{Card}(\mathcal{S}_{n,h}^{\mathrm{t}_m}).$$

By averaging over all folds for SKCV and blocks for SBCV, we obtain a representative measure of the typical sizes of the dead zone and training domain across the entire cross-validation process. This allows for a fair comparison between SBCV and SKCV methods regarding how the buffer size h affects the amount of data excluded from the training set (dead zone) and the remaining data used for training.

The associated percentages are calculated by dividing the average sizes by the total number of points $n = 4800$ and multiplying by 100.

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 $\mathcal{S}_n^{\mathrm{v}}$ 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.

Method	h	ave(Card($\mathcal{S}_n^{\mathrm{D}}$)) (%)	ave(Card($\mathcal{S}_n^{\mathrm{t}}$)) (%)
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%)

Table 1.1: Comparison of average dead zone (ave(Card($\mathcal{S}_n^{\mathrm{D}}$))) and training domain (ave(Card($\mathcal{S}_n^{\mathrm{t}}$))) sizes in number and percentages for SBCV and SKCV, with Card($\mathcal{S}_n^{\mathrm{v}}$) = 24.

Comparison of the empirical risks with the ideal risk

We now turn to examine the different cross-validation methods and how they behave with respect to one another; their performance is also compared to the ideal risk defined in (2.1.1), which we aim to reach.

We simulate $M = 100$ copies of Y under the design of equation (1.2.8). Let us recall that we consider $\theta \in \{3, 6, 9, 12, 15, 20\}$ to depict various schemes of dependence.

We consider the following cross-validation methods:

- Ordinary leave-one-out.
- Ordinary K-fold cross-validation, using $K = 200$ folds.
- Spatial leave-one-out, considering different buffer sizes with $h \in \{5, 10, 15, 20\}$.
- Spatial K-fold cross-validation, keeping $K = 200$ and $h \in \{5, 10, 15, 20\}$.

- Blocked cross-validation (without spatial buffering), using $B = 200$ blocks of size 6×4, aligning the number of folds and the size of the validation set in each iteration with those used in KCV.
- Spatial blocked cross-validation, adding spatial buffering around each block with $h \in \{5, 10, 15, 20\}$.

For each θ, we compute the ideal risk $\mathcal{R}_n^{I(l)}$ and the risk estimates $\mathcal{R}_{n,h}^{\text{method}(l)}$ obtained from the methods described above for $l = 1$ to $M = 100$, considering different h values. $h = 0$ corresponds to the methods without spatial buffering.

Figure 1.8 displays the ideal risk and cross-validation risk estimates derived from 100 replications of the response variable Y across different cross-validation methods. For $h = 0$ SLOO, SKCV and SBCV become LOO, KCV and BCV.

We see that 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 suggests a potential underestimation of the true risk in non-spatial cross-validation methods. This reinforces the idea that it is necessary to propose spatial adaptations of classic cross-validation methods.

For small values of h, the methods SBCV, SKCV and SLOO perform similarly, indicating that when the buffer size is minimal, the distinction between these spatially aware methods is less pronounced. As the buffer size h increases, differences among the methods become more apparent, reflecting how each method handles the exclusion of nearby spatial points differently.

As expected, the necessity for spatially aware cross-validation methods becomes more critical as the spatial dependence parameter θ increases. Larger values of θ correspond to stronger spatial autocorrelation, which means that observations are correlated over longer distances. Consequently, a larger buffer size h is required to adequately reduce the influence of spatial autocorrelation between training and validation sets.

However, at higher values of θ, we observe that none of the cross-validation methods manage to reach the ideal risk, underscoring a collective limitation in accurately estimating the predictive risk in the presence of strong spatial dependence. This suggests that as spatial autocorrelation intensifies, it becomes increasingly challenging for cross-validation methods to provide unbiased risk estimates.

To illustrate this, let us consider the case $\theta = 12$. The average value of the ideal risk $\mathcal{R}_n^{I(l)}$ is 1.2092. None of the spatial cross-validation methods achieve this value; the closest is SBCV with $h = 20$, which yields an average risk estimate of 1.1228, followed by SLOO with $h = 20$ at 1.0642.

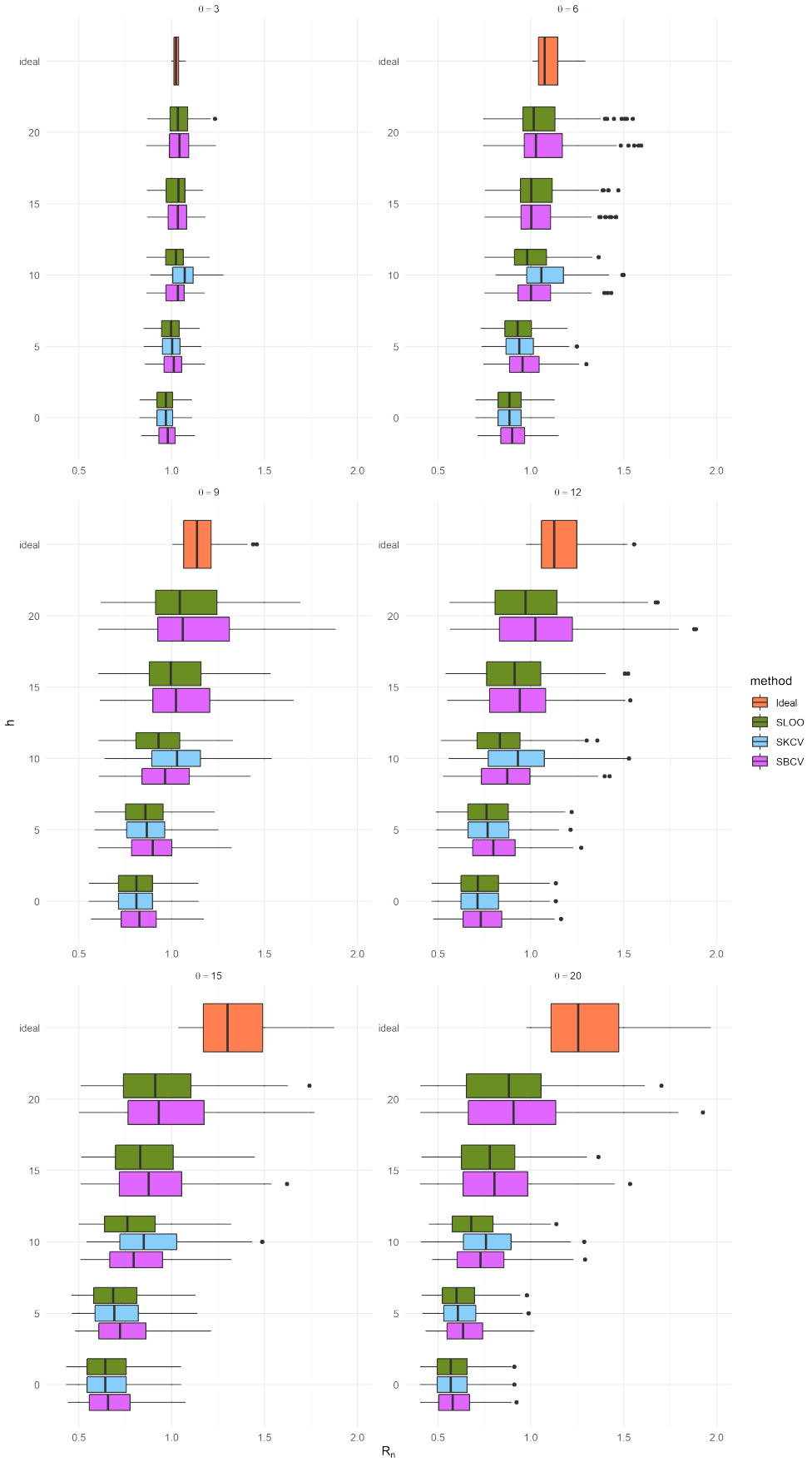


Figure 1.8: Ideal risk and cross-validation risk estimates derived from 100 replications of the response variable Y across different cross-validation methods. For $h = 0$, SLOO, SKCV and SBCV become LOO, KCV and BCV, respectively.

While increasing the buffer size brings the risk estimates closer to the ideal risk, this improvement comes at the cost of increased variance in the risk estimates. Specifically, SBCV with $h = 20$ has a sample variance of 0.1753, and SLOO has a variance of 0.1282, whereas the sample variance of the ideal risk is much lower with a value of 0.0526. This indicates that larger buffer sizes introduce greater uncertainty into the risk estimates.

We could not consider larger buffer sizes because they would result in a substantial loss of data due to the expansion of the dead zone, leaving too few points for effective model training. Notably, we encountered computational limitations with SKCV at higher values of h, specifically $h = \{15, 20\}$, where the training set became empty after excluding the validation and dead zone areas. This highlights a practical constraint in applying spatial cross-validation methods with large buffer sizes, especially in datasets with limited sample sizes.

As θ increases, we see a corresponding rise in both the 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. Similarly, the risks estimated by both spatial and non-spatial cross-validation methods increase in variance with θ. Upon examining the spatial cross-validation methods more closely, we observe 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, we find that the results are generally similar. However, SLOO typically provides marginally lower risk estimates than SBCV for the same buffer size h, and exhibits slightly reduced variability. This suggests that both methods respond similarly to the spatial structure of the data for a given buffer size, but SLOO may offer a slight advantage in terms of both accuracy and precision.

These findings are supported by Figures 1.9 and 1.10, which illustrates the accuracy of risk estimates for SLOO and SBCV relative to the ideal risk. The accuracy is computed as,

$$\delta^2(h) = \left(\frac{1}{M} \sum_{l=1}^{M} \left(\mathcal{R}_{n,h}^{(l)} - \mathcal{R}_n^{I(l)} \right) \right)^2, \tag{1.2.9}$$

The variance of these estimates is also analyzed, calculated by,

$$\sigma^2_{\mathcal{R}_{n,h}} = \frac{1}{M} \sum_{l=1}^{M} \left(\mathcal{R}_{n,h}^{(l)} - \overline{\mathcal{R}_{n,h}} \right)^2, \tag{1.2.10}$$

where $\overline{\mathcal{R}_{n,h}} = \frac{1}{M} \sum_{l=1}^{M} \mathcal{R}_{n,h}^{(l)}$ is the average estimated risk over all simulation runs.

These two measures are presented in the spirit of a bias-variance decomposition, although $\delta^2(h)$ is not formally the squared bias. Specifically, $\delta^2(h)$ represents the squared difference between the average estimated risk and the average ideal risk, providing an indication of the accuracy (bias) of the cross-validation method. The variance $\sigma^2_{\mathcal{R}_{n,h}}$ reflects the variability of the risk estimates across different simulations. Figures 1.9 and 1.10 present the curves of $\delta^2(h)$, $\sigma^2_{\mathcal{R}_{n,h}}$, and their sum $\delta^2(h) + \sigma^2_{\mathcal{R}_{n,h}}$, as functions of the buffer size h, for different values of the spatial dependence parameter θ, and for the risk estimates obtained using SLOO and SBCV methods.

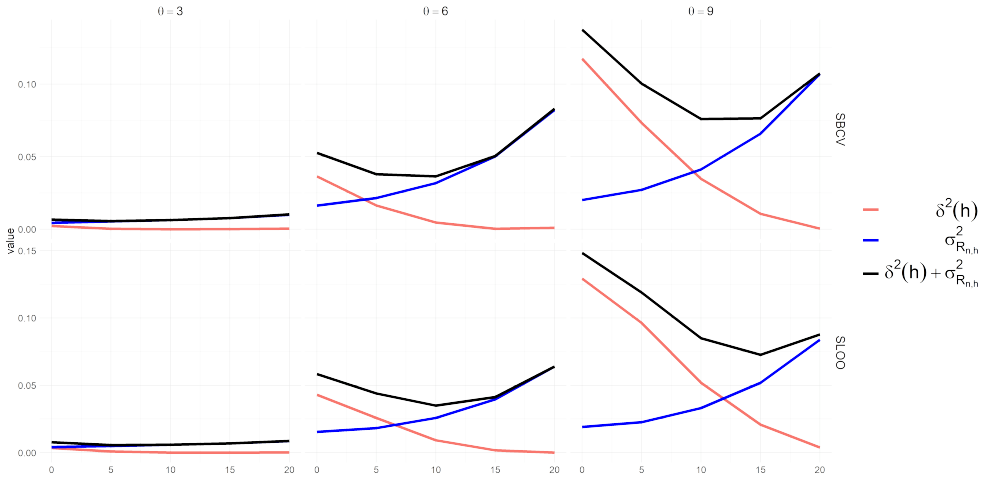


Figure 1.9: Comparative accuracy $\delta^2(h)$, variance $\sigma^2_{\mathcal{R}_{n,h}}$ of risk estimates, and their sum for SLOO and SBCV and increasing values of h for $\theta = (3, 6, 9)$.

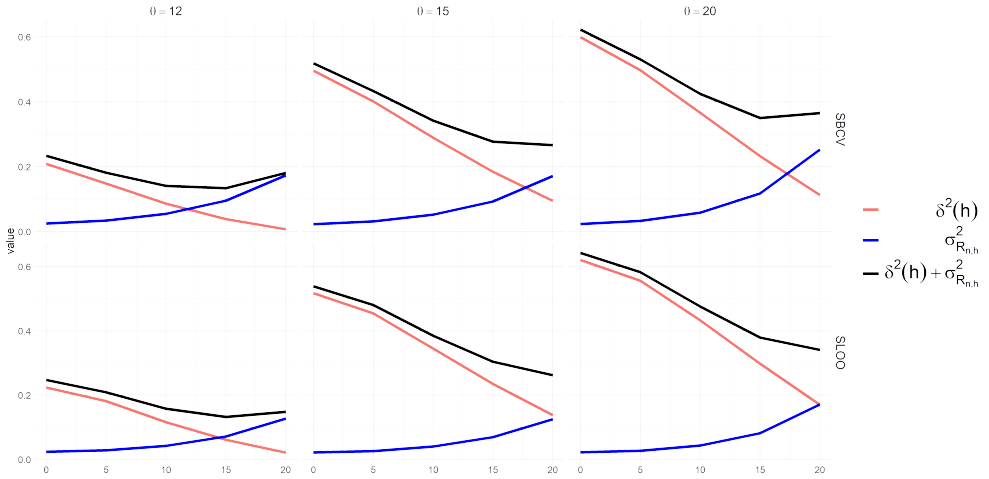


Figure 1.10: Comparative accuracy $\delta^2(h)$, variance $\sigma^2_{\mathcal{R}_{n,h}}$ of risk estimates and their sum for SLOO and SBCV and increasing values of h for $\theta = (12, 15, 20)$.

Starting with $\theta = 3$ on the left of Figure 1.9, representing low spatial dependence,

we observe that increasing the buffer size h has little influence on both $\delta^2(h)$ and $\sigma^2_{\mathcal{R}_{n,h}}$. The accuracy measure $\delta^2(h)$ remains nearly constant and close to zero, indicating that the cross-validation estimates are already close to the ideal risk even without a buffer. This suggests that when spatial dependence is weak, the necessity for spatial buffering is minimal, and traditional cross-validation methods may suffice.

As the strength of spatial dependence increases (larger θ), we notice that the initial value of $\delta^2(0)$ is farther from zero, indicating a larger discrepancy between the estimated risk and the ideal risk when no buffer is used. This highlights the bias introduced by spatial autocorrelation when training and validation sets are not sufficiently separated. As the buffer size h increases, $\delta^2(h)$ decreases, approaching zero. This improvement in accuracy underscores the importance of increasing the distance between the training and validation sets to mitigate the influence of spatial autocorrelation.

However, for the two largest values of θ, representing strong spatial dependence, we observe that $\delta^2(h)$ does not reach zero within the range of buffer sizes we considered. This suggests that even with larger buffers, the cross-validation estimates cannot fully eliminate the bias introduced by strong spatial dependence, possibly due to practical limitations such as the finite size of the dataset or the maximum feasible buffer size before the training set becomes too small.

Simultaneously, we observe that as h increases, the variance of the risk estimates $\sigma^2_{\mathcal{R}_{n,h}}$ also increases. This is attributable to the reduction in the effective size of the training set as more data points are excluded by the buffer. A smaller training set leads to less stable model estimates and, consequently, higher variability in the risk estimates across simulations. The effect of increased variance is more pronounced with stronger spatial dependence (larger θ). Additionally, SBCV generally exhibits higher variance and lower accuracy than SLOO across different buffer sizes. This indicates that SLOO may be more reliable in terms of both precision (lower variance) and accuracy (lower bias) in estimating the predictive risk.

An important observation from Figures 1.9 and 1.10 is the presence of an apparent optimal buffer size that minimizes the total error, represented by the sum $\delta^2(h) + \sigma^2_{\mathcal{R}_{n,h}}$. Examining the curves for $\theta = 12$ in the left panels of Figure 1.10 , we observe that the accuracy improves for both SLOO and SBCV as the buffer size h increases, indicated by a decrease in $\delta^2(h)$. However, this improvement in accuracy comes at the cost of increased variance $\sigma^2_{\mathcal{R}_{n,h}}$. Initially, increasing h reduces the bias (improves accuracy) more than it increases the variance, leading to a net decrease in total error. Beyond a certain point, further increases in h lead to a smaller reduction in bias but a more substantial increase in variance, resulting in a net increase in total error. This demonstrates the classical bias-variance trade-off, where an optimal balance between bias and variance yields the most

reliable risk estimates. Increasing h reduces bias due to mitigating spatial autocorrelation but increases variance due to a reduced effective sample size.

Overall, these results emphasize the importance of carefully selecting the buffer size h in spatial cross-validation methods. A larger buffer size effectively reduces the bias introduced by spatial autocorrelation between training and validation sets but can lead to higher variance in risk estimates due to fewer data points available for model training. Balancing this trade-off is crucial for obtaining reliable risk estimates.

Comparison of block sizes in SBCV

In examining SBCV, we explore the comparison of different block sizes. These blocks are crucial as they affect both the training and validation phases of the cross-validation process. Larger blocks encompass more spatial context, potentially capturing more spatial variability but at the cost of reduced granularity in local spatial patterns. Conversely, smaller blocks provide finer spatial resolution but may not adequately represent larger-scale spatial dependencies.

In practical terms, the optimal block size balances these considerations, aiming to capture significant spatial variability while maintaining sufficient sample size within each block for robust estimation. 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 1.11 illustrates the impact of different block sizes on risk estimates in SBCV, categorized by varying levels of spatial dependence. A consistent trend is observed where risk estimates increase with the enlargement of block size. Particularly 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. The necessity of using spatial buffering remains evident, especially as the strength of spatial dependence intensifies, even for the largest block size considered.

Conclusion

In summary, SKCV should not be considered due to the significant loss of observations it entails, which can adversely affect the reliability of risk estimates. In contrast, SBCV and SLOO emerge as superior methods for risk estimation, producing results that align more closely with the ideal risk. However, a crucial aspect in the application of these methods is the determination of the buffer size h, which directly influences the balance between bias and variance in the risk estimates.

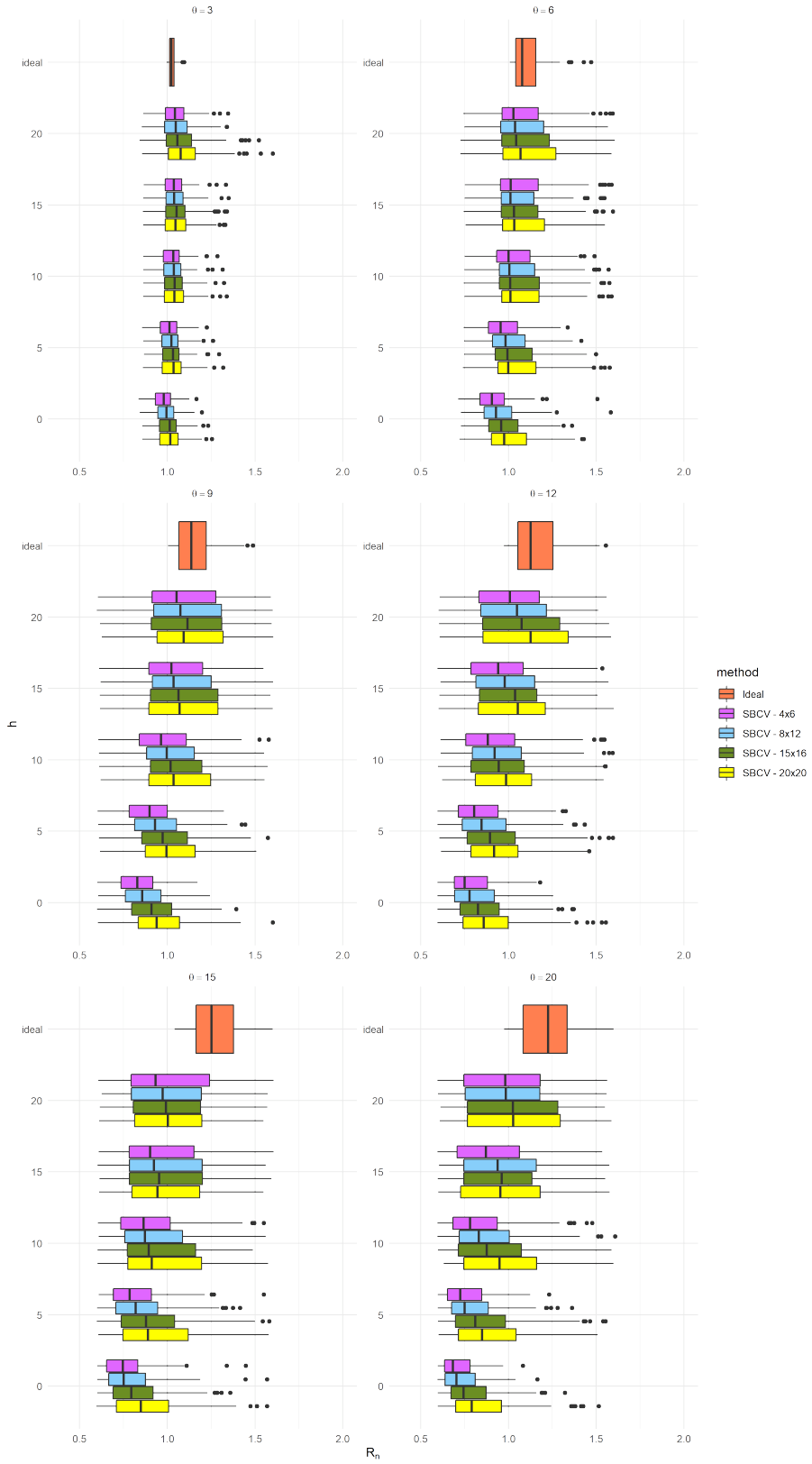


Figure 1.11: Risk estimates $\mathcal{R}^{\text{SBCV}}_{n,h}$ based on 100 replications of the response variable Y, compared to the ideal risk across various block sizes.

When choosing between SBCV and SLOO, computational efficiency becomes a significant factor. SBCV offers a substantial reduction in computation time compared to SLOO, requiring only 200 iterations versus SLOO's 4800, thereby significantly reducing computational demands. However, SBCV introduces additional complexity in selecting the appropriate block shape and size, which requires careful consideration to ensure optimal performance.

Given these considerations, determining the optimal buffer size h becomes imperative for both SLOO and SBCV methods. The buffer size not only affects the degree of independence between training and validation sets but also impacts the effective sample size used for model estimation. An inappropriate choice of h can lead to biased risk estimates due to residual spatial autocorrelation or increased variance resulting from insufficient training data. The challenge lies in balancing the trade-off between reducing bias (by increasing h to minimize spatial autocorrelation between training and validation sets) and controlling variance (by keeping h small enough to retain an adequate number of training observations).

In the following chapter, we will address this critical issue by proposing methods for determining the optimal buffer size h for both SLOO and SBCV. We will explore empirical approaches, aiming to enhance the validity of risk estimates obtained from spatial cross-validation.

Chapter 2

Determination of optimal buffer size

2.1 Ideal risk and spatial cross-validation risk discrepancy

The selection of h in cross-validation with spatial buffering is crucial. It is well understood that h must be large enough to ensure independence between the training and validation sets. However, if h is too large, it may result in the elimination of too many observations, leading to poor estimates. In this chapter, we explore the selection of the buffer size h in spatial cross-validation, focusing on Spatial leave-one-out and Spatial blocked cross-validation. First, let us show that an optimal distance between the validation set and the training set exists. To do this we compute the ideal risk and we define the optimal h as the value that minimizes the difference between this ideal risk and the risk estimated using spatial cross-validation.

We recall the definitions of the *empirical ideal* risk and the empirical risks we are working with. Let us assume that we have M independent realizations $\mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \ldots, \mathbf{Y}^{(M)}$ of the spatial process $\mathbf{Y}$. Let $\mathbf{Y}^{(l)}$ denote the l-th realization.

The **empirical ideal risk** for the l-th realization is defined as,

$$\mathcal{R}_n^{I(l)} = \frac{1}{M-1} \sum_{\substack{j=1 \\ j \neq l}}^{M} \frac{1}{n} \sum_{\mathbf{s} \in \mathcal{S}_n} \left(\hat{Y}^{(l)}(\mathbf{s}) - Y^{(j)}(\mathbf{s}) \right)^2, \tag{2.1.1}$$

where $\hat{Y}^{(l)}(\mathbf{s}) = \mathbf{X}(\mathbf{s})\hat{\boldsymbol{\beta}}^{(l)}$, and $\hat{\boldsymbol{\beta}}^{(l)}$ is the estimator computed from the l-th realization,

$$\hat{\boldsymbol{\beta}}^{(l)} = (\mathbf{X}\mathbf{X})^{-1}\mathbf{X}\mathbf{Y}^{(l)}.$$

The **SLOO risk** for the l-th realization is defined as,

$$\mathcal{R}_{n,h}^{\mathrm{SLOO}(l)} = \frac{1}{n}\sum_{m=1}^{n}\left(Y^{(l)}(\mathbf{s}_m) - \mathbf{X}(\mathbf{s}_m)\hat{\boldsymbol{\beta}}_h^{\mathrm{t}_m(l)}\right)^2, \tag{2.1.2}$$

where $\hat{\boldsymbol{\beta}}_h^{\mathrm{t}_m(l)}$ is the estimator computed on the training set $\mathcal{D}_{n,h}^{\mathrm{t}_m(l)}$, defined as,

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

with $\mathbf{X}_{m,h}$ and $\mathbf{Y}_{m,h}^{(l)}$ representing the design matrix and response vector excluding observations from the validation set $\mathcal{S}_n^{\mathrm{v}_m}$ and the buffer zone $\mathcal{S}_{n,h}^{\mathrm{D}_m}$.

Similarly, the **SBCV empirical risk** for the l-th realization is defined as,

$$\mathcal{R}_{n,h}^{\mathrm{SBCV}(l)} = \frac{1}{B}\sum_{m=1}^{B}\mathcal{R}_{n,h}^{\mathrm{v}_m(l)}, \tag{2.1.3}$$

where B is the number of spatial blocks. Now the subscript h highlights the buffer and l the l-th realization.

These risks are computed for each realization using buffer sizes $h \in \mathcal{H}$, a set of possible values of h.

For simplicity, we denote the empirical risk obtained for the l-th realization as $\mathcal{R}_{n,h}^{(l)}$, which can refer to either the SLOO or SBCV method.

2.2 "Ideal" buffer size

For each l we want to compare the empirical ideal risk $\mathcal{R}_n^{I(l)}$ to the risk $\mathcal{R}_{n,h}^{(l)}$ obtained by a spatial cross-validation method. To assess the discrepancy between them, we consider the mean squared differences across all realizations l,

$$\hat{\mathcal{L}}(h) = \frac{1}{M}\sum_{l=1}^{M}\left(\mathcal{R}_{n,h}^{(l)} - \mathcal{R}_n^{I(l)}\right)^2. \tag{2.2.1}$$

Our objective is to determine as the best buffer size the optimal h denoted by h_I, which minimizes $\hat{\mathcal{L}}(h)$,

$$h_I = \underset{h\in\mathcal{H}}{\operatorname{argmin}}\,\hat{\mathcal{L}}(h). \tag{2.2.2}$$

Adopting this procedure ensures that we choose h, based on the observed data, in such a way that the spatial cross-validation risk is close to the (empirical) ideal risk. We start

our study by analyzing the behavior of $\hat{\mathcal{L}}(h)$. We decompose it into several components. We can rewrite the difference $\mathcal{R}_n^{I(l)} - \mathcal{R}_{n,h}^{(l)}$ by adding and subtracting the averaged values

$$\overline{\mathcal{R}_n^I} = \frac{1}{M}\sum_{l=1}^{M} \mathcal{R}_n^{I(l)} \tag{2.2.3}$$

and

$$\overline{\mathcal{R}_{n,h}} = \frac{1}{M}\sum_{l=1}^{M} \mathcal{R}_{n,h}^{(l)}. \tag{2.2.4}$$

We obtain

$$\mathcal{R}_n^{I(l)} - \mathcal{R}_{n,h}^{(l)} = \left(\mathcal{R}_n^{I(l)} - \overline{\mathcal{R}_n^I}\right) - \left(\mathcal{R}_{n,h}^{(l)} - \overline{\mathcal{R}_{n,h}}\right) + \left(\overline{\mathcal{R}_n^I} - \overline{\mathcal{R}_{n,h}}\right). \tag{2.2.5}$$

By substituting this expression into equation ((2.2.1)) and expanding the square, we obtain,

$$\hat{\mathcal{L}}(h) = T_1 + T_2 + T_3 + T_4 + T_5, \tag{2.2.6}$$

where the five terms are defined as:

$$T_1 = \frac{1}{M}\sum_{l=1}^{M}\left(\mathcal{R}_n^{I(l)} - \overline{\mathcal{R}_n^I}\right)^2 \tag{2.2.7}$$

$$T_2 = \frac{1}{M}\sum_{l=1}^{M}\left(\mathcal{R}_{n,h}^{(l)} - \overline{\mathcal{R}_{n,h}}\right)^2 \tag{2.2.8}$$

$$T_3 = \left(\overline{\mathcal{R}_n^I} - \overline{\mathcal{R}_{n,h}}\right)^2 \tag{2.2.9}$$

$$T_4 = -\frac{2}{M}\sum_{l=1}^{M}\left(\mathcal{R}_n^{I(l)} - \overline{\mathcal{R}_n^I}\right)\left(\mathcal{R}_{n,h}^{(l)} - \overline{\mathcal{R}_{n,h}}\right) \tag{2.2.10}$$

$$T_5 = \frac{2}{M}\sum_{l=1}^{M}\left[\left(\mathcal{R}_n^{I(l)} - \overline{\mathcal{R}_n^I}\right) - \left(\mathcal{R}_{n,h}^{(l)} - \overline{\mathcal{R}_{n,h}}\right)\right]\left(\overline{\mathcal{R}_n^I} - \overline{\mathcal{R}_{n,h}}\right). \tag{2.2.11}$$

Let us interpret these terms:

- T_1 is the empirical variance of the ideal risk estimates $\mathcal{R}_n^{I(l)}$ across realizations.
- T_2 is the empirical variance of the cross-validation risk estimates $\mathcal{R}_{n,h}^{(l)}$ across realizations.
- T_3 is the squared difference between the expected ideal risk and the expected cross-validation risk; it represents the squared *bias* between the two methods.

- T_4 is minus twice the empirical covariance between $\mathcal{R}_n^{I(l)} - \overline{\mathcal{R}_n^I}$ and $\mathcal{R}_{n,h}^{(l)} - \overline{\mathcal{R}_{n,h}}$.
- T_5 involves the difference between the centered risks and the mean difference, scaled by the mean difference.

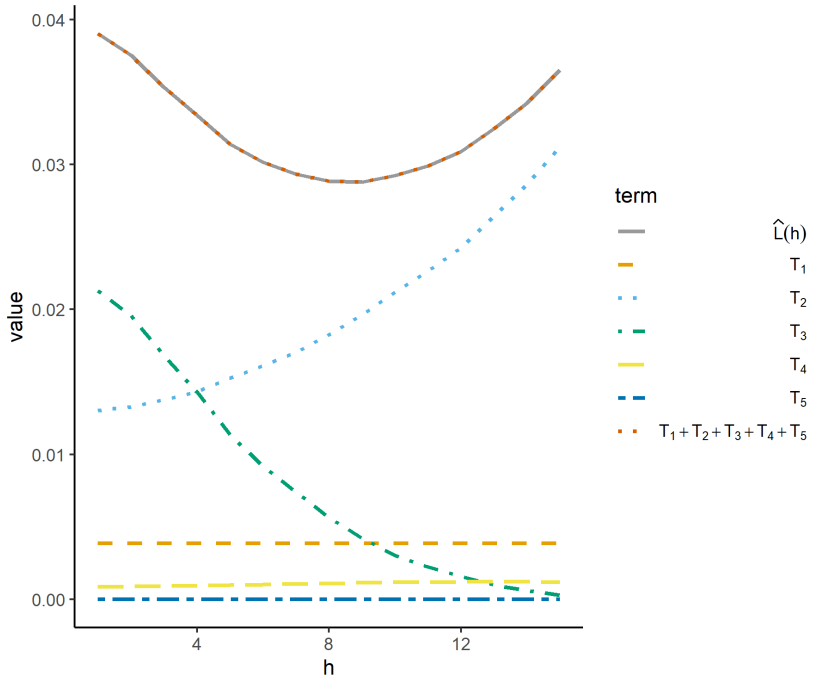


Figure 2.1: Decomposition of $\hat{\mathcal{L}}(h)$ into components T_1 to T_5 for $\theta = 5$.

To visualize the contributions of these terms to $\hat{\mathcal{L}}(h)$, we conduct the following study. Using the same simulation design as in Chapter 1, we consider $M = 100$ simulations of $\mathbf{Y}$ with a non-stationary trend, with $\theta \in \{0.5, 1, 2, \ldots, 20\}$. We compute the empirical risks $\mathcal{R}_n^{I(l)}$ and $\mathcal{R}_{n,h}^{(l)}$ for $l = 1, \ldots, M$ and $h \in \mathcal{H} = \{1, \ldots, 20\}$, $\mathcal{R}_{n,h}^{(l)}$ being obtaind using a SLOO procedure. Then we evaluate $\hat{\mathcal{L}}(h)$. We present the results for $\theta = 5$ given that the results are similar for the other values of θ. From Figure 2.1 and equation (2.2.7), we observe that T_1 remains constant with respect to h, indicating that the variance of the ideal risk estimates does not depend on h. T_4 and T_5 are relatively small and do not vary significantly with h, suggesting minimal impact on the behavior of $\hat{\mathcal{L}}(h)$. On the contrary, T_2 and T_3 show substantial variation with h and dominate the behavior of $\hat{\mathcal{L}}(h)$.

Specifically T_2 represents the variance of the cross-validation risk estimates. As h increases, the buffer size enlarges, reducing the number of training observations. This leads to increased variability in the estimator $\hat{\boldsymbol{\beta}}$ and, consequently, an increase in T_2.

T_3 is the squared bias between the estimations of the expected ideal risk and the expected cross-validation risk. At smaller values of h, the cross-validation risk is biased due

to spatial autocorrelation between training and validation sets. As h increases, this bias decreases because the buffer reduces spatial dependence, leading to a decrease in T_3.

Figure 2.2 illustrates the dominant terms T_2 and T_3, and their combined effect $T_2 + T_3$ on $\hat{\mathcal{L}}(h)$. Notably, the curve $T_2 + T_3$ shows the same general behavior as the five-term expansion of $\hat{\mathcal{L}}(h)$, but with a uniformly lower overall level, reflecting that T_1, T_4, and T_5 have relatively minor constant contributions. The vertical line indicates the value of h where $\hat{\mathcal{L}}(h)$ is minimum. We observe that this minimum is achieved where the increase in T_2 (variance) balances the decrease in T_3 (bias). This exemplifies the classical *bias-variance trade-off*: as h grows, bias decreases but variance increases. The curve for $T_2 + T_3$ mirrors this trade-off as well. We can conclude that minimizing $\hat{\mathcal{L}}(h)$ effectively calibrates the balance between bias and variance in cross-validation risk estimates.

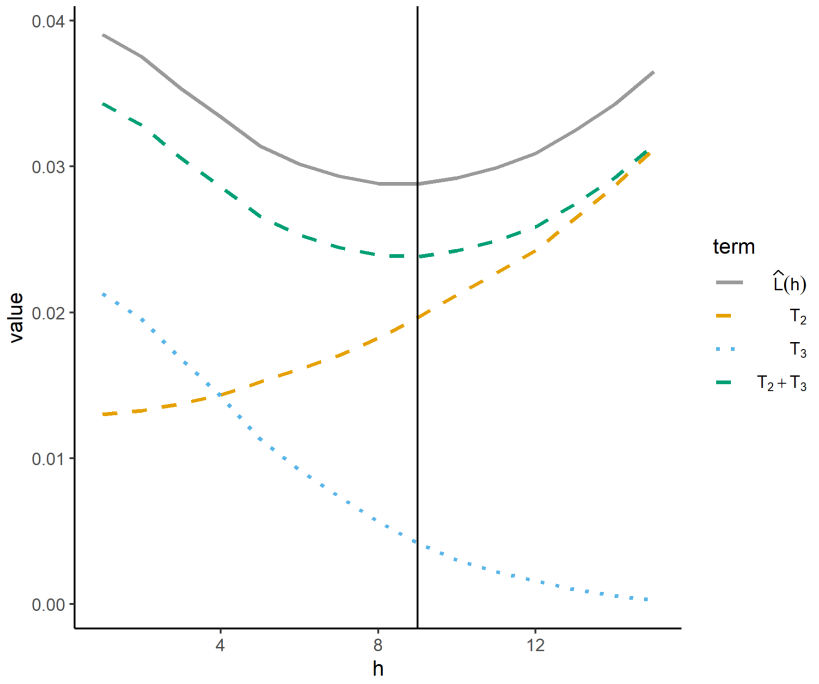


Figure 2.2: Dominant terms T_2 and T_3 contributing to $\hat{\mathcal{L}}(h)$ for $\theta = 5$. The green curve shows the sum $T_2 + T_3$. The vertical line indicates the value of h where $\hat{\mathcal{L}}(h)$ reaches its minimum.

Selecting $h_I = \arg\min_{h \in \mathcal{H}} \hat{\mathcal{L}}(h)$ as the buffer hyperparameter in spatial cross-validation should yield cross-validation risk estimates that approximate the ideal risk as closely as possible. The challenge, then, is to devise a procedure or method to estimate this hyperparameter h_I in practice. On what follows, first, we present a naive and fast method to approximate h_I. Then, in Chapter 3, we propose to mimic the empirical ideal risk by replacing the M copies of Y by bootstrap copies of Y.

2.3 A naive choice of h

In this section we discuss two existing approaches to determine h; a variogram-range method, proposed by Le Rest et al. (2014); and the Moran's I method, proposed by Karasiak et al. (2022). Then we propose an alternative procedure that we call the SAC-0.1 method.

2.3.1 The variogram-range method

Le Rest et al. (2014) and Telford and Birks (2009) propose defining h as the range of the variogram estimated from the model's residuals. A residual's variogram describes how the variance of the residual differences changes with distance, reflecting the spatial correlation structure of those residuals. Formally, for a random field $\{\varepsilon(\mathbf{s}) : \mathbf{s} \in \mathcal{S}\}$, the variogram $\gamma(\cdot)$ is given by

$$\gamma(\mathbf{h}) = \frac{1}{2}\,\mathrm{Var}\big(\varepsilon(\mathbf{s}+\mathbf{h}) - \varepsilon(\mathbf{s})\big), \tag{2.3.1}$$

assuming stationarity of increments. In most practical contexts, this is expressed as $\gamma(h)$ by letting $h = \|\mathbf{h}\|$.

In many geostatistical settings, the variogram approaches a plateau at large distances. This plateau is called the *sill*, and it represents the limiting value of $\gamma(h)$ once spatial correlation is effectively lost. The *range* is then defined as the distance at which $\gamma(h)$ has reached the sill.

The key assumption is that once the variogram attains its range, residuals can be treated as practically uncorrelated. Consequently, defining h as this range should reduce spatial dependence between training and validation sets.

We still consider a spatial model of the form

$$Y(\mathbf{s}) \;=\; \mathbf{X}(\mathbf{s})\boldsymbol{\beta} + \varepsilon(\mathbf{s}).$$

After an estimation procedure, we obtain the residuals $\hat{\varepsilon}(\mathbf{s}) = Y(\mathbf{s}) - \mathbf{X}(\mathbf{s})\hat{\boldsymbol{\beta}}$. An experimental variogram $\hat{\gamma}(h)$ of these residuals is then calculated. For example, let us consider the exponential variogram model,

$$\gamma(h) \;=\; c_0 + c\left[1 - \exp\!\Big(-\tfrac{h}{a}\Big)\right].$$

Then, the distance at which $\hat{\gamma}(h)$ reaches its sill $c_0 + c$ can serve as the buffer size h_{range}. Although this approach aligns well with geostatistical principles, it presupposes a specific variogram model, and misspecification can compromise the accuracy of h_{range}.

2.3.2 The Moran's I Method

Karasiak (2022) proposes a method based on Moran's I (Moran, 1950), which is a model-free measure of spatial autocorrelation. Moran's index I_M for observations $Y(\mathbf{s})$ at locations $\mathbf{s} \in S_n$ is typically computed by discretizing the set of distances into bins $\{h_1, h_2, \dots\}$. For each bin h, one defines weights $w_{i,j}$ indicating whether $d(\mathbf{s}_i, \mathbf{s}_j) \approx h$. Moran's statistic is defined by,

$$I_M = \frac{n}{S_0} \frac{\sum_{i,j} w_{i,j} \big(Y(\mathbf{s}_i) - \overline{Y}\big)\big(Y(\mathbf{s}_j) - \overline{Y}\big)}{\sum_{i=1}^{n} \big(Y(\mathbf{s}_i) - \overline{Y}\big)^2} \tag{2.3.2}$$

where $S_0 = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j}$. There exist statistical procedures to test the null hypothesis of no spatial correlation measured by I_M. Typically, one applies a permutation test. Karasiak and his co-authors approach is to evaluate Moran's I as a function of distance h and identifies the distance at which it becomes statistically insignificant. The reasoning is that, beyond this distance, the data no longer exhibit detectable spatial correlation. Defining the buffer size in cross-validation at that point should help to mitigate bias due to spatial dependence, while avoiding the need to fit a variogram or assume a specific spatial correlation model.

If we denote by $I_M(h)$ the Moran's I statistic as a function of lag distance h, and suppose that a statistical test indicates that $I_M(h)$ is no longer significant at some $h = h_{\text{Moran}}$, we define h_{Moran} as the buffer size. This approach is purely data-driven and does not require a particular model structure, but it can be computationally intensive for large datasets and sensitive to the choice of distance bins and significance thresholds.

Karasiak et al. (2022) adopt a simplified version of this criterion for one application by declaring independence when $I_M(h)$ itself reaches a value of 0.05, implying that once the observed Moran's I drops to 0.05, residual spatial correlation is deemed negligible substituting a formal significance test for a fixed cutoff. We will denote this distance by $h_{\text{Moran}_{0.05}}$.

2.4 The SAC-0.1 method

In this section, we propose an alternative method, called the SAC-0.1 method, that directly connects the optimal buffer size with the level of *spatial autocorrelation* in the data. Throughout, we refer to spatial autocorrelation as *SAC*. Specifically, our approach examines the spatial autocorrelation of the residuals, post-estimation, to inform the choice of buffer size.

Spatial Autocorrelation

For a spatial process $\mathbf{Z}$, the spatial autocorrelation (SAC) at location $\mathbf{s} \in S_n$ in the direction $\mathbf{e}$ and at distance k is defined by:

$$\rho_{\mathbf{e}}(s,k) = \frac{\text{cov}(Z(s+k\mathbf{e}), Z(s))}{\text{var}(Z(s))} = \frac{\mathbb{E}[(Z(s) - \mathbb{E}[Z(s)])(Z(s+k\mathbf{e}) - \mathbb{E}[Z(s+k\mathbf{e})])]}{\mathbb{E}[(Z(s) - \mathbb{E}[Z(s)])^2]}. \tag{2.4.1}$$

The empirical global autocorrelation from spatial averages is given by:

$$\hat{\rho}_{\mathbf{e}}(k) = \frac{1}{|S_{\mathbf{e},k}|} \sum_{s \in S_{\mathbf{e},k}} \frac{(Z(s) - \overline{Z})(Z(s+k\mathbf{e}) - \overline{Z})}{\sum_{s \in S_n}(Z(s) - \overline{Z})^2} \tag{2.4.2}$$

where $S_{\mathbf{e},k} = \{s \in S_n : s + k\mathbf{e} \in S_n\}$ and $\overline{Z} = \frac{1}{n}\sum_{s \in S_n} Z(s)$. Let us note that $\hat{\rho}_{\mathbf{e}}(.)$ can be computed without any model assumption.

A first idea is to choose for the optimal buffer size the lag at which the spatial autocorrelation becomes non-significant; this criterion would be analogous to the range-variogram method or the Moran's index method. Our approach is different.

First, let us note that $\hat{\mathcal{L}}(h)$ reaches a minimum as a function of h. This is seen in Figure 1 in the case $\theta = 5$ but in fact it happens for any value of θ. This minimum gives the optimal buffer size h_I in terms of approaching the empirical ideal risk.

Determining h_I needs M copies of the process $\mathbf{Y}$ which is unrealistic in practice. We then propose to use the spatial autocorrelation to approach h_I.

Together with $\hat{\mathcal{L}}(h)$ we also compute the spatial correlation $\hat{\rho}_{\mathbf{e}}(h)$ at distance h, and we evaluate $\hat{\rho}_{\mathbf{e}}(h_I)$. Our finding is that $\hat{\rho}_{\mathbf{e}}(h_I)$ is always close to 0.1 for any value of θ and under different modelling scenarios. This gives **our criterion**: we select the buffer size that we will denote by now $h_{0.1}$ as being the lag at which the spatial autocorrelation reaches 0.1,

$$h_{0.1} = \underset{h \in \mathcal{H}}{\text{argmin}} \, |\hat{\rho}(h) - 0.1|. \tag{2.4.3}$$

Let us present our results in details. For each lag, we compute the empirical spatial correlation $\hat{\rho}_{\mathbf{e}}(\cdot)^{(l)}$, defined in (2.4.2), for $l = 1, \ldots, M$, and average the spatial autocorrelations to obtain:

$$\hat{\rho}_{\mathbf{e}}(\cdot) = \frac{1}{M}\sum_{l=1}^{M} \hat{\rho}_{\mathbf{e}}(\cdot)^{(l)}. \tag{2.4.4}$$

We present the results for the spatial autocorrelation computed along the horizontal direction. The findings are consistent with those obtained in the vertical direction, which aligns with the assumption of isotropy in our analysis. We will then simplify the notation of $\hat{\rho}_{\mathbf{e}}(\cdot)$ by $\hat{\rho}(\cdot)$.

For each θ, which characterizes the strength of the dependence, we compute the averaged spatial autocorrelation $\hat{\rho}(\cdot)$ defined in (2.4.4) at different lags (on the residuals). Figures 3 to 14 display the plots of both $\hat{\mathcal{L}}(h)$ and $\hat{\rho}(h)$ for different values of θ and under different scenarios. Tables 1 to 6 give the values of h_I, $h_{0.1}$, the associated values $\hat{\mathcal{L}}(h_I)$, $\hat{\mathcal{L}}(h_{0.1})$, and the absolute and relative deviations between those two values.

Scenarios

We consider several scenarios based on two types of regression models: one with a constant trend and one with a non-stationary trend. In the constant-trend model,

$$\mathbf{Y} = \boldsymbol{\varepsilon}. \tag{2.4.5}$$

In the non-stationary trend model, we use the same specification as in Chapter 1, namely

$$Y(\mathbf{s}) = \mathbf{X}(\mathbf{s})\boldsymbol{\beta} + \varepsilon(\mathbf{s}), \tag{2.4.6}$$

where

$$\mathbf{X}(\mathbf{s}) = (1,\ s_1 - 40,\ s_2 - 30) \quad \text{and} \quad \boldsymbol{\beta} = (0.9,\ 1/32,\ 1/24).$$

We conduct simulation studies under each scenario for various strengths of spatial dependence, denoted by $\theta \in \{0.5, 1, 2, \ldots, 20\}$. The same simulation design as in Chapter 1 is used: $\mathcal{S}_n$ is a rectangular 80×60 lattice of sites, hence $n = 4800$.

For each trend scenario, we run $M = 100$ simulations for each value of θ. We estimate the parameters by ordinary least squares of $\boldsymbol{\beta}$. In the case $\mathbf{Y} = \boldsymbol{\varepsilon}$, we estimate the model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, with $\mathbf{X} = (\mathbf{1}, \mathbf{X}_1, \mathbf{X}_2)$ where $\mathbf{X}_1 = \mathbf{s}_1$ and $\mathbf{X}_2 = \mathbf{s}_2$ are linked to the locations of the observations. In the case of the non-stationary trend, $\boldsymbol{\beta}$ is estimated under the true model. This allows us to explore both a simplified scenario with model misspecification and a scenario where the true model is correctly specified. Then we consider different spatial cross-validation procedures in the evaluation of $\mathcal{R}_{n,h}$ and then $\hat{\mathcal{L}}(h)$; we consider SLOO and SBCV methods with different block sizes. This gives six scenarios.

- Constant trend and SLOO
- Constant trend and SBCV with blocks of size 4×6
- Constant trend and SBCV with blocks of size 8×12
- Non-stationary trend and SLOO
- Non-stationary trend and SBCV with blocks of size 4×6
- Non-stationary trend and SBCV with blocks of size 8×12

Our general findings is that h_I is very close to the lag at which $\hat{\rho}(.)$ reaches 0.1. Interestingly this holds across all values of θ. This provides us a straighforward method to determine the appropriate buffer size needed for both the SLOO and SBCV spatial cross-validation procedures. Our results are consistent across both trend scenarios and the three spatial cross-validation procedures that are considered.

In what follows, we present a detailed analysis of our results. For each value of θ, we report the *ideal* radius h_I and the corresponding minimum discrepancy:

$$\hat{\mathcal{L}}(h_I) = \min_{h \in \mathcal{H}} \hat{\mathcal{L}}(h). \tag{2.4.7}$$

Next, we present h_I's approximation $h_{0.1}$ and its associated discrepancy $\hat{\mathcal{L}}(h_{0.1})$. Additionally we compute the absolute deviation between the discrepancies $\hat{\mathcal{L}}(h_I) - \hat{\mathcal{L}}(h_{0.1})$ and the relative deviation with respect to $\hat{\mathcal{L}}(h_I)$, that is $\dfrac{\hat{\mathcal{L}}(h_I) - \hat{\mathcal{L}}(h_{0.1})}{\hat{\mathcal{L}}(h_I)}$.

Constant trend scenario

Figure 2.3 illustrates the results for weak to medium spatial dependence in the SLOO method, while Figures 2.5 and 2.7 display SBCV results for block sizes 4×6 and 8×12, respectively. Figures 2.4, 2.6 and 2.8 present the results for stronger dependence. In all these figures the plots on top represent the average sum of squared differences between the corresponding spatial cross-validation risk and the ideal risk as a function of buffer size $\hat{\mathcal{L}}(h)$. The lower plot displays the SAC measure across different lags. The optimal buffer, indicated by a vertical red line, is determined at the point where the top curve reaches its minimum as defined in (2.2.2), signifying the smallest deviation from the ideal risk. The vertical blue line represents the buffer closest to an SAC value of 0.1 as defined in equation (2.4.3). Due to space constraints, only a subset of θ values is depicted, but comprehensive results for the entire range of θ values can be found in Tables 2.1, 2.2 and 2.3.

Constant trend and SLOO

The behavior of the discrepancy function $\hat{\mathcal{L}}(h)$ for SLOO under weak spatial dependence, as illustrated in the top left panel of Figure 2.3, aligns with expectations. The minimum discrepancy occurs at a small buffer size, indicating that even classical LOO cross-validation ($h = 0$) would provide a reasonably accurate risk estimate. As the buffer size increases, the discrepancy also grows, likely due to the reduction in available data points for training. Moving across the top panels of Figure 2.3, where spatial dependence strengthens, the effectiveness of classical LOO diminishes, necessitating a greater separation between the training and validation sets to minimize the discrepancy.

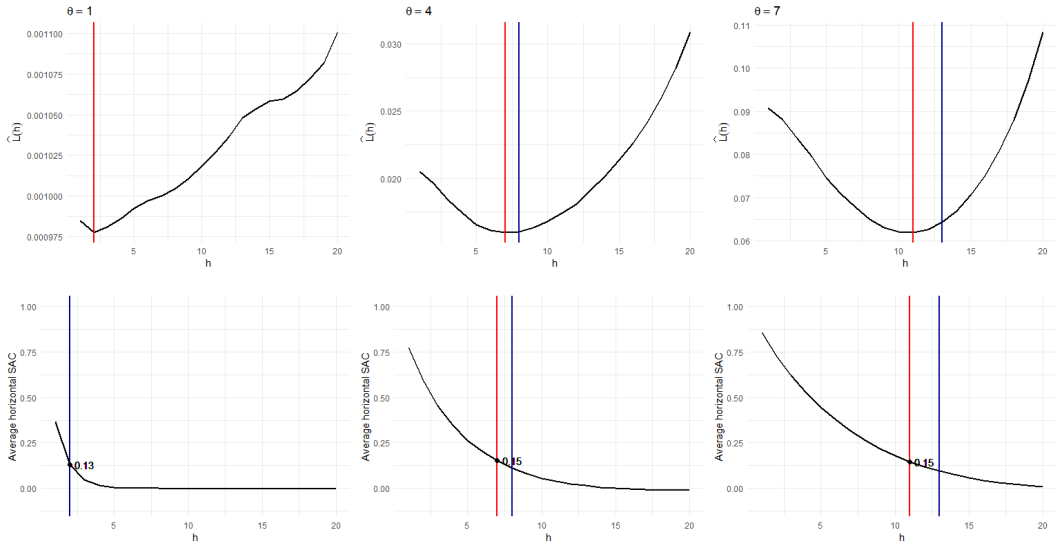


Figure 2.3: Plots of $\hat{\mathcal{L}}(h)$ (first row) for SLOO and spatial autocorrelation (second row) for $\theta = 1,\ 4$, and 7 in the scenario of a constant trend. The optimal buffer h_I is indicated by a vertical red line, and $h_{0.1}$ is located by the vertical blue line.

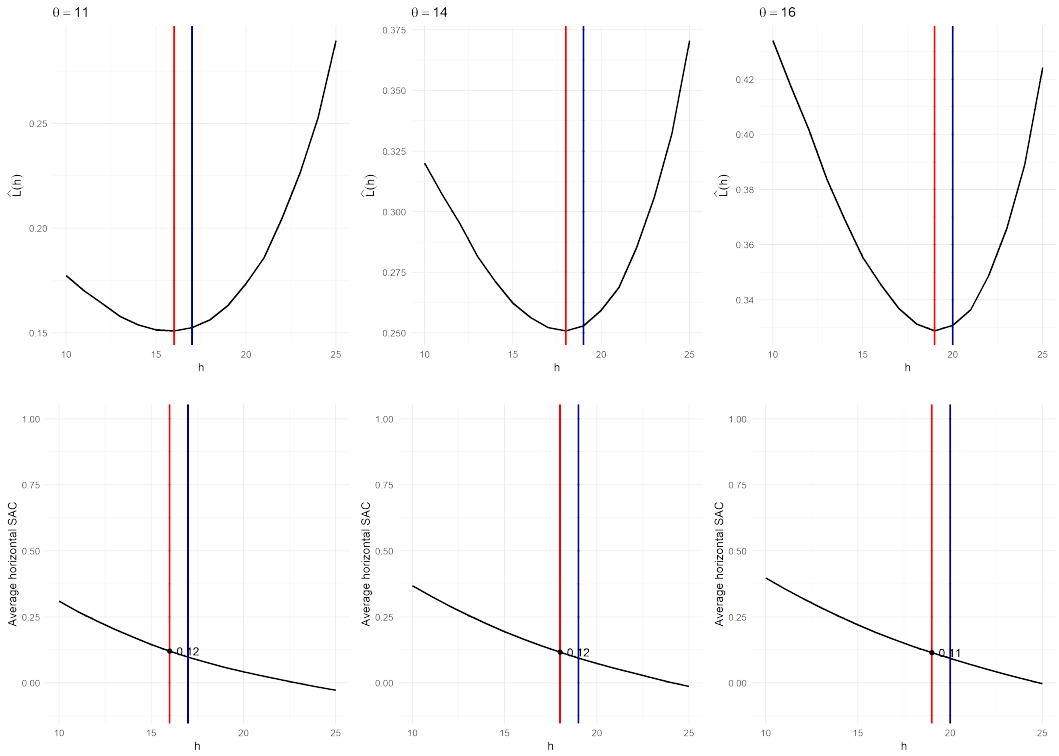


Figure 2.4: Plots of $\hat{\mathcal{L}}(h)$ (first row) for SLOO and spatial autocorrelation (second row) for $\theta = 11,\ 14$, and 16 in the scenario of a constant trend. The optimal buffer h_I is indicated by a vertical red line, and $h_{0.1}$ is located by the vertical blue line.

As spatial dependence intensifies, as seen in the top middle and top right panels of Figure 2.3 and top left panel of Figure 2.4, the minimum discrepancy still exists, but it appears within a broader zone. Multiple buffer sizes perform similarly, suggesting some flexibility in choosing a good buffer size in scenarios with medium spatial dependence.

In the scenarios of higher spatial dependence, depicted in the top middle and top left panels of Figure 2.4, the discrepancy curve narrows significantly, showing a pronounced and sharp minimum. This pattern likely arises due to the more substantial data loss required to achieve independence between the training and validation sets. Here, the discrepancy initially drops steeply as the buffer size increases, reaches a distinct minimum, and then rapidly climbs again. This behavior indicates that, in cases of strong spatial dependence, the optimal buffer size is critically constrained, often approaching half the length of one side of the spatial lattice, highlighting the challenge of finding the right balance between independence and data availability.

θ	h_I	$\hat{\mathcal{L}}(h_I)$	$h_{0.1}$	$\hat{\mathcal{L}}(h_{0.1})$	$\hat{\mathcal{L}}(h_{0.1}) - \hat{\mathcal{L}}(h_I)$	$\frac{(\hat{\mathcal{L}}(h_{0.1}) - \hat{\mathcal{L}}(h_I))}{\hat{\mathcal{L}}(h_I)}$
0.5	1	0.000509	1	0.000509	0.000000	0%
1	2	0.000978	2	0.000978	0.000000	0%
2	4	0.003474	5	0.003505	0.000030	0.8672%
3	6	0.008312	7	0.008388	0.000076	0.9011%
4	7	0.015969	8	0.016024	0.000055	0.3450%
5	9	0.027187	10	0.027632	0.000445	1.6111%
6	10	0.042421	11	0.042932	0.000510	1.1896%
7	11	0.061899	13	0.064362	0.002462	3.8265%
8	12	0.085650	14	0.089169	0.003519	3.9470%
9	13	0.113678	15	0.118814	0.005136	4.3228%
10	15	0.122834	16	0.123703	0.000869	0.7025%
11	16	0.150939	17	0.152425	0.001486	0.9751%
12	17	0.181953	18	0.184362	0.000241	1.3066%
13	17	0.215212	18	0.215882	0.000670	0.3144%
14	18	0.250901	19	0.253004	0.002103	0.8313%
15	19	0.289298	19	0.289298	0	0%
16	19	0.328760	20	0.355497	0.001958	0.5920%
17	23	0.218303	20	0.264982	0.007028	3.1191%
18	24	0.241364	20	0.303125	0.013322	5.2308%
19	25	0.265218	21	0.343512	0.014449	5.1666%
20	25	0.290384	21	0.371778	0.021895	7.0112%

Table 2.1: Buffer sizes and comparison of their associated SLOO risks values in the scenario of a constant trend.

The challenges posed by settings with stronger spatial dependence are clearly reflected in Table 2.1. As the spatial dependence parameter θ increases, the minimum discrepancy value $\hat{\mathcal{L}}(h_I)$ also rises, indicating a greater difficulty in accurately estimating the risk. Specifically, the minimum discrepancy starts at 5.1×10^{-4} for $\theta = 0.5$ and progressively increases, reaching 0.29 for $\theta = 20$.

The values of $\hat{\mathcal{L}}(h_{0.1})$ and $\hat{\mathcal{L}}(h_I)$ shown in Table 2.1 are closely aligned across different values of θ, reflecting a strong concordance between the buffer size that minimizes the SLOO risk and the buffer size associated with an SAC of 0.1. For most values of θ, the percentage differences between these two criteria are minimal, often under 1%. However, as θ increases, particularly from 18 onwards, there is a noticeable divergence between the two criteria, with the percentage difference exceeding 7% when $\theta = 20$. This suggests that while the 0.1 criterion and the minimization of SLOO risk tend to agree for smaller values of θ, the alignment weakens as spatial dependence strengthens, highlighting the need for more robust criteria under these conditions.

SBCV with block size 4×6

While the SAC-0.1 method appears to hold for SBCV with the smaller block size (4×6), the degree of deviation from the ideal buffer size h_I increases considerably when a larger block size is used.

Examining the behavior of the discrepancy function for SBCV with block size 4×6 in the top panels of Figure 2.5 and Figure 2.6, we observe a similar pattern to that of the discrepancy function for SLOO. In each case, there is a clear minimum value, indicating the optimal buffer size. In scenarios with low and medium spatial dependence, there exists an optimal region where the discrepancy values remain relatively similar across a range of buffer sizes, suggesting robustness in the selection of h. However, in scenarios with high spatial dependence, selecting the optimal buffer size becomes more challenging, as evidenced by a sharper and more constrained minimum in the discrepancy function.

A key difference is that the minimum of the discrepancy function for SBCV with block size 4×6 is reached at a lower buffer size h compared to SLOO. This indicates that, for SBCV with smaller block sizes, a smaller buffer is sufficient to minimize the discrepancy, potentially due to the reduced spatial autocorrelation after applying the blocking strategy.

In Table 2.2, we present the results for SBCV with the smaller block size (4×6). We observe that, for smaller values of θ (indicating weaker spatial dependence), the percentage differences between the SAC-0.1 method's discrepancy $\hat{\mathcal{L}}(h_{0.1})$ and the ideal discrepancy $\hat{\mathcal{L}}(h_I)$ are relatively modest, generally up to 10%. However, for θ values between 7 and 12, the deviations become more pronounced, reaching up to 19% when $\theta = 9$. Interestingly, for $\theta > 12$, the deviations diminish once more, dropping back below 10%.

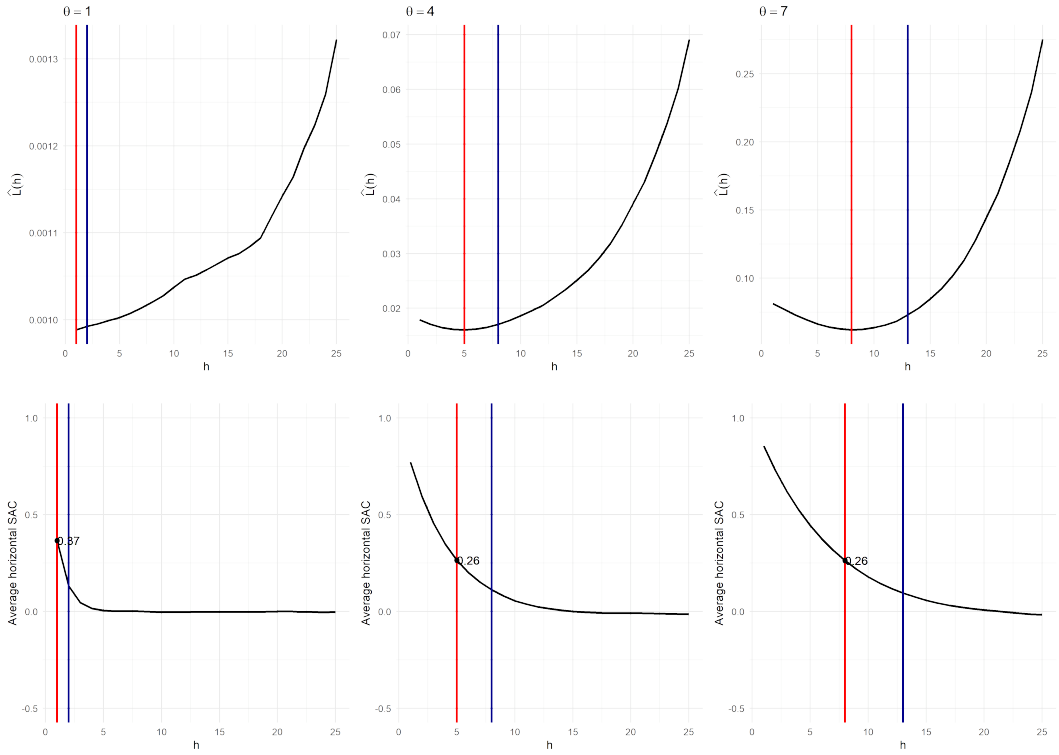


Figure 2.5: Plots of $\hat{\mathcal{L}}(h)$ (first row) for SBCV with block size 4×6 and spatial autocorrelation (second row) for $\theta = 11,\ 14$, and 16 in the scenario of a constant trend. The optimal buffer h_I is indicated by a vertical red line, and $h_{0.1}$ is located by the vertical blue line.

This pattern indicates that while the SAC-0.1 method remains reasonably effective for smaller block sizes in SBCV under weak spatial dependence, its accuracy diminishes in the presence of moderate spatial dependence. The increased deviations at intermediate θ values suggest that the SAC-0.1 method may not adequately adjust for the spatial autocorrelation affecting the residuals during cross-validation in these scenarios. Furthermore, we observe that the SAC-0.1 method consistently overestimates the optimal buffer size h_I. This overestimation can be particularly problematic when the optimal buffer size is already large due to strong spatial dependence. The resulting large buffer can reduce the effective sample size available for model training.

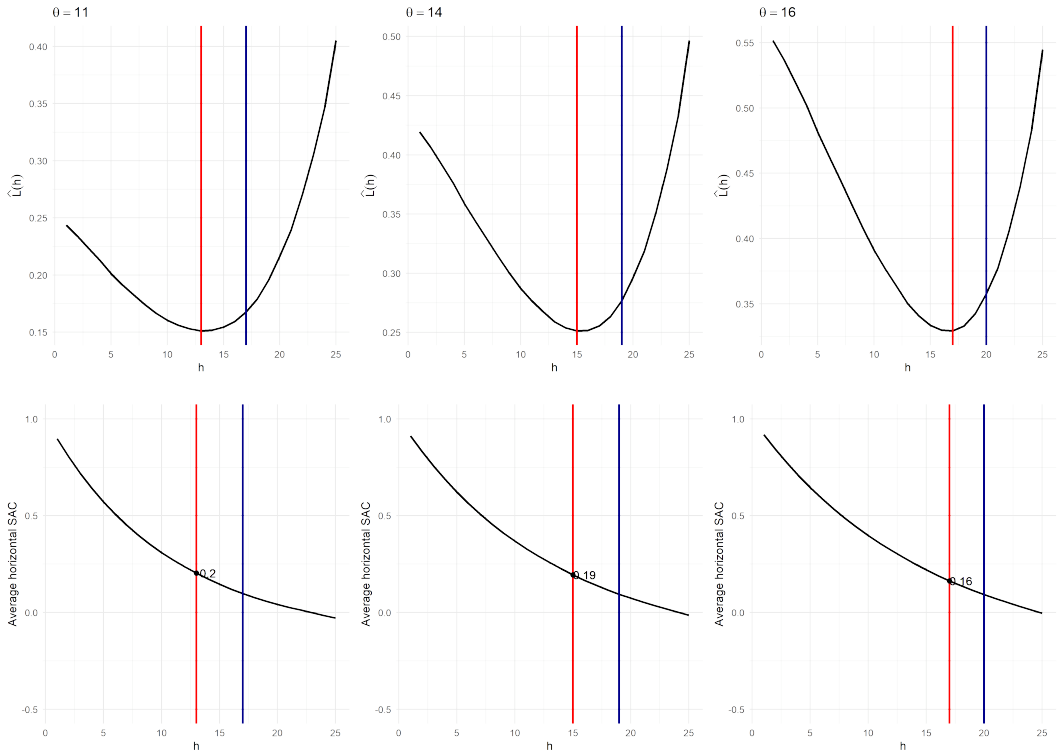


Figure 2.6: Plots of $\hat{\mathcal{L}}(h)$ (first row) for SBCV with block size 4×6 and spatial autocorrelation (second row) for $\theta = 11,\ 14$, and 16 in the scenario of a constant trend. The optimal buffer h_I is indicated by a vertical red line, and $h_{0.1}$ is located by the vertical blue line.

θ	h_I	$\hat{\mathcal{L}}(h_I)$	$h_{0.1}$	$\hat{\mathcal{L}}(h_{0.1})$	$\hat{\mathcal{L}}(h_{0.1}) - \hat{\mathcal{L}}(h_I)$	$\frac{(\hat{\mathcal{L}}(h_{0.1}) - \hat{\mathcal{L}}(h_I))}{\hat{\mathcal{L}}(h_I)}$
0.5	4	0.000510	1	0.000511	0.000001	0.18%
1	1	0.000988	2	0.000992	0.000004	0.39%
2	2	0.003510	5	0.003648	0.000138	3.95%
3	3	0.008362	7	0.008896	0.000533	6.38%
4	5	0.016059	8	0.017067	0.001008	6.27%
5	6	0.027268	10	0.032382	0.002970	10.89%
6	7	0.042526	11	0.046958	0.004317	10.42%
7	8	0.062073	13	0.073073	0.011000	17.72%
8	9	0.085884	14	0.101433	0.015549	18.11%
9	10	0.113792	15	0.135725	0.021933	19.28%
10	12	0.123216	16	0.134676	0.011460	9.30%
11	13	0.151146	17	0.167588	0.016443	10.88%
12	14	0.181916	18	0.204790	0.022874	12.57%
13	15	0.215483	18	0.232802	0.017318	8.04%
14	15	0.251343	19	0.276918	0.025575	10.18%
15	16	0.289071	19	0.308334	0.019262	6.66%
16	17	0.329264	20	0.357714	0.001958	8.64%
17	20	0.219121	20	0.219122	0	0%
18	21	0.242321	20	0.243871	0.001550	0.64%
19	22	0.266244	21	0.267913	0.001670	0.63%
20	23	0.290730	21	0.295900	0.005171	1.78%

Table 2.2: Buffer sizes and comparison of their associated SBCV risks values in the scenario of a constant trend for block size 4×6

.

SBCV with block size 8×12

While looking at the Figures 2.7 and 2.8 corresponding to SBCV with block size 8×12 we see once again a similar structure of the discrepancy function as discussed before for SLOO and SBCV with block size 4×6, once again with a minimum value reached for a smaller value of h, providing further evidence of the ability of the blocking strategy to disrupt the spatial dependence structures present in the data.

When using the larger block size (8×12) in SBCV, as shown in Table 2.3, the discrepancies between the SAC-0.1 method and the ideal buffer size become even more pronounced. Here, the SAC-0.1 method consistently overestimates the ideal buffer size, as evidenced by $\hat{\mathcal{L}}(h_{0.1}) > \hat{\mathcal{L}}(h_I)$ for all considered values of θ. The deviations grow substantially even for small values of θ, with percentage differences exceeding 50% for $\theta = 7$ and higher. This trend suggests that the larger the block size, the less reliable the SAC-0.1 method becomes in capturing the true risk minimization criterion.

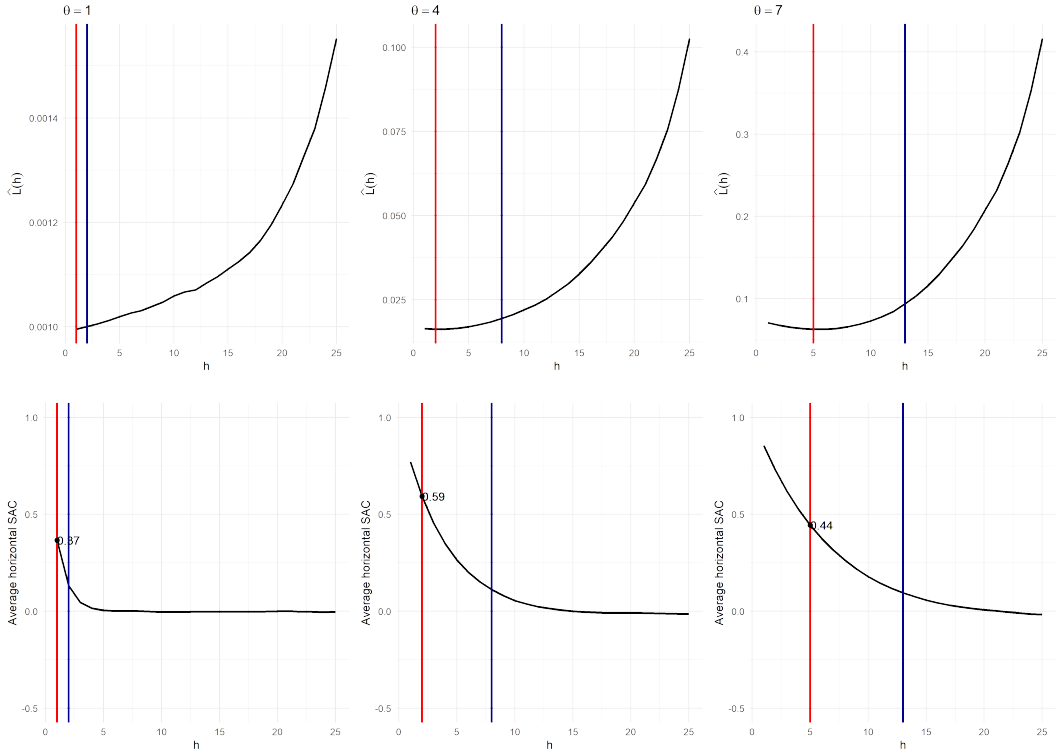


Figure 2.7: Plots of $\hat{\mathcal{L}}(h)$ (first row) for SBCV with block size 8×12 and spatial autocorrelation (second row) for $\theta = 1,\ 4$, and 7 in the scenario of a constant trend. The optimal buffer h_I is indicated by a vertical red line, and $h_{0.1}$ is located by the vertical blue line.

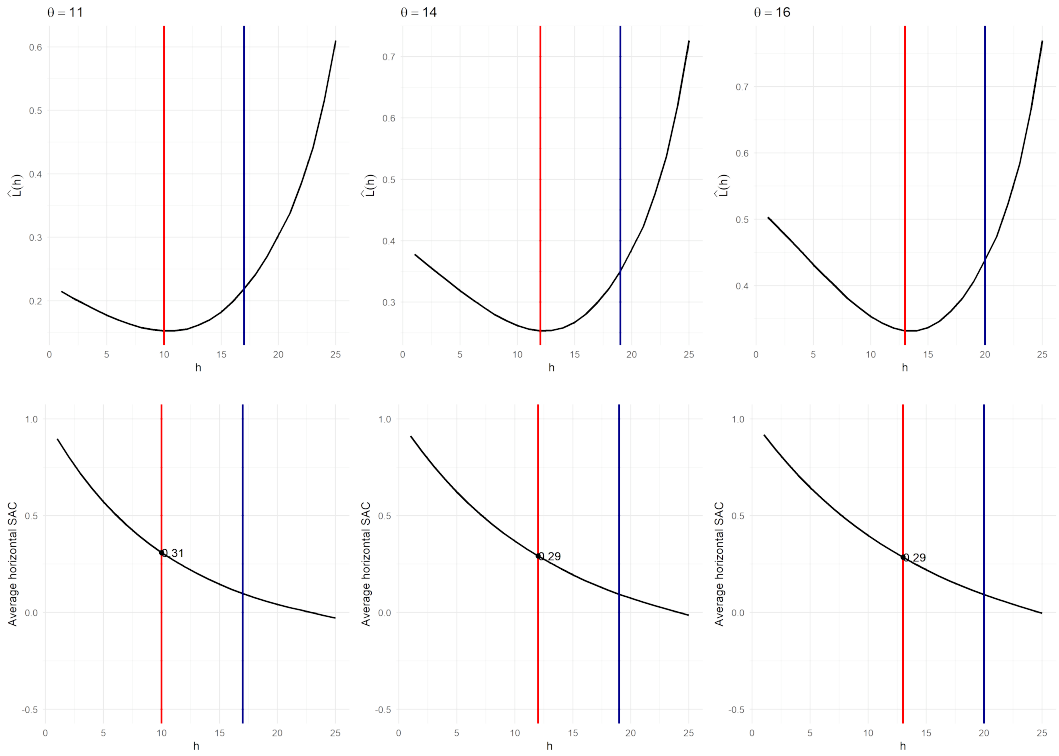


Figure 2.8: Plots of $\hat{\mathcal{L}}(h)$ (first row) for SBCV with block size 8×12 and spatial autocorrelation (second row) for $\theta = 11,\ 14$, and 16 in the scenario of a constant trend. The optimal buffer h_I is indicated by a vertical red line, and $h_{0.1}$ is located by the vertical blue line.

θ	h_I	$\hat{\mathcal{L}}(h_I)$	$h_{0.1}$	$\hat{\mathcal{L}}(h_{0.1})$	$\hat{\mathcal{L}}(h_{0.1}) - \hat{\mathcal{L}}(h_I)$	$\frac{(\hat{\mathcal{L}}(h_{0.1}) - \hat{\mathcal{L}}(h_I))}{\hat{\mathcal{L}}(h_I)}$
0.5	1	0.000511	1	0.000511	0	0%
1	1	0.000995	2	0.001000	0.000005	0.51%
2	1	0.003564	5	0.003828	0.000264	7.4%
3	1	0.008456	7	0.009787	0.001331	15.74%
4	2	0.016327	8	0.019414	0.003176	19.56%
5	3	0.027635	10	0.036067	0.008432	30.51%
6	5	0.043162	11	0.057124	0.013962	32.35%
7	5	0.062981	13	0.093826	0.030845	48.97%
8	6	0.087074	14	0.132429	0.0453548	52.08%
9	7	0.115162	15	0.179880	0.064718	56.20%
10	10	0.124259	16	0.173586	0.049326	39.70%
11	10	0.152554	17	0.218810	0.066256	43.43%
12	11	0.183421	18	0.268338	0.084917	46.30%
13	12	0.217076	18	0.294850	0.07777	35.83%
14	12	0.253300	19	0.350644	0.09734	38.43%
15	13	0.291254	19	0.377898	0.086644	29.75%
16	13	0.331785	20	0.438937	0.107152	32.30%
17	17	0.220747	20	0.229383	0.005636	2.55%
18	18	0.244050	20	0.246096	0.002046	0.84%
19	19	0.267944	21	0.270312	0.002368	0.88%
20	20	0.292561	21	0.293029	0.000468	0.16%

Table 2.3: Buffer sizes and comparison of their associated SBCV risks values in the scenario of a constant trend for block size 8×12

.

Non-stationary trend scenario

Figure 2.9 shows the results for weak to medium spatial dependence ($\theta = 1, 4, 7$), in the SLOO method, while Figures 2.11 and 2.13 display the SBCV results for block sizes 4×6 and 8×12, respectively. For stronger dependence ($\theta = 11, 14, 16$), results are presented in Figures 2.10, 2.12, and 2.14. Similarly to the constant trend scenario only a subset of θ values is shown in Figures 2.10, 2.12, and 2.14 and complete results are available in Tables 2.4, 2.5 and 2.6.

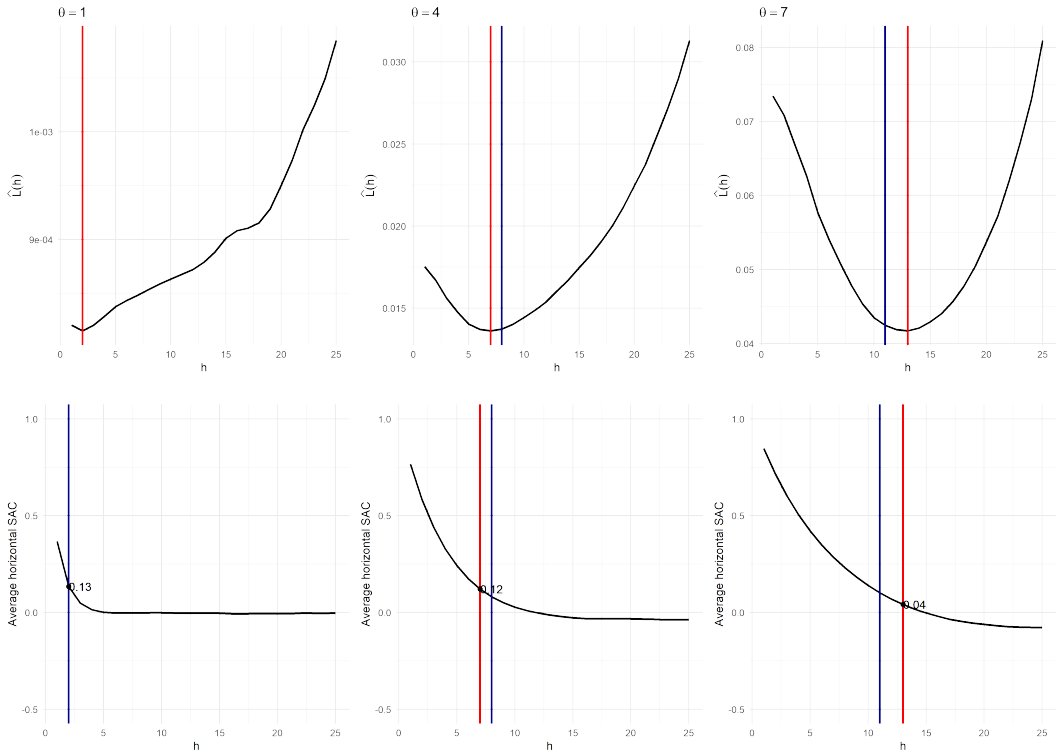


Figure 2.9: Plots of $\hat{\mathcal{L}}(h)$ (first row) for SLOO and spatial autocorrelation (second row) for $\theta = 1,\ 4$, and 7 in the scenario of a non stationary trend. The optimal buffer h_I is indicated by a vertical red line, and $h_{0.1}$ is located by the vertical blue line.

SLOO

In Table 2.4 we observe that the approximate $h_{0.1}$ coincides quite well with the optimal h_I for $\theta = 0.5$ to 14. More precisely, the results are excellent for weak dependence up to $\theta = 10$; then we observe some deviations between $\hat{\mathcal{L}}(h_I)$ and $\hat{\mathcal{L}}(h_{0.1})$, which can represent up to 10% relative to $\hat{\mathcal{L}}(h_I)$. For larger values of θ, $h_{0.1}$ underestimates the optimal buffer size, and the relative deviation between the associated risks can rise up to 20%. However, we can see that choosing $h_{0.1}$ gives a value $\hat{\mathcal{L}}(h_{0.1})$ close to its minimum $\hat{\mathcal{L}}(h_I)$, meaning that on average, the empirical risk $\mathcal{R}_{n,h_{0.1}}$ is not so far from the ideal risk $\mathcal{R}_n^{\hat{I}}$.

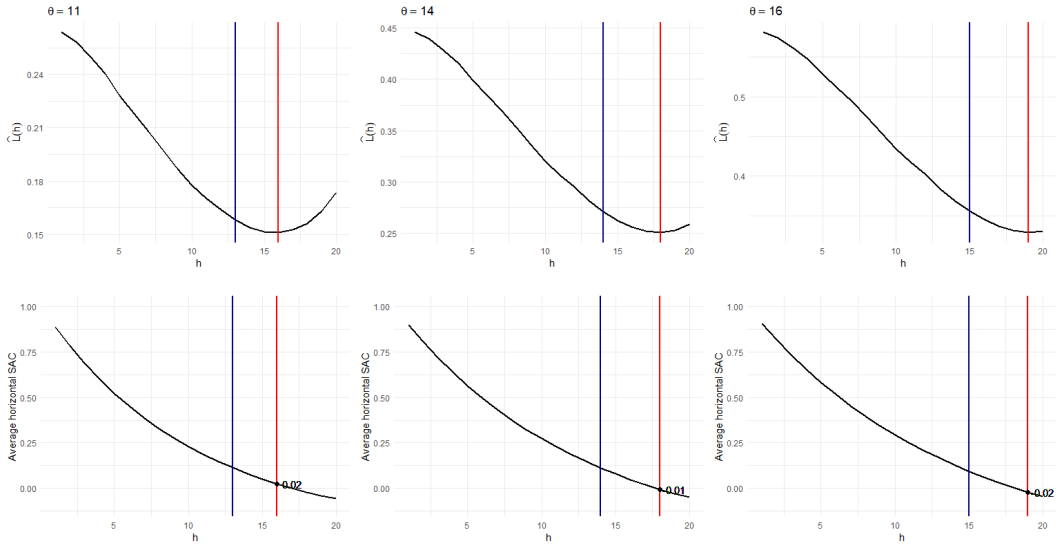


Figure 2.10: Plots of $\hat{\mathcal{L}}(h)$ (first row) for SLOO and spatial autocorrelation (second row) for $\theta = 11,\ 14$, and 16 in the scenario of a non stationary trend. The optimal buffer h_I is indicated by a vertical red line, and $h_{0.1}$ is located by the vertical blue line.

The results show that the SAC-0.1 method for determining the optimal h for SLOO behaves well except for the case of strong dependence, for $\theta = 16$ to 20; clearly, $h_{0.1}$ underestimates h_I much more. But this is expected. Indeed, we face two obstacles. Our lattice length is 80 and its width is 60; on one hand, as we increase the lag at which the spatial autocorrelation is calculated, fewer sites are involved in the computation. On the other hand, when the buffer radius reaches 20, the dead zone forms a large disc of observations around each point during the SLOO procedure, resulting in the omission of too many observations in the training step. We could change the criterion to a threshold of 0.01, but we find this approach unsatisfactory. A better option might be to consider a penalized criterion that accounts for the number of omitted observations; this is the object of another work. All in all, despite the increasing difference between $h_{0.1}$ and h_I, the percentage deviation between the associated risks remains low again. Therefore, we may conclude that the SAC-0.1 method provides a reasonable approximation of h_I, and using a larger buffer could potentially render the SLOO method ineffective.

θ	h_{I}	$\hat{\mathcal{L}}(h_{\mathrm{I}})$	$h_{0.1}$	$\hat{\mathcal{L}}(h_{0.1})$	$\hat{\mathcal{L}}(h_{0.1}) - \hat{\mathcal{L}}(h_{\mathrm{I}})$	$\frac{\hat{\mathcal{L}}(h_{0.1}) - \hat{\mathcal{L}}(h_{\mathrm{I}})}{\hat{\mathcal{L}}(h_{\mathrm{I}})}$
0.5	1	0.000375	1	0.000375	0.000000	0%
1	2	0.000815	2	0.000815	0.000000	0%
2	4	0.003311	5	0.003351	0.000040	1.22%
3	7	0.005601	6	0.005610	0.000009	0.16%
4	7	0.013620	8	0.013726	0.000105	0.77%
5	8	0.020609	9	0.020672	0.000063	0.30%
6	11	0.039018	10	0.039213	0.000194	0.49%
7	13	0.041715	11	0.042448	0.000732	1.75%
8	11	0.061415	12	0.061721	0.000306	0.4977%
9	15	0.093764	12	0.099700	0.005936	6.33%
10	15	0.120469	13	0.123649	0.003180	2.639%
11	17	0.126297	13	0.141160	0.014863	11.7680%
12	16	0.180465	13	0.188353	0.007889	4.3709%
13	15	0.284693	14	0.284741	0.000048	0.0170%
14	18	0.184455	14	0.204315	0.001986	10.7669%
15	21	0.341614	14	0.400517	0.058903	17.2426%
16	22	0.367361	14	0.451605	0.084244	22.9323%
17	22	0.352275	15	0.424286	0.072011	20.4418%
18	19	0.358603	15	0.382810	0.024206	6.7502%
19	20	0.527353	15	0.568523	0.041170	7.8069%
20	20	0.475823	15	0.514405	0.038517	8.1084%

Table 2.4: Buffer sizes and comparison of their SLOO associated risks in the scenario of a non-stationary trend.

SBCV with blocks of size 4×6

Table 2.5 present the results for SBCV with blocks size 4×6. The SAC-0.1 method generally provides accurate approximations, particularly in cases of weaker spatial dependence. Up to a θ value of around 10, the deviations between $\hat{\mathcal{L}}(h_I)$ and $\hat{\mathcal{L}}(h_{0.1})$ remain relatively minor, often below 10%. However, as spatial dependence increases, the accuracy of the SAC-0.1 method declines, with maximum deviations reaching approximately 12% at $\theta = 16$.

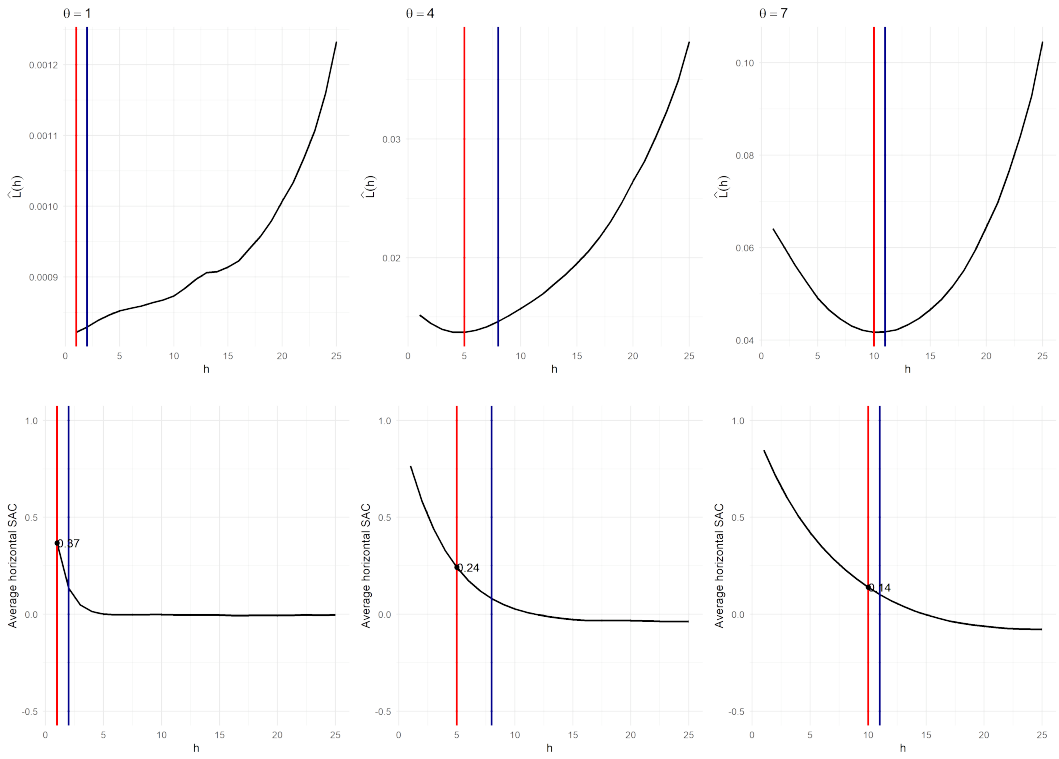


Figure 2.11: Plots of $\hat{\mathcal{L}}(h)$ (first row) for SBCV with block size 4×6 and spatial autocorrelation (second row) for $\theta = 1,\ 4$, and 7 in the scenario of a non stationary trend. The optimal buffer h_I is indicated by a vertical red line, and $h_{0.1}$ is located by the vertical blue line.

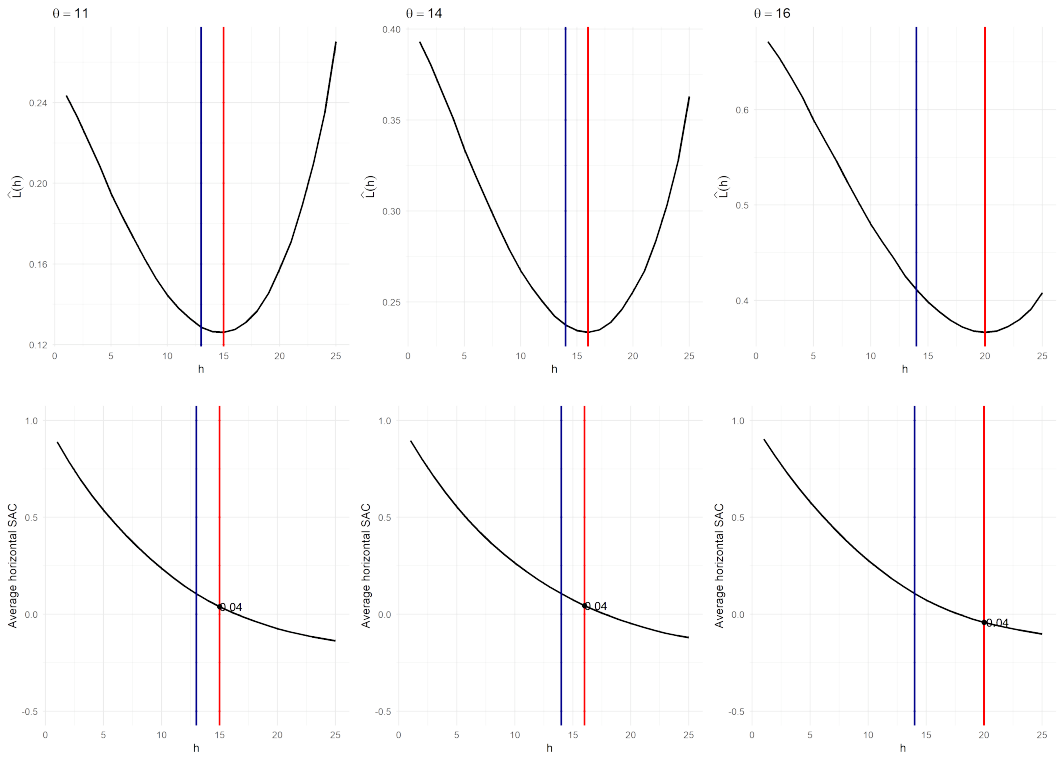


Figure 2.12: Plots of $\hat{\mathcal{L}}(h)$ (first row) for SBCV with block size 4×6 and spatial autocorrelation (second row) for $\theta = 11,\ 14$, and 16 in the scenario of a non stationary trend. The optimal buffer h_I is indicated by a vertical red line, and $h_{0.1}$ is located by the vertical blue line.

θ	h_I	$\hat{\mathcal{L}}(h_I)$	$h_{0.1}$	$\hat{\mathcal{L}}(h_{0.1})$	$\hat{\mathcal{L}}(h_{0.1}) - \hat{\mathcal{L}}(h_I)$	$\frac{(\hat{\mathcal{L}}(h_{0.1}) - \hat{\mathcal{L}}(h_I))}{\hat{\mathcal{L}}(h_I)}$
0.5	5	0.000375	1	0.000377	0.000000	0.02%
1	1	0.000822	2	0.000829	0.000007	0.91%
2	1	0.003322	5	0.003497	0.000174	5.26%
3	4	0.005627	6	0.005736	0.000108	1.91%
4	5	0.013712	8	0.014624	0.000911	6.64%
5	6	0.020627	9	0.021957	0.001330	6.45%
6	8	0.039337	10	0.040128	0.000791	2.01%
7	10	0.041708	11	0.041781	0.000074	0.18%
8	9	0.061748	12	0.067037	0.005289	8.57%
9	13	0.093945	12	0.094338	0.000343	0.42%
10	13	0.121325	13	0.121325	0	0%
11	15	0.126145	13	0.128632	0.002487	1.97%
12	13	0.181019	13	0.181019	0	0%
13	13	0.166703	14	0.166721	0.000019	0.01%
14	16	0.233397	14	0.237415	0.004018	1.72%
15	16	0.379178	14	0.383427	0.004249	1.12%
16	20	0.366519	14	0.411259	0.044740	12.21%
17	20	0.352611	15	0.386753	0.034143	9.68%
18	17	0.359293	15	0.362901	0.003608	1.00%
19	18	0.527562	15	0.539656	0.012094	2.29%
20	17	0.476353	15	0.485188	0.008352	1.85%

Table 2.5: Buffer size and comparison of their associated SBCV risks in the scenario of a non stationary trend, for block size of 4×6

.

SBCV with blocks of size 8×12

Table 2.6 present the results for SBCV with blocks size 8×12. The results are more complex than for the smaller block size. While the SAC-0.1 method initially seems to perform well, discrepancies between $h_{0.1}$ and h_I grow as spatial dependence intensifies, particularly for θ values above 14. Interestingly, while one might expect larger blocks to better mitigate spatial dependence between training and testing sets, thereby improving the reliability of risk estimates, the data suggest that this effect is nuanced. For stronger spatial dependence, the SAC-0.1 method's performance deteriorates more significantly, with some relative deviations exceeding 30%.

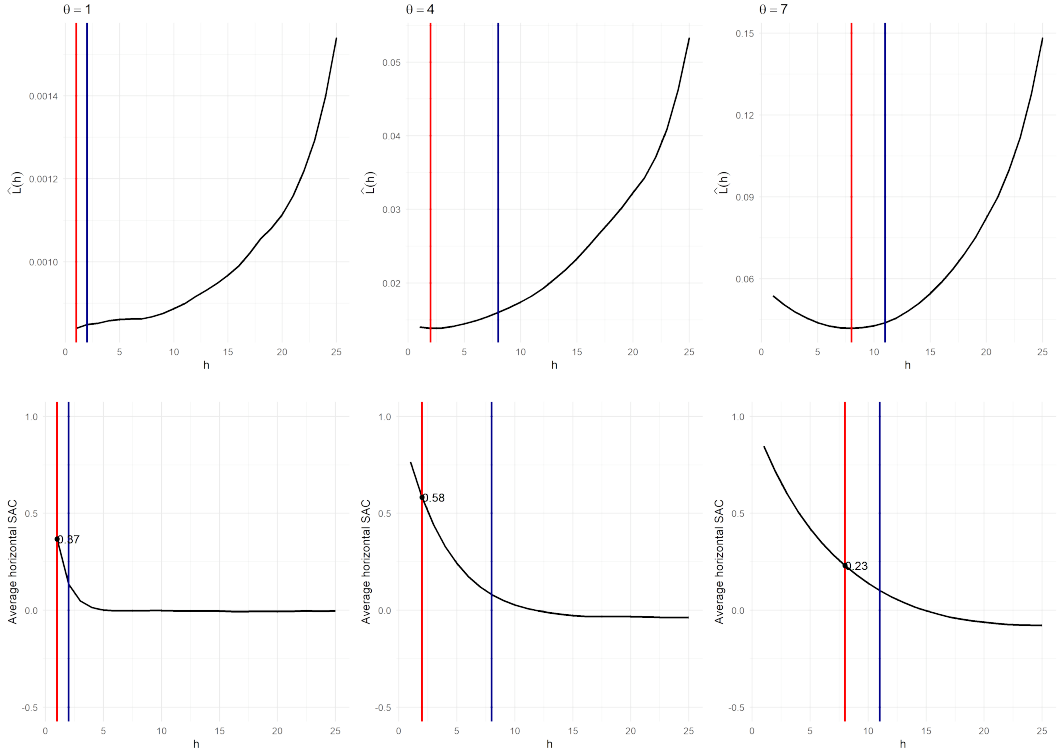


Figure 2.13: Plots of $\hat{\mathcal{L}}(h)$ (first row) for SBCV with block size 8×12 and spatial autocorrelation (second row) for $\theta = 1,\ 4$, and 7 in the scenario of a non stationary trend. The optimal buffer h_I is indicated by a vertical red line, and $h_{0.1}$ is located by the vertical blue line.

Larger blocks can indeed reduce some spatial dependence between training and testing sets, as they encapsulate larger areas of spatial structure, potentially leading to more independent validation sets. However, the trade-off is that within each block, the level of spatial autocorrelation might remain high, especially in the presence of strong spatial trends. This dual effect could explain why larger blocks don't always improve the accuracy of the SAC-0.1 method as expected.

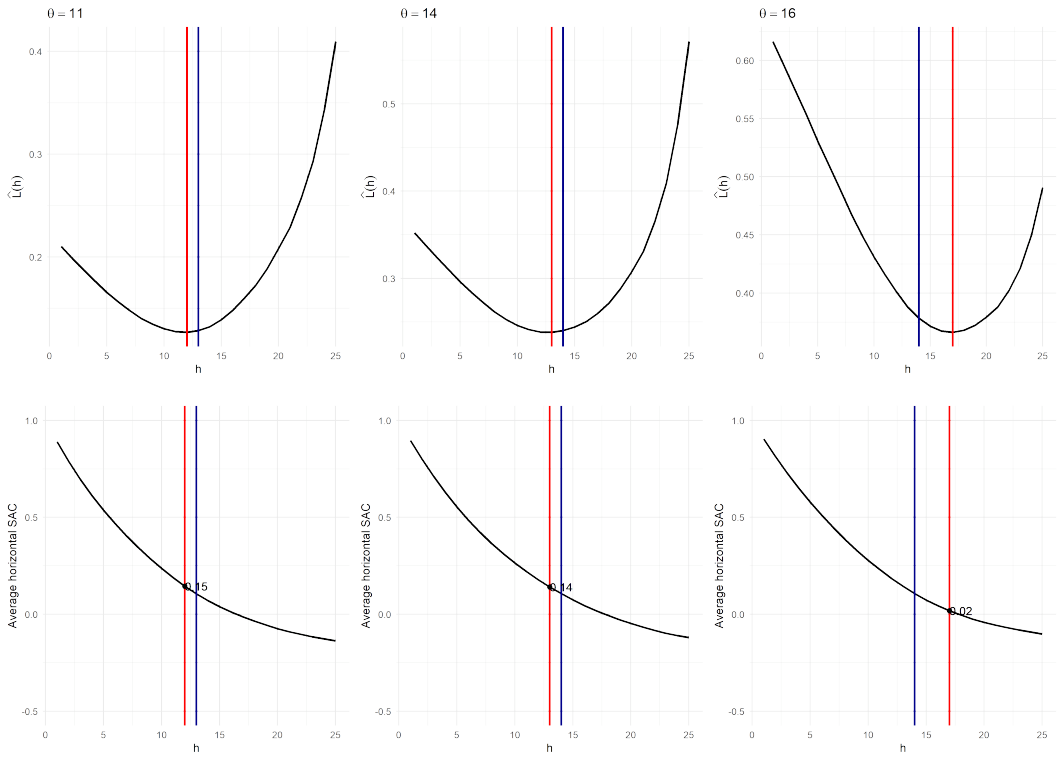


Figure 2.14: Plots of $\hat{\mathcal{L}}(h)$ (first row) for SBCV with block size 8×12 and spatial autocorrelation (second row) for $\theta = 11,\ 14$, and 16 in the scenario of a non stationary trend. The optimal buffer h_I is indicated by a vertical red line, and $h_{0.1}$ is located by the vertical blue line.

θ	h_I	$\hat{\mathcal{L}}(h_I)$	$h_{0.1}$	$\hat{\mathcal{L}}(h_{0.1})$	$\hat{\mathcal{L}}(h_{0.1}) - \hat{\mathcal{L}}(h_I)$	$\frac{(\hat{\mathcal{L}}(h_{0.1}) - \hat{\mathcal{L}}(h_I))}{\hat{\mathcal{L}}(h_I)}$
0.5	3	0.000379	1	0.000380	0.000001	0.34%
1	1	0.000840	2	0.000849	0.000009	1.06%
2	1	0.003392	5	0.003630	0.000238	7.01%
3	2	0.005698	6	0.006048	0.000350	6.15%
4	2	0.013860	8	0.016032	0.002172	15.67%
5	4	0.020938	9	0.024478	0.003540	16.91%
6	6	0.039989	10	0.046202	0.006213	15.54%
7	8	0.041876	11	0.043940	0.002064	4.93%
8	6	0.062344	12	0.081893	0.019549	31.36%
9	10	0.094601	12	0.096592	0.001990	2.10%
10	10	0.123113	13	0.133280	0.010167	8.25%
11	12	0.126674	13	0.128289	0.001614	1.27%
12	10	0.183207	13	0.190192	0.006985	3.8%
13	11	0.168838	14	0.186129	0.017291	10.24%
14	13	0.238217	14	0.240096	0.001878	0.79%
15	13	0.381345	14	0.384325	0.002980	0.78%
16	17	0.366283	14	0.378485	0.012202	3.33%
17	17	0.354030	15	0.358773	0.004743	1.34%
18	14	0.361346	15	0.364844	0.003498	0.97%
19	15	0.528556	15	0.528556	0	0%
20	14	0.478480	15	0.481252	0.002772	0.58%

Table 2.6: Buffer size and comparison of their associated SBCV risks in the scenario of a non stationary trend, for block size of 8×12

.

Summary of the results

We conducted a simulation study under six different spatial cross-validation scenarios, SLOO and SBCV, with or without a non-stationary trend, and with two different SBCV block sizes. The simulations were performed on a 80×60 lattice ($n = 4800$) across a range of spatial dependence strengths, $\theta \in \{1, \ldots, 20\}$.

- **Main finding.** In all scenarios, the ideal buffer size h_I, which minimizes the spatial CV risk $\hat{\mathcal{L}}(h)$, is close to the lag $h_{0.1}$ at which the empirical spatial autocorrelation function $\hat{\rho}(\cdot)$ reaches 0.1. This relationship largely holds across *all* values of θ.
- **Practical implication.** The fact that $h_I \approx h_{0.1}$ provides a straightforward *heuristic* for choosing the buffer size in both SLOO and SBCV methods. Across most spatial dependence levels and trend assumptions, using $h_{0.1}$ yields a spatial CV risk that is close to the optimal risk obtained at h_I.
- **Consistency across trend scenarios.** The heuristic remains valid both when the underlying process has:
  1. A *constant* trend ($\mathbf{Y} = \boldsymbol{\varepsilon}$).
  2. A *non-stationary* linear trend ($Y(\mathbf{s}) = \mathbf{X}(\mathbf{s})\boldsymbol{\beta} + \varepsilon(\mathbf{s})$).

  In both cases, $\hat{\rho}(\cdot) \approx 0.1$ is a strong indicator of the buffer size that minimizes the cross-validation discrepancy.
- **Block sizes in SBCV.** Two block sizes were investigated in SBCV 4×6 and 8×12. For weaker to moderate spatial dependence, the 0.1 heuristic consistently identifies a buffer size whose CV risk is close to that of the true optimum h_I. For stronger dependence or larger block sizes, we observe higher deviations between $\hat{\mathcal{L}}(h_I)$ and $\hat{\mathcal{L}}(h_{0.1})$. Even then, the *relative risk difference* remains modest for most θ values.
- **Limitations.**
  1. When θ is large, $h_{0.1}$ can *underestimate* h_I, sometimes by a notable margin. However, the absolute risk difference often remains small.
  2. Using large blocks in SBCV or imposing large buffer sizes can remove many training points, potentially affecting model fitting.

Overall, the study demonstrates that setting the buffer size to the lag at which the empirical spatial autocorrelation function reaches 0.1 offers a practical and robust guideline for spatial cross-validation. This recommendation holds for both SLOO and SBCV approaches and remains largely effective under varying degrees of spatial dependence.

2.5 Comparison of the SAC-0.1 method with other approaches

We presented two existing methods, range-variogram and Moran's index methods, and our new approach. We are going to compare the results of these 3 procedures.

Figure 2.15 compares the three methods (including the two variants of the Moran's Index method) for determining the buffer size by presenting boxplots of the estimated values over 100 simulations for $\theta = 5$ and $\theta = 15$. The horizontal red line indicates the ideal buffer size h_I. Of the three methods, the Moran's Index approach proposed by Karasiak yields slightly larger estimates of h and is more accurate when spatial dependence is high.In contrast, the range-variogram method introduced by Le Rest tends to produce the smallest estimates, appearing more accurate when spatial dependence is mild. Our SAC-0.1 method appears to be between the two others.

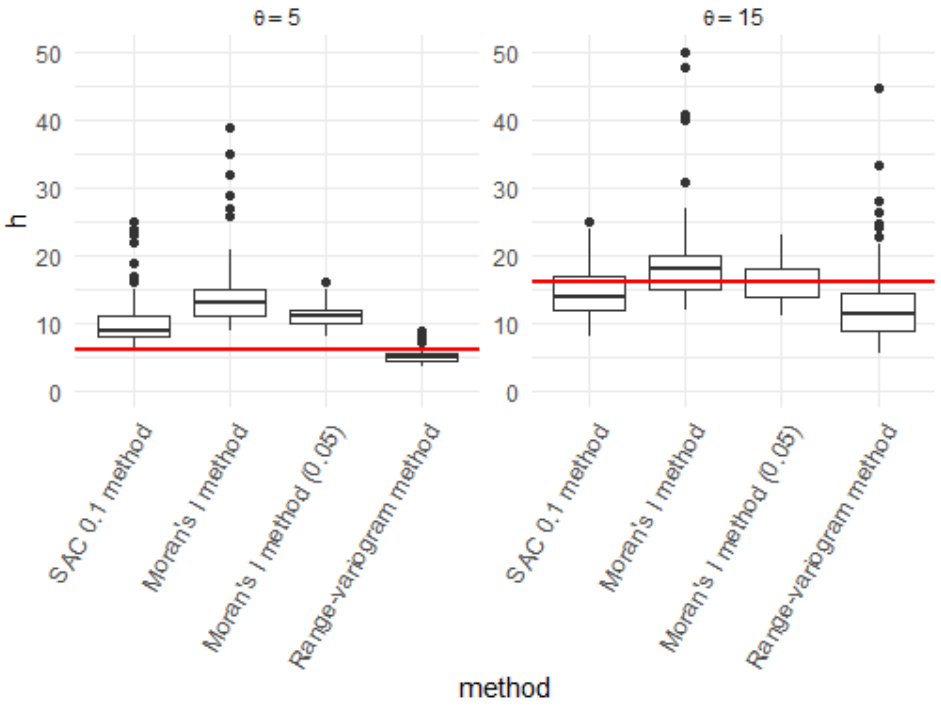


Figure 2.15: Boxplots of buffer radius selected using the different estimation methods. The horizontal red line represents h_I.

In the setting of low spatial dependence, all three methods produce estimates with relatively low variance, with the range-variogram approach exhibiting the smallest variance among the three. However, variance increases for both methods as θ grows, reflecting greater difficulty in selecting an optimal buffer size when spatial dependence is pronounced. Overall, these results suggest that the Moran's Index method may be the more conservative

of the three, while the range-variogram approach can underestimate h when dependence is strong. They also underscore the challenges of managing higher spatial dependence in buffer size selection.

Now, we will compare these methods in terms of the discrepancy function defined in Equation (2.2.1), which measures how close the spatial cross-validation risk with buffer, computed by each method, is to the ideal risk.

To compare our results with the variogram-based method, we need to fit the variogram of the residuals. We have chosen to do this under the most favorable scenario, i.e., fitting the variogram under the true model using an exponential variogram model. We considered adjustment ranges from 1 to 25. Accordingly, our method measured spatial autocorrelation for lag distances ranging from 1 to 25.

We computed the SLOO and SBCV estimates of the risk for each simulation l using the buffer sizes $h^{(l)}_{\text{range}}$, $h^{(l)}_{\text{Moran}}$, $h^{(l)}_{\text{Moran}_{0.05}}$ and $h^{(l)}_{0.1}$. Formally, for SLOO we used equation (2.1.2) and for SBCV equation (2.1.3), yielding $\mathcal{R}^{(l)}_{n,h^{(l)}_{\text{range}}}$, $\mathcal{R}^{(l)}_{n,h^{(l)}_{\text{Moran}}}$, $\mathcal{R}^{(l)}_{n,h^{(l)}_{\text{Moran}_{0.05}}}$ and $\mathcal{R}^{(l)}_{n,h^{(l)}_{0.1}}$. We also computed the ideal risk $\mathcal{R}^{I(l)}_n$. Each of these quantities was then used to evaluate the mean squared difference between the estimated risk and $\mathcal{R}^{I(l)}_n$, averaged over the 100 simulations.

Tables 2.7, 2.8, and 2.9 summarize these results. They report the discrepancies $\hat{\mathcal{L}}(h_{\text{range}})$, $\hat{\mathcal{L}}(h_{\text{Moran}})$, $\hat{\mathcal{L}}(h_{\text{Moran}_{0.05}})$ and $\hat{\mathcal{L}}(h_{0.1})$, where $\hat{\mathcal{L}}(h_{\text{range}})$ is defined analogously to $\hat{\mathcal{L}}(h_{0.1})$ by,

$$\hat{\mathcal{L}}(h_{range}) = \frac{1}{M}\sum_{l=1}^{M}(\mathcal{R}^{(l)}_{n,h^{(l)}_{range}} - \mathcal{R}^{I(l)}_n)^2. \tag{2.5.1}$$

Analogous definitions hold for $\hat{\mathcal{L}}(h_{\text{Moran}})$ and $\hat{\mathcal{L}}(h_{\text{Moran}_{0.05}})$.

2.5.1 SLOO Case

Table 2.7 presents the values of $\hat{\mathcal{L}}(h_{range})$, $\hat{\mathcal{L}}(h_{\text{Moran}})$, $\hat{\mathcal{L}}(h_{\text{Moran}_{0.05}})$, and $\hat{\mathcal{L}}(h_{0.1})$ for SLOO across various values of θ.

All methods indicate that the mean square differences generally increase with θ, reflecting a growing challenge in estimating spatial effects as spatial dependence intensifies. At lower values of θ (e.g., $\theta \leq 3$), all methods perform similarly, with minimal differences in the discrepancy values. This suggests that under conditions of low spatial variability, the choice of method has limited impact on performance.

However, as θ increases, differences become more pronounced. From $\theta \geq 11$, the range-variogram method consistently exhibits the largest discrepancies $\hat{\mathcal{L}}(h_{range})$, emphasizing

θ	$\hat{\mathcal{L}}(h_{0.1})$	$\hat{\mathcal{L}}(h_{range})$	$\hat{\mathcal{L}}(h_{\text{Moran}})$	$\hat{\mathcal{L}}(h_{\text{Moran}_{0.05}})$
0.5	0.0004	0.0004	0.0004	0.0004
1	0.0009	0.0008	0.0008	0.0008
2	0.0039	0.0035	0.0037	0.0036
3	0.0060	0.0066	0.0065	0.0062
4	0.0177	0.0157	0.0190	0.0162
5	0.0252	0.0246	0.0316	0.0249
6	0.0473	0.0502	0.0661	0.0549
7	0.0494	0.0557	0.0536	0.0480
8	0.0933	0.0921	0.1242	0.1156
9	0.1224	0.1303	0.1139	0.1046
10	0.1432	0.1622	0.1741	0.1562
11	0.1589	0.1727	0.1560	0.1442
12	0.1963	0.3345	0.2339	0.2269
13	0.2233	0.2584	0.2216	0.1961
14	0.2443	0.3676	0.2915	0.3070
15	0.4421	0.5841	0.4300	0.4228
16	0.4588	0.5232	0.4088	0.4258
17	0.4571	0.5174	0.4110	0.4049
18	0.4016	0.5094	0.3992	0.3894
19	0.6121	0.7019	0.5880	0.5880
20	0.5256	0.6774	0.5354	0.5354

Table 2.7: Comparison of the SAC-0.1 method with the range-variogram method used in SLOO.

its sensitivity to higher levels of spatial variability. This behavior is particularly evident for $\theta \geq 15$, where discrepancies increase sharply for all methods, with the range-variogram method showing the steepest growth.

The Moran-based metrics ($\hat{\mathcal{L}}(h_{\text{Moran}})$ and $\hat{\mathcal{L}}(h_{\text{Moran}_{0.05}})$) show a more moderate increase compared to $\hat{\mathcal{L}}(h_{range})$ at higher θ values. However, they exhibit significant jumps at specific thresholds, such as at $\theta = 12$ and $\theta = 15$, indicating a non-linear response to increasing spatial complexity. At higher θ values (e.g., $\theta \geq 17$), $\hat{\mathcal{L}}(h_{\text{Moran}})$ appears to plateau, suggesting an upper limit to its sensitivity.

The metric $\hat{\mathcal{L}}(h_{0.1})$ generally increases with θ but shows a less volatile trajectory compared to $\hat{\mathcal{L}}(h_{range})$. This could indicate that $\hat{\mathcal{L}}(h_{0.1})$ offers a more stable approximation in the presence of extreme spatial dependence.

Finally, there are irregular trends for some metrics, such as a drop in $\hat{\mathcal{L}}(h_{\text{Moran}})$ at

$\theta = 7$, followed by an increase at $\theta = 8$ and beyond. This warrants further investigation as it could reflect a transition point in the spatial structure's impact on the methods. These observations highlight the trade-offs between methods, with the range-variogram method capturing extreme spatial variability but potentially overestimating discrepancies, while the Moran-based and SAC-0.1 methods offer more balanced performance under high spatial dependence.

2.5.2 SBCV Case

Tables 2.8 and 2.9 present the comparison of our SAC-0.1 method with the range-variogram method and Moran's Index methods for SBCV with block sizes of 4×6 and 8×12, respectively.

θ	$\hat{\mathcal{L}}(h_{0.1})$	$\hat{\mathcal{L}}(h_{range})$	$\hat{\mathcal{L}}(h_{\text{Moran}})$	$\hat{\mathcal{L}}(h_{\text{Moran}_{0.05}})$
0.5	0.0004	0.0004	0.0004	0.0004
1	0.0009	0.0008	0.0009	0.0009
2	0.0043	0.0034	0.0037	0.0039
3	0.0063	0.0061	0.0066	0.0072
4	0.0188	0.0148	0.0176	0.0220
5	0.0271	0.0233	0.0282	0.0378
6	0.0518	0.0469	0.0651	0.0827
7	0.0519	0.0504	0.0505	0.0618
8	0.1138	0.0988	0.1491	0.1625
9	0.1298	0.1223	0.1092	0.1309
10	0.1595	0.1611	0.1798	0.2106
11	0.1665	0.1604	0.1521	0.1823
12	0.1990	0.3735	0.2521	0.2766
13	0.2563	0.2802	0.2203	0.2780
14	0.2409	0.3929	0.3343	0.3326
15	0.4558	0.6333	0.4370	0.4708
16	0.4434	0.5110	0.4006	0.4110
17	0.4523	0.5190	0.3838	0.4252
18	0.4003	0.5532	0.3881	0.4324
19	0.6191	0.7568	0.6301	0.6243
20	0.5311	0.7681	0.5421	0.5891

Table 2.8: Comparison of the SAC-0.1 method with the range-variogram method and the Moran's Index used in SBCV with block size 4×6.

For the SBCV method with a block size of 4×6, the results show that at lower θ values (e.g., $\theta \leq 3$), all methods perform comparably, with minimal differences in the discrepancy function values. As θ increases, the SAC-0.1 method generally demonstrates lower discrepancy values compared to the range-variogram and Moran's Index methods. However, there are specific instances, such as at $\theta = 7$ and $\theta = 9$, where Moran's Index achieves lower discrepancies than the SAC-0.1 method, indicating its potential for competitive performance under certain conditions. At higher θ values (e.g., $\theta = 16$), the SAC-0.1 method consistently provides the most reliable risk estimates, outperforming the other approaches.

θ	$\hat{\mathcal{L}}(h_{0.1})$	$\hat{\mathcal{L}}(h_{range})$	$\hat{\mathcal{L}}(h_{\text{Moran}})$	$\hat{\mathcal{L}}(h_{\text{Moran}_{0.05}})$
0.5	0.0004	0.0004	0.0004	0.0004
1	0.0009	0.0008	0.0009	0.0009
2	0.0048	0.0035	0.0038	0.0041
3	0.0070	0.0060	0.0072	0.0084
4	0.0211	0.0151	0.0199	0.0269
5	0.0314	0.0237	0.0343	0.0482
6	0.0649	0.0484	0.0858	0.1115
7	0.0597	0.0483	0.0570	0.0773
8	0.1552	0.1206	0.2064	0.2309
9	0.1501	0.1221	0.1260	0.1711
10	0.1976	0.1755	0.2274	0.2789
11	0.1931	0.1602	0.1810	0.2395
12	0.2265	0.4607	0.3111	0.3718
13	0.3326	0.3393	0.2895	0.4007
14	0.2626	0.4979	0.4312	0.4677
15	0.5095	0.7465	0.4877	0.5601
16	0.4625	0.5374	0.3927	0.4598
17	0.4790	0.5561	0.3839	0.4921
18	0.4346	0.6639	0.4168	0.5256
19	0.6704	0.8854	0.7050	0.7203
20	0.5867	0.9743	0.6213	0.7209

Table 2.9: Comparison of the SAC-0.1 method with the range-variogram method and the Moran's Index used in SBCV with block size 8×12.

For the larger block size (8×12), the differences between the methods become more pronounced. The larger block size amplifies discrepancies for the range-variogram method, particularly at higher θ values, whereas the SAC-0.1 method maintains lower discrepancy

values across all θ levels. Interestingly, Moran's Index occasionally outperforms the SAC-0.1 method, as seen at $\theta = 7$, $\theta = 9$, and $\theta = 13$. Despite these instances, the SAC-0.1 method remains more robust overall, particularly in scenarios with strong spatial autocorrelation or high θ values.

Moran's Index appears to perform well in certain intermediate θ ranges but becomes less effective as θ increases further, particularly for larger block sizes. This behavior highlights its sensitivity to spatial dependence and block size, suggesting that while it can be competitive in specific cases, it may not be as reliable under stronger spatial dependencies or larger blocks.

Overall, the comparison underscores the strengths and limitations of each method. While all methods perform similarly under weak spatial dependencies, the SAC-0.1 method provides the consistent and robust result. Nevertheless, Moran's Index demonstrates competitive performance in certain scenarios, suggesting it could be a viable alternative, specially in intermediate spatial dependence settings.

Conclusion

Each of these methods estimates the buffer size h by examining the dependence structure in the model's residuals. The variogram-based method proposed by Le Rest et al. identifies the distance at which spatial dependence effectively disappears by presupposing a particular variogram form. In our simulations, we used the correct variogram model, and under this assumption, the method performs quite well at recovering the effective range. However, additional simulation experiments, where the variogram model was not specified correctly, reveal a marked increase in variability for both model selection and the resulting h_{range}. As a result, while the variogram-based approach can be effective under certain conditions, the results for both SLOO and SBCV suggest that it may still yield higher discrepancy values, especially at larger spatial dependencies or block sizes. This underlines the method's sensitivity to both the assumed variogram model and the scale of spatial dependence.

The method by Karasiak et al. leverages Moran's I to determine where autocorrelation ceases to be significant, offering a statistically rigorous criterion. In the SLOO setting, $\text{Moran}_{0.05}$ often performs competitively, sometimes outperforming other methods at intermediate levels of spatial dependence. However, its tendency to overestimate h in certain settings can lead to increased discrepancies under stronger spatial dependencies or larger block sizes.

Our SAC-0.1 method, by contrast, provides a straightforward heuristic that bypasses model fitting or statistical testing, instead relying on a fixed threshold to define a distance of negligible autocorrelation. The results for both SLOO and SBCV demonstrate its robust-

ness: in SLOO, the 0.1 threshold and $\text{Moran}_{0.05}$ yield comparably low discrepancies, while under SBCV, the SAC-0.1 method consistently maintains, or improves upon, this performance across a range of spatial dependencies and block sizes. This reliability underscores its utility as a practical and effective alternative, particularly in settings characterized by strong spatial dependencies or large block sizes.

Although these approaches are relatively simple to implement, they remain "naive" in the sense that they do not directly target the buffer size that minimizes the discrepancy between the cross-validation risk and the ideal risk. A more advanced strategy would aim to replicate or approximate h_I, the buffer size that aligns the spatial cross-validation risk most closely with the ideal risk. The next chapter addresses this challenge by developing a procedure to approximate this optimal buffer size, thereby moving beyond purely heuristic or model-based methods.

Chapter 3

Refined methods for determining the optimal buffer size

3.1 Sample spatial leave-one-out (SSLOO)

In this chapter, we propose a more sophisticated procedure that uses a parametric bootstrap to approximate the ideal empirical risk when only a single realization of $\mathbf{Y}$ is available. The core idea is to simulate multiple fields that resemble the original data in both mean structure and covariance, effectively mimicking the scenario of having M independent realizations. Specifically, we generate B bootstrap replicates by resampling from the regression residuals under a fitted spatial covariance model, thereby preserving spatial autocorrelation. This approach builds upon established resampling techniques for time series (Lahiri, 2003, Chapter 2) and for spatial data (Cressie, 2015, Chapter 7).

Although the parametric bootstrap procedure described above is conceptually straightforward, it can become computationally expensive, particularly for spatial leave-one-out. Generating the B bootstrap copies is relatively fast, but repeatedly computing both the bootstrapped ideal empirical risk $\mathcal{R}_n^{I\star(b)}$ and the bootstrapped spatial cross-validation empirical risk $\mathcal{R}_{n,h}^{\star(b)}$ (to be defined later) for each replicate can be time-consuming. Consequently, it is important to develop practical acceleration strategies that reduce the overall runtime without sacrificing too much accuracy.

A first way to speed up the search for the optimal buffer involves restricting the range of candidate buffer sizes to lags where the spatial autocorrelation is close to 0.1. This idea stems from the "SAC-0.1" heuristic proposed in Chapter 2, which pinpointed a meaningful distance at which autocorrelation becomes modest. By focusing on a narrower interval around that distance, we reduce the number of buffer sizes that need to be tested.

Another approach to lessen computational demands, specifically for SLOO, is to subsample the data. Instead of leaving out one location at a time across all n points, we use only a fraction p of the available locations. Below, we investigate how this "sample SLOO" strategy affects the estimation of the optimal buffer size. In Sample Spatial Leave-One-Out, that we denote SSLOO from now on, we randomly select a subset of the observation locations of size np, for some proportion $p \in (0, 1]$. For each realization $\mathbf{Y}^{(l)}$, let $\mathcal{S}_{np}^{(l)}$ denote the subset of sites chosen *at random*, noting that this selection differs across realizations l. We then compute a "sample" predictive empirical risk,

$$\mathcal{R}_{np,h}^{(l)} \;=\; \frac{1}{np} \sum_{\mathbf{s}_m \,\in\, \mathcal{S}_{np}^{(l)}} \Big(Y^{(l)}(\mathbf{s}_m) \;-\; \mathbf{X}(\mathbf{s}_m)\, \hat{\boldsymbol{\beta}}_h^{t_m(l)} \Big)^2, \tag{3.1.1}$$

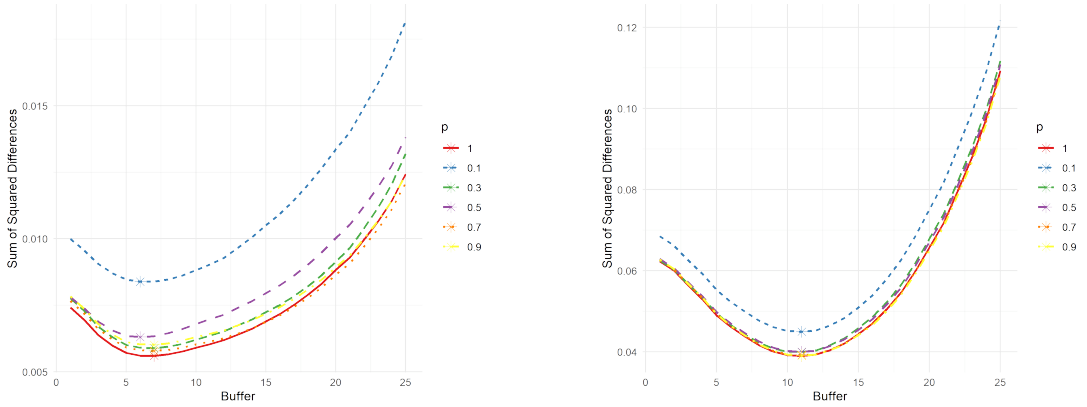
where $\hat{\boldsymbol{\beta}}_h^{t_m(l)}$ is the SLOO estimator that excludes the buffer around the location $\mathbf{s}_m$. This sample empirical risk is then compared to the ideal empirical risk $\mathcal{R}_n^{I(l)}$ (computed over $\mathcal{S}_n$) via the discrepancy measure

$$\hat{\mathcal{L}}(h, p) \;=\; \frac{1}{M} \sum_{l=1}^{M} \big(\mathcal{R}_{np,h}^{(l)} \;-\; \mathcal{R}_n^{I(l)} \big)^2. \tag{3.1.2}$$

Note that the ideal empirical risk is still evaluated on the full set of n locations, whereas the sample empirical risk uses only the subset $\mathcal{S}_{np}^{(l)}$. Moreover, $\mathcal{S}_{np}^{(l)}$ is generated anew for each simulation replicate l.

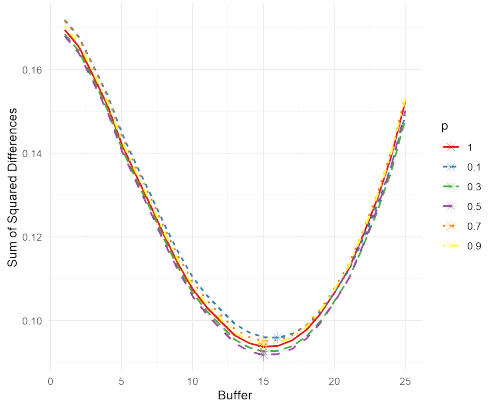
Figure 3.1 presents the behavior of $\hat{\mathcal{L}}(h, p)$ for various proportions $p \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1\}$ and under different strengths of spatial dependence, characterized by $\theta \in \{3, 6, 9, 12, 15, 20\}$. In each subplot, the curves corresponding to the different values of p are depicted, with the case $p = 1$ serving as the benchmark.

Overall, the minima across most of these curves coincide with or lie very close to each other; we can observe some larger discrepancies for small values of p, which is quite understandable. Importantly, running SSLOO on about half the locations ($p = 0.5$) seems to preserve a good approximation to the optimal buffer size while greatly reducing computational cost. Hence, a practical choice is to subsample 50% of the data for SLOO in cases where runtime is a critical concern.

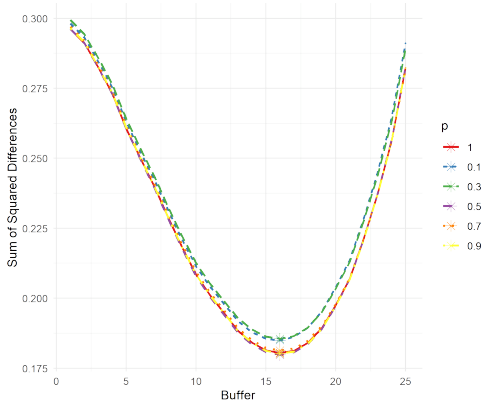


(a) $\theta = 3$

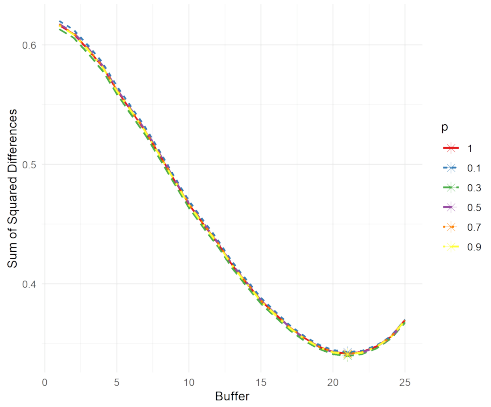
(b) $\theta = 6$



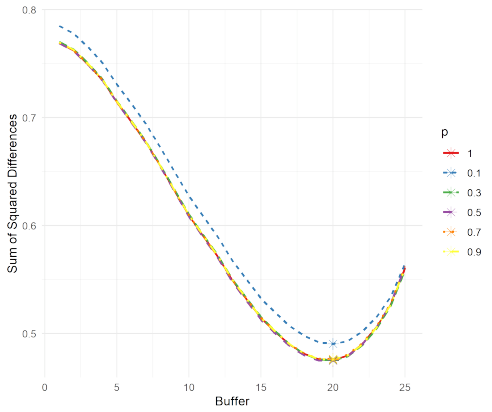
(c) $\theta = 9$



(d) $\theta = 12$



(e) $\theta = 15$



(f) $\theta = 20$

Figure 3.1: Plots of $\hat{\mathcal{L}}(h, p)$ for medium to strong spatial dependence ($\theta = 3, 6, 9, 12, 15, 20$). Each curve corresponds to a different subset proportion p. The refence curve is obtained by using all data points, with $p = 1$.

3.2 Parametric Bootstrap

Our proposed parametric bootstrap method involves the following steps:

1. Fit the linear model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ to obtain

$$\hat{\boldsymbol{\beta}} \;=\; (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\,\mathbf{X}^{\mathrm{T}}\,\mathbf{Y},$$

   which leads to fitted values $\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ and residuals $\mathbf{e} = \mathbf{Y} - \hat{\mathbf{Y}}$.

2. Estimate the sample variogram of the residuals $\mathbf{e}$ and fit a parametric covariance function to capture the spatial dependence.

3. For $b = 1$ to B:

   (a) Simulate a new set of residuals, $\boldsymbol{\varepsilon}^{\star(b)}$, by drawing from the fitted covariance model.

   (b) Construct the bootstrap response

$$\mathbf{Y}^{\star(b)} = \mathbf{X}\,\hat{\boldsymbol{\beta}} + \boldsymbol{\varepsilon}^{\star(b)}.$$

Although we conceptually replace the M hypothetical independent realizations with B bootstrap copies $\{\mathbf{Y}^{\star(b)}\}_{b=1}^{B}$, it is common practice to set $B > M$. Generating additional bootstrap replicates reduces Monte Carlo variability in the estimates of both the "ideal" risk and the optimal buffer size, while providing a richer sample of the parametric model's uncertainty. The bootstrap's flexibility allows us to artificially create as many replicates as needed to approximate the underlying distribution with greater precision. By drawing a sufficiently large B, we can mitigate the random fluctuations inherent in any finite sample and obtain more stable estimates.

Bootstrap ideal and spatial CV empirical risks

For each bootstrap copy $\mathbf{Y}^{\star(b)}$, we define,

$$\mathcal{R}_n^{I\star(b)} \;=\; \frac{1}{B-1}\sum_{\substack{j=1\\ j\neq b}}^{B}\frac{1}{n}\sum_{\mathbf{s}\in\mathcal{S}_n}\Big(\hat{Y}^{\star(b)}(\mathbf{s}) \;-\; Y^{\star(j)}(\mathbf{s})\Big)^2, \tag{3.2.1}$$

where $\hat{Y}^{\star(b)}(\mathbf{s}) = \mathbf{X}(\mathbf{s})\,\hat{\boldsymbol{\beta}}^{(b)}$ and $\hat{\boldsymbol{\beta}}^{(b)}$ is estimated by regressing $\mathbf{Y}^{\star(b)}$ on $\mathbf{X}$.

We also compute a bootstrap analogue of the SLOO empirical risk,

$$\mathcal{R}_{n,h}^{\star(b)} \;=\; \frac{1}{n}\sum_{m=1}^{n}\Big(Y^{\star(b)}(\mathbf{s}_m) \;-\; \mathbf{X}(\mathbf{s}_m)\,\hat{\boldsymbol{\beta}}_h^{t_m(b)}\Big)^2, \tag{3.2.2}$$

where $\hat{\boldsymbol{\beta}}_h^{t_m(b)}$ denotes the OLS estimator obtained by excluding the buffer of radius h around the held-out location $\mathbf{s}_m$. **Identifying the optimal bootstrap buffer**

We finally choose $h_I^\star$ by minimizing

$$\hat{\mathcal{L}}^\star(h) \;=\; \frac{1}{B}\sum_{b=1}^{B}\Big(\mathcal{R}_{n,h}^{\star(b)} - \mathcal{R}_n^{I\star(b)}\Big)^2. \tag{3.2.3}$$

Thus, $\hat{\mathcal{L}}^\star(h)$ measures how far the spatial CV empirical risk at buffer h deviates from the ideal empirical risk across all bootstrap samples. The minimizing $h_I^\star$ serves as our best approximation of the true optimal buffer when only one realization of $\mathbf{Y}$ is available.

By explicitly modeling the covariance structure and bootstrapping residuals, this parametric approach should capture the spatial dependence. However, it strongly depends on the way and under which model the residuals variogram is fitted. Obviously, the bootstrap copies of $\mathbf{Y}$ are significantly influenced by this adjustment. We emphasize this crucial point by conducting a brief study on variogram fitting.

Variogram estimation

To obtain the spatial covariance model from the residuals $\mathbf{e}$, we first compute an empirical variogram, then fit a theoretical variogram model using the `fit.variogram` function in the `gstat` package of `R`. In practice, this process involves two main steps:

1. **Compute the empirical variogram.**

   We partition the pairwise distances between locations into bins (or "lags") and calculate the average semivariance for each bin. The semivariance at lag h is defined as

   $$\hat{\gamma}(h) \;=\; \frac{1}{2\,|\mathcal{N}(h)|}\sum_{\{i,j\}\,\in\,\mathcal{N}(h)}\big(e(\mathbf{s}_i) - e(\mathbf{s}_j)\big)^2,$$

   where $\mathcal{N}(h)$ represents the set of all pairs of locations separated by an approximate distance of h.

2. **Fit a theoretical variogram model**

   We used the `fit.variogram` function with an exponential model, matching the true exponential form used to generate the data, to fit the theoretical variogram to the empirical variogram points. By default, this procedure employs a weighted nonlinear least squares approach, thereby estimating the sill and range parameters in a manner consistent with the known underlying spatial process.

   $$\min_{\boldsymbol{\phi}}\sum_k w_k\,\Big(\hat{\gamma}(h_k) - \gamma(h_k\,;\,\boldsymbol{\phi})\Big)^2,$$

where $\boldsymbol{\phi}$ are the parameters of the model (e.g., nugget, sill, and range), $\hat{\gamma}(h_k)$ is the empirical semivariance at lag h_k, and $\gamma(h_k\,;\,\boldsymbol{\phi})$ is the theoretical model value at that same lag.

Once the variogram is fitted, nugget, partial sill, and range are estimated. These parameters define the covariance model used in the parametric bootstrap, ensuring that the simulated residual fields $\boldsymbol{\varepsilon}^{\star(b)}$ accurately capture the spatial dependence observed in the original data.

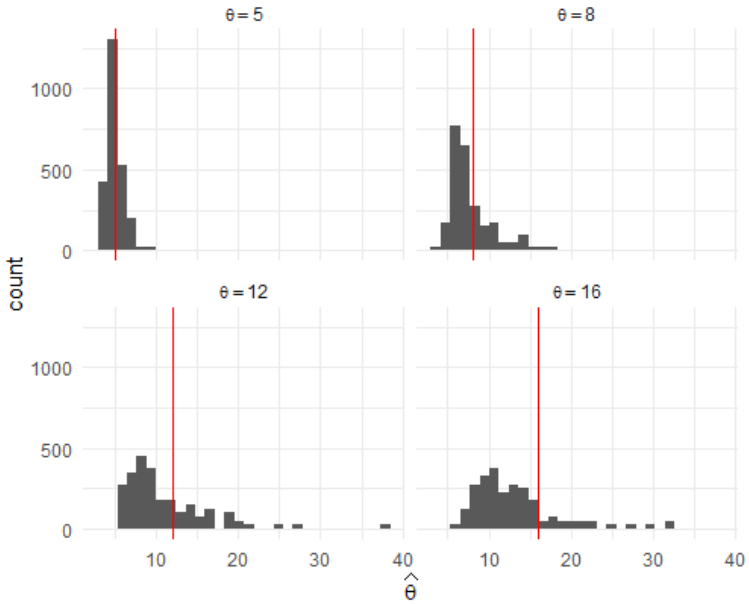


Figure 3.2: Distribution of $\hat{\theta}$ values obtained using the `fit.variogram` function for true θ values of 5, 8, 12, and 16. The red vertical line indicates the true θ value.

Figure 3.2 presents histograms of the estimated values $\hat{\theta}$ of θ from variogram models fitted to residuals under varying levels of spatial dependence. The results indicate that as the intensity of spatial dependence increases, the accuracy of the range estimates declines. This is reflected in larger deviations from the true θ values and greater variance in the estimates.

A major consequence is that the bootstrap copies may exhibit either weaker or stronger spatial dependence compared to the original realization. Moreover, this point will be highlighted when the variogram is fitted using an incorrect model.

3.3 Ideal bootstrap buffer size

3.3.1 Sample spatial leave-one-out with bootstrap

We now examine how to determine the optimal buffer size using a parametric bootstrap applied to only a proportion p of the available observations. For each bootstrap copy $\mathbf{Y}^{\star(b)}$, we randomly draw np points from the original n locations and compute two quantities,

$$\mathcal{R}_n^{I\star(b)} \;=\; \frac{1}{B-1}\sum_{\substack{j=1\\ j\neq b}}^{B}\;\frac{1}{n}\sum_{\mathbf{s}\in S_n}\Big(\hat{Y}^{\star(b)}(\mathbf{s}) \;-\; Y^{\star(j)}(\mathbf{s})\Big)^2, \tag{3.3.1}$$

and

$$\mathcal{R}_{np,h}^{\star(b)} \;=\; \frac{1}{np}\sum_{\mathbf{s}_m\,\in\, S_{np}^{(b)}}\Big(Y^{\star(b)}(\mathbf{s}_m) \;-\; \mathbf{X}(\mathbf{s}_m)\,\hat{\boldsymbol{\beta}}_h^{t_m(b)}\Big)^2. \tag{3.3.2}$$

In (3.3.1), $\mathcal{R}_n^{I\star(b)}$ approximates the "ideal" empirical risk by comparing the fitted field $\hat{\mathbf{Y}}^{\star(b)}$ with each of the other bootstrap samples $\mathbf{Y}^{\star(j)}$. In (3.3.2), $\mathcal{R}_{np,h}^{\star(b)}$ is a "sample" SLOO empirical risk computed only over the subset $S_{np}^{(b)} \subseteq S_n$. Note that if $p = 1$ we obtain the bootstrap empirical risk for SLOO defined in Equation (3.2.2). We then find

$$\hat{\mathcal{L}}^{\star}(h,p) \;=\; \frac{1}{B}\sum_{b=1}^{B}\Big(\mathcal{R}_{np,h}^{\star(b)} \;-\; \mathcal{R}_n^{I\star(b)}\Big)^2. \tag{3.3.3}$$

Let us note $h_{I,p}^{\star}$ that minimizes $\hat{\mathcal{L}}^{\star}(h,p)$,

$$h_{I,p}^{\star} = \underset{h\in\mathcal{H}}{\operatorname{argmin}}\,\hat{\mathcal{L}}^{\star}(h,p) \tag{3.3.4}$$

Restricting the search interval

Because strong spatial dependence often leads to larger true optimal buffer sizes h_I, we restrict our search to the nonsymmetric interval $[\,h_{0.1}-3,\; h_{0.1}+5\,]$. This interval is chosen to account for the tendency of the SAC-0.1 heuristic to underestimate the optimal buffer size in highly autocorrelated settings. By extending the range further above $h_{0.1}$ than below, we ensure that the search space adequately captures the potential upward bias in h_I caused by strong spatial dependence, while still allowing for some flexibility in cases where the heuristic might slightly overestimate.

Simulation design and computational setup

Due to time constraints, we selected at random 10 realizations $\mathbf{Y}$ from the original pool of $M = 100$ simulated fields to illustrate the procedure. We ran separate experiments for both the constant (null) trend and the non-stationary trend scenarios, fixing $p = 0.5$ and $B = 499$. All computations were performed on an Intel® Core™ i7-8850H CPU (2.60

GHz, 6 cores, 32 GB RAM), using `R`'s `foreach` function with 12 clusters to parallelize the SLOO empirical risk calculations.

Running times. The unmodified algorithm run in all n onservations required about 34.86 hours. Restricting the search interval for h yielded a drop to 13.86 hours, and further adopting the subsampling approach reduced runtime to 5.45 hours, an overall 83% decrease.

Results

Figures 3.3 and 3.4 display the distribution of the estimated $h^{\star}_{I,p}$ across 10 replicates for $\theta \in \{5, 8, 12, 16\}$. Vertical red lines indicate the true h_I as defined in Equation (2.2.2). In the constant-trend case (Figure 3.3), the estimates cluster tightly around h_I for $\theta = 5$ and $\theta = 8$, signifying high accuracy. As θ increases to 12 or 16, the estimates exhibit more spread but still cover the true value.

Similar behavior emerges under the non-stationary trend in Figure 3.4,Variability increases with stronger dependence, yet the true value remains within the estimated range. We see that for $\theta = 16$ the algorithm fails to reach h_I due to restricting the search interval to $[\, h_{0.1} - 3,\ h_{0.1} + 5\,]$. Nevertheless, in both scenarios, the bootstrap estimates stick around h_I, lending further confidence to the robustness of this methodology. Consistent with earlier findings, even mild deviations of h from h_I produce similar empirical risks $\mathcal{R}_{n,h}$, reinforcing the notion that precise estimation of h is helpful but not exceedingly sensitive in practice.

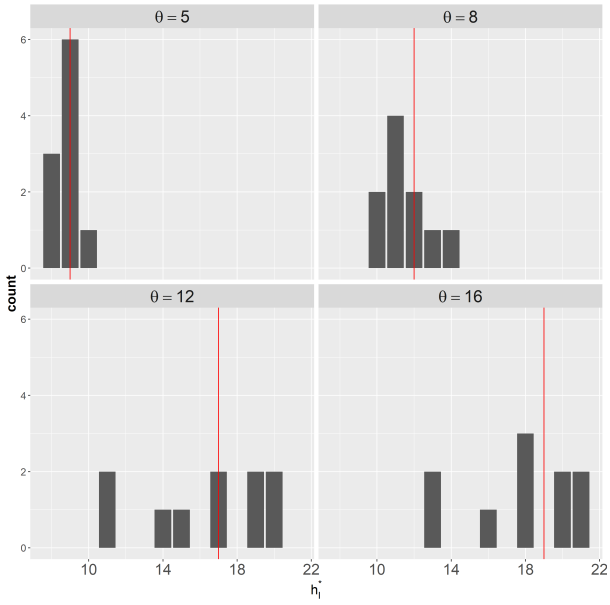


Figure 3.3: Counts of parametric bootstrap estimates of $h^{\star}_{I,p}$ under a constant trend, for $\theta \in \{5, 8, 12, 16\}$. Each bar corresponds to a single replicate, and the red vertical line marks the true h_I.

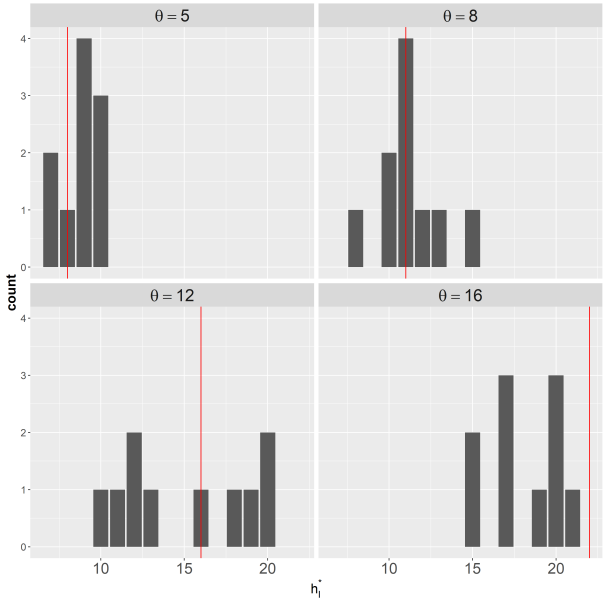


Figure 3.4: Counts of parametric bootstrap estimates of $h^{\star}_{I,p}$ under a non-stationary trend, for $\theta \in \{5, 8, 12, 16\}$. The red vertical line indicates h_I.

3.3.2 Spatial blocked cross-validation with bootstrap

We perform the algorithm described in Section 3.2, computing a bootstrap analogue of the SBCV empirical risk. Meaning no sampling of the available data is done to reduce

computation time. In particular, for each bootstrap replicate $\mathbf{Y}^{\star(b)}$, the SBCV estimate becomes

$$\mathcal{R}_{n,h}^{\star(b)} = \frac{1}{K}\sum_{m=1}^{K}\frac{1}{|\mathrm{v}_m|}\sum_{\mathbf{s}\in\mathrm{v}_m}\left(Y^{\star(b)}(\mathbf{s}) - \mathbf{X}(\mathbf{s})\,\hat{\boldsymbol{\beta}}_h^{\mathrm{t}_m(b)}\right)^2, \tag{3.3.5}$$

where K is the number of spatial blocks, v_m is the set of locations comprising block m, and $\hat{\boldsymbol{\beta}}_h^{\mathrm{t}_m(b)}$ denotes the OLS estimator computed on the training set after excluding a buffer of size h around the v_m locations.

As described earlier, we limit the buffer search to the interval $[h_{0.1}-3,\ h_{0.1}+5]$. Due to time constraints, we selected at random 10 realizations $\mathbf{Y}$ from the original set of $M = 100$ simulated fields to illustrate the procedure. Additionally, we focused on the non-stationary trend scenario only.

Results

Figures 3.5 and 3.6 summarize how the bootstrap procedure estimates the optimal buffer size h_I under a non-stationary trend for various levels of spatial dependence θ. Each barplot depicts the distribution of bootstrap-based estimates $h_I^\star$ for 10 replicates; the true optimal value h_I is shown by the vertical red line. We compare two different SBCV block configurations: a smaller 4×6 block layout (Figure 3.5) and a larger 8×12 layout (Figure 3.6). We use $B = 499$ bootstrap replicates and 10 different realizations of $\mathbf{Y}$, each with a non-stationary trend. Results are shown for $\theta \in \{5, 8, 12, 16\}$

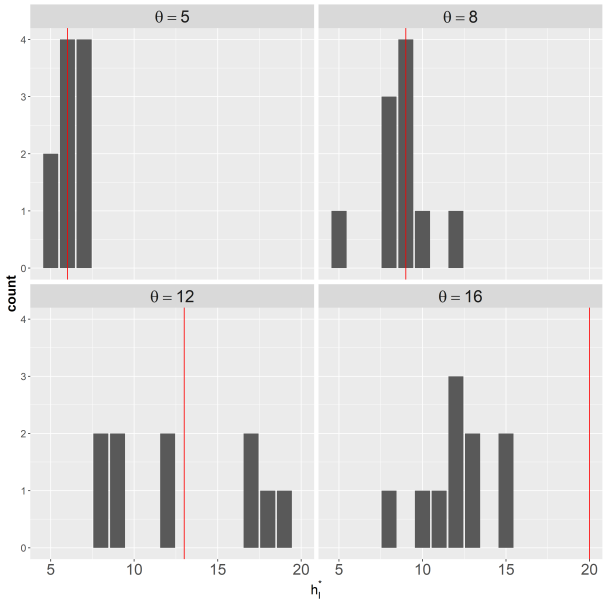


Figure 3.5: Counts of parametric bootstrap estimates $h_I^\star$ in SBCV with a 4×6 block design for $\theta \in \{5, 8, 12, 16\}$. The red line indicates the true h_I.

The results differ depending on the block size.

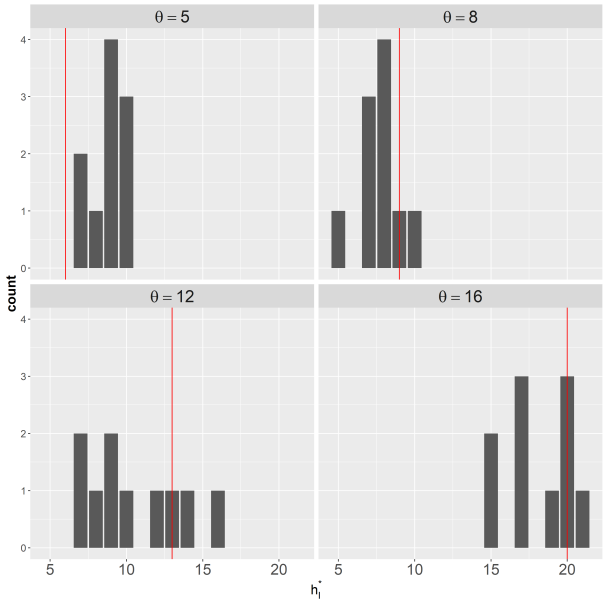


Figure 3.6: Counts of parametric bootstrap estimates $h_I^\star$ in SBCV with a 8×12 block design for $\theta \in \{5, 8, 12, 16\}$. The red line indicates the true h_I.

Block size 4×6

Figure 3.5 illustrates how the estimates of $h_I^\star$ evolve with increasing spatial dependence θ.

- $\theta = 5$: The estimates cluster tightly around the true h_I, suggesting strong accuracy in weaker spatial dependence.
- $\theta = 8$: Slightly more variation appears, yet most estimates remain close to the true value.
- $\theta = 12$: The spread grows wider, indicating that although the estimates still center near h_I, there is increasing dispersion.
- $\theta = 16$: The estimates become more widely scattered, frequently overshooting h_I. This pattern suggests that with a smaller block size, it becomes harder for SBCV to account for the strong spatial autocorrelation, reducing accuracy and precision.

Overall, the results show that as θ increases, the bootstrap estimates display higher variability and tend to deviate more from h_I. Adjustments or additional refinements may be needed in high-dependence scenarios.

Block size 8×12

Figure 3.6 highlights the performance of a larger block layout.

- $\theta = 5$: The bootstrap estimates overestimate h_I with low variation suggesting diminished accuracy in weaker spatial dependence for bigger block size.
- $\theta = 8$: The bootstrap estimates again lie tightly around h_I with low variation, mirroring the strong performance observed in the smaller θ range of the 4×6 design.
- $\theta = 12$: Although the estimates exhibit more spread than at lower θ, the overall dispersion remains narrower than in the 4×6 case, suggesting a benefit from larger blocks.
- $\theta = 16$: Even under strong spatial correlation, the bootstrap estimates concentrate closer to h_I compared to the 4×6 layout. The 8×12 design appears to better accommodate high dependence, improving both accuracy and consistency of $h_I^\star$.

Summary

Overall, the parametric bootstrap procedure coupled with partial sampling ($p = 0.5$) appears to reliably approximates the ideal buffer size h_I for SLOO under both constant and non-stationary trends. Restricting the search interval for h and adopting subsampling together reduced computational times by about 83%, demonstrating a significant practical advantage. Although estimates of $h_{I,p}^\star$ become more variable at higher spatial dependence ($\theta = 12$ or 16), they still center around the true h_I. Furthermore, consistent with prior findings, the empirical risks $\mathcal{R}_{n,h}$ remain relatively stable even when h deviates mildly from h_I.

Besides, our findings illustrate the sensitivity of SBCV to block size in the presence of strong spatial dependence. Smaller blocks (4×6) capture the optimal h_I well when θ is moderate but become less reliable as autocorrelation intensifies. Larger blocks (8×12) improve accuracy and precision for higher-dependence regimes, underscoring the importance of selecting an appropriate block size for SBCV in strongly autocorrelated data.

This conclusions should be considered with caution, as they are based on the study of only 10 replications of the process $\mathbf{Y}$.

3.4 Application

In this section we test our proposed cross-validation strategies using the synthetic species-abundance dataset of Roberts et al. (2017). Widely adopted as a benchmark in spatial CV studies, see Mahoney et al. (2023a), this dataset enables us to illustrate how both the heuristic SAC-0.1 method method and our parametric bootstrap approach perform under realistic constraints, including partially misspecified models.

Data generation

In Roberts et al. (2017), both the explanatory variables and the response are generated on a 50×50 lattice. This setup emulates species-abundance data driven by spatially autocorrelated environmental factors and mechanisms such as disease prevalence or finite resource availability.

Four core variables form the base of the original simulation:

- $\mathbf{X}_1$: an environmental factor (e.g., soil or topography), with exponential covariance (var $= 0.1$, scale $= 0.1$);
- $\mathbf{X}_2$: precipitation, also exponential (var $= 0.3$, scale $= 0.1$);
- $\mathbf{X}_3$: temperature, with Gaussian covariance (var $= 0.1$, scale $= 0.3$);
- $\mathbf{X}_4$: disease prevalence ($p_d = 0.05$).

They are used to derive several additional covariates. In particular,

$$\mathbf{X}_5 = \mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3 + (\mathbf{X}_2 \times \mathbf{X}_3), \quad \mathbf{X}_{11} = \frac{\mathbf{X}_2}{\mathbf{X}_3 + 273},$$

$$\mathbf{X}_{12} = \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{\mathbf{X}_3^2}{4}\Big), \quad \mathbf{X}_{13} = \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{\mathbf{X}_2^2}{4}\Big).$$

The species-abundance field $\mathbf{Y}$ is modeled as

$$\mathbf{Y} = \mathbf{X}_1 + \mathbf{X}_5 + \mathbf{X}_{12} + \mathbf{X}_{13},$$

subject to two resource-based constraints:

$$\mathbf{Y} = \min(\mathbf{Y}, \mathbf{X}_{11}) \quad \text{if } \mathbf{Y} > \mathbf{X}_{11}, \qquad \mathbf{Y} = \min(\mathbf{Y}) \quad \text{if } \mathbf{X}_4 = 1.$$

An example of the resulting simulated landscape is shown in Figure 3.7.

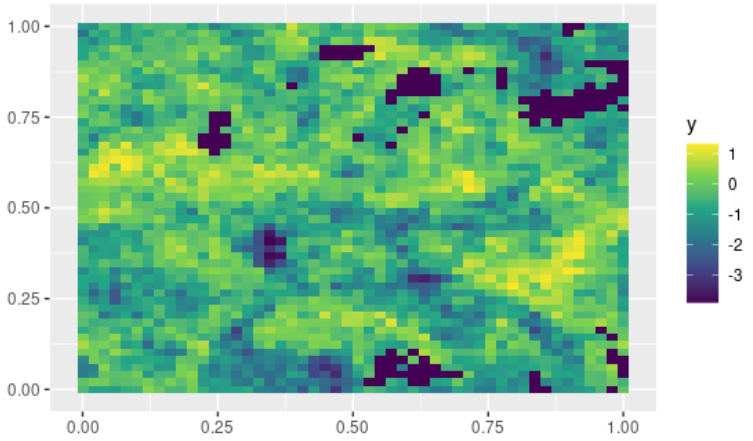


Figure 3.7: Example of a simulated 50×50 landscape from the Roberts et al. (2017) species-abundance model.

Fitted model and experimental design

We generated $M = 100$ realizations of $\mathbf{Y}$ and, for each, fit a linear model

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

where $\mathbf{X} = [\mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_6, \mathbf{X}_7, \mathbf{X}_8, \mathbf{X}_9, \mathbf{X}_{10}]$. Here, $\mathbf{X}_6$, $\mathbf{X}_7$, and $\mathbf{X}_8$ follow exponential covariance ($\sigma^2 = 0.1$, $\theta = 0.1$), while $\mathbf{X}_9$ and $\mathbf{X}_{10}$ follow Gaussian covariance ($\sigma^2 = 0.1$, $\theta = 0.3$). None of these covariates precisely matches the original data-generating mechanism, thus representing a partially misspecified model. This design tests how well our buffer-size selection methods adapt to realistic conditions where covariates or interactions may be omitted or approximated.

SLOO and SBCV using the SAC-0.1 method method

We first apply the SAC-0.1 method rule to both SLOO and SBCV methods. Figures 3.8 and 3.9 show representative plots of $\hat{\mathcal{L}}(h)$ (left panels) alongside the empirical spatial autocorrelation (right panels). We used 5×5 blocks to keep block sizes consistent with earlier experiments on an 80×60 lattice, which used 4×6 blocks.

In each figure, the red line denotes the true optimal buffer h_I, whereas the blue line indicates $h_{0.1}$. Despite the model mismatch, the SAC-0.1 heuristic consistently approximates h_I in both SLOO and SBCV. As expected, SBCV slightly inflates the estimated risk under block-based sampling, but the overall discrepancy remains modest.

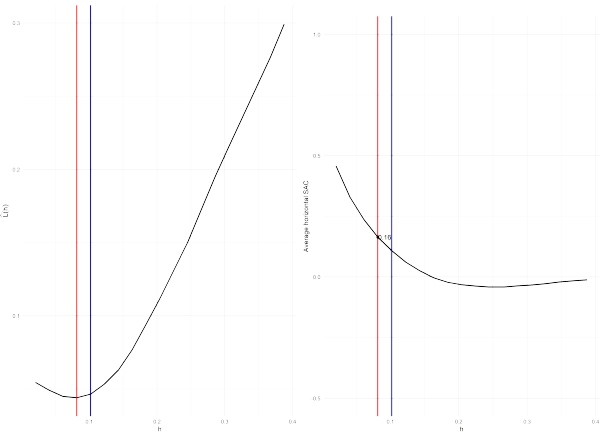


Figure 3.8: Plots of $\hat{\mathcal{L}}(h)$ for SLOO (left) and the corresponding spatial autocorrelation function (right). The red vertical line is the true h_I, while the blue line indicates $h_{0.1}$.

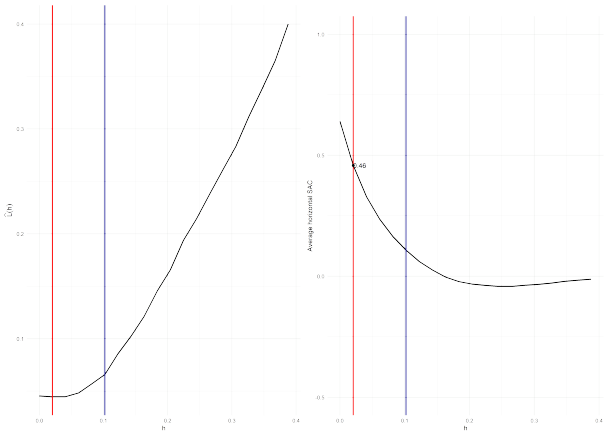


Figure 3.9: Plots of $\hat{\mathcal{L}}(h)$ for SBCV with a 5×5 block layout (left) and the corresponding spatial autocorrelation (right). The red vertical line marks h_I; the blue line marks $h_{0.1}$.

Bootstrap-based estimates

To gain further insight into buffer-size estimation, we adopt our parametric bootstrap procedure. Figures 3.10 and 3.11 depict results for 10 distinct realizations of $\mathbf{Y}$, using $B = 499$ bootstrap replicates. For SLOO, we only sample 50% of the data in each replicate (SSLOO), while SBCV again uses 5×5 blocks and all n locations. Each bar in the figures corresponds to a single realization; the red vertical line shows the true h_I.

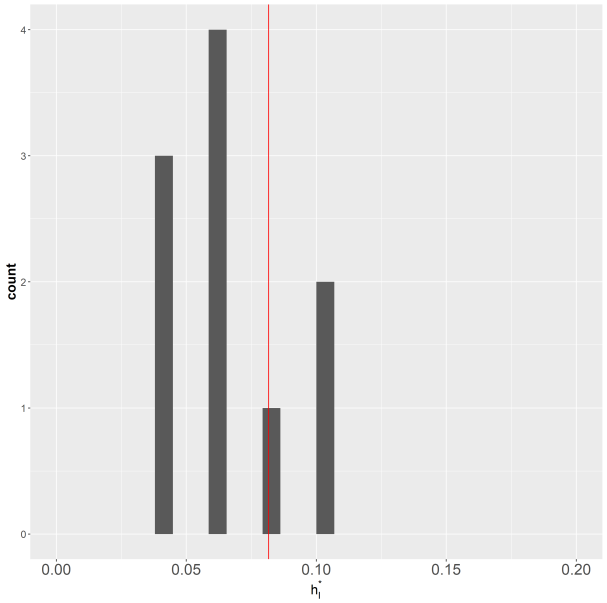


Figure 3.10: Bootstrap-based estimates of $h^{\star}_{I,p}$ with $p = 0.5$ obtained after SSLOO procedure. The red line indicates the true h_I.

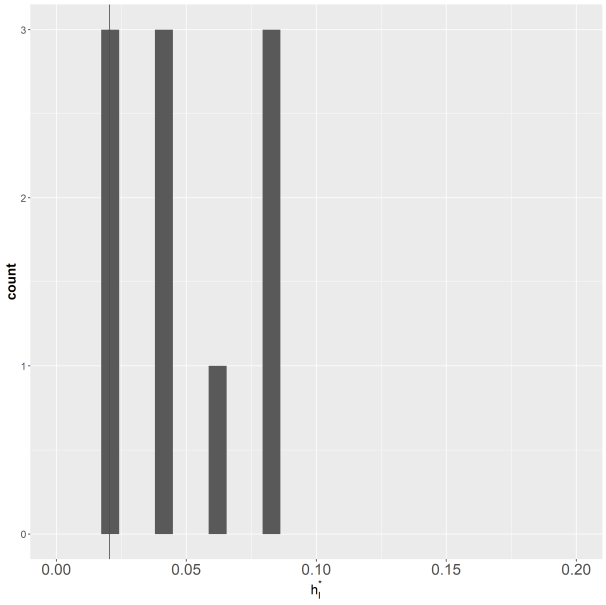


Figure 3.11: Bootstrap-based estimates $h_I^\star$ for SBCV obtained after SBCV procedure. The red line represents the true h_I.

Overall, in both SLOO and SBCV modes, the bootstrap estimates cluster around the true buffer h_I, exhibiting only mild variability. Even under partial misspecification, the estimated h values remain close to the target. Minor deviations of h from h_I do not drastically affect the cross-validation risk, aligning with our earlier observations that CV performance is relatively robust to small buffer-size errors.

Summary

This application to the Roberts et al. (2017) dataset demonstrates the effectiveness of our methods under a complex, ecologically motivated spatial process. The SAC-0.1 method heuristic offers a straightforward yet accurate way to approximate h_I, whereas the more involved parametric bootstrap approach provides a refined view of buffer-size uncertainty. Even with a partially misspecified model, both approaches identify buffer sizes that yield reliable risk estimates. These results underscore the potential of our proposed strategies for real-world ecological and environmental analyses, where the true underlying processes are often only partially known or observed.

3.5 Conclusions

This chapter introduced a parametric bootstrap framework for determining an appropriate buffer size in spatial cross-validation when only one realization of a spatial process is

available. By fitting a spatial covariance model to regression residuals and generating multiple bootstrap replicates, we approximated the "ideal" predictive empirical risk that would otherwise require many independent realizations. The buffer size was then chosen to minimize the discrepancy between the bootstrap based empirical risk and the spatial CV empirical risk, either for spatial leave-one-out or spatial blocked cross-validation.

To control the computational cost of SLOO, we introduced a subsampling scheme (SSLOO) that evaluates only a fraction of the locations. Combined with a focused search interval guided by empirical autocorrelation criteria, this reduced runtime by more than 80% in certain simulations without significantly affecting accuracy.

Through extensive simulation experiments, the bootstrap approach consistently approximated the optimal buffer size across varying strengths of spatial autocorrelation. While small deviations from the true optimal buffer size did not appreciably affect the cross-validation empirical risk, limiting the data used for each SLOO step provided a practical means of scaling these methods to larger datasets.

Finally, applying this method to a synthetic species-abundance dataset from Roberts et al. (2017) confirmed its robustness. Even under a partially misspecified linear model, the parametric bootstrap produced estimates of the buffer size near the "ideal" value. Further, we observed that larger SBCV blocks could mitigate strong spatial dependence, underscoring the importance of block configuration in spatial cross-validation.

Chapter 4

Ridge regularization for spatial autoregressive models with multicollinearity issues

This chapter reproduces the published version of the following paper

Chavez-Chong, C. O., Hardouin, C., & Fermin, A. K. (Year). *Ridge regularization for spatial autoregressive models with multicollinearity issues. AStA Advances in Statistical Analysis*, (2024), Pages. DOI: 10.1007/s10182-024-00496-0.

Introduction

The conventional approach for addressing spatial dependence in data involves employing parametric models that explicitly incorporate spatial dependence structures (e.g., Anselin, 1988; Diggle and Ribeiro, 2007). However, in practical applications, these models are often susceptible to specification errors and structural overfitting, which can undermine model evaluation and predictive performance. Consequently, robust methods for model validation, selection, and performance assessment are essential when dealing with spatially dependent data.

In this chapter, we introduce a novel methodology for constructing an explanatory spatial autoregressive model in the presence of multicollinearity. Our approach leverages Ridge regularization to mitigate the adverse effects of collinearity. We propose new estimation algorithms that simultaneously estimate the regression coefficients and the spatial

dependence parameter. A spatial cross-validation procedure is employed to tune the regularization parameter, and variable importance is assessed via permutation tests—since traditional inferential tests are not valid following Ridge regularization. The efficacy of our methodology is demonstrated through extensive numerical experiments on simulated data, and its practical applicability is further illustrated by applying it to a real dataset to evaluate the impact of selected socio-economic variables on COVID-19 intensity in France.

4.1 Introduction

Statistical models intend to represent, understand and interpret the information lying in a data set. Among others, linear regression analysis is a popular tool used to model underlying relationships between the response variable and covariates (or explanatory variables). This method is widely used in practice due to its advantages, namely the simplicity of the model structure and the consequent ease of interpretation of its results, in addition to the good properties of the estimator, which is unbiased and efficient. However, the assumptions of this model are violated in the case of spatial data, particularly the independence of residuals. There are many ways to take into account spatial dependence in linear regression models. Among them, we shall consider the class of spatial autoregressive models which arose in spatial econometrics. Spatial Autoregressive models, also called Simultaneous Autoregressive models, are widely used, see for instance the seminal works of Anselin, LeSage and their co-authors, mainly (Anselin, 1988) and (LeSage, 2008); their success relies on their intuitive writing, similar to ordinary regression models with the addition of a "spatial lag" term. We shall deal in this work with two well-known key models, the mixed spatial lag model and the spatial error model.

A common trait of real data is the presence of multicollinearity, i.e. some covariates are approximately linear combination of others. This phenomenon causes high variability of parameter estimators and biased inference statistics. Also, for model interpretation, the effects of the variables cannot be distinguished and extrapolation is likely to be misleading (Alin, 2010). Regularization techniques for regression are frequently applied to solve the issue. A quite general theory on the properties of the regularization techniques has been developed over the last decades, see for example (Hoerl and Kennard, 1988), (Zou and Hastie, 2005) and (Tibshirani, 1996). The most classical techniques, Ridge and Lasso, constrain the norm of the vector $\boldsymbol{\beta}$ of regression coefficients by adding a regularization term to the function to be optimized. This introduces some bias, but can greatly reduce the variance. The difference between Ridge and Lasso lies in the regularization term; the Ridge imposes a L^2-penalty which is, $\gamma \times \sum_{i=1}^{p} \beta_i^2$ while Lasso considers a L^1-penalty equal to $\gamma \times \sum_{i=1}^{p} |\beta_i|$. In both cases, γ is a regularization hyperparameter to be tuned. Ridge regression always keeps all the covariates in the model. On the contrary, Lasso produces a

parsimonious model by allowing some coefficient estimates to be set to 0. Thus, Lasso is mostly used for variable selection, and in high dimensional framework (with more variables than observations), while Ridge is mostly used to deal with multi-collinearity problems. This is precisely our goal, to estimate a spatial autoregressive model with highly collinear variables, without eliminating some of them.

In other terms, our objective is to build an explanatory model, which is different from building a predictive model. Indeed, when one wants to build a predictive model, the main goal is predictive accuracy. The objective is to obtain good predictions for the outcome, without any further consideration about the significance of the covariates and even their collinearity. In the presence of two or more highly collinear variables, Lasso may select only one of them, which is not good for the interpretation. In an explanatory model objective, we want to identify the variables which are statistically significant. The knowledge of non-significant variables also provides valuable information for the practitioners; statistical non-significance doesn't mean that there's no relationship between variables; its non-significance could be that their association is not big enough to be statistically detected by tests. Moreover, let us note that non-significance of one variable in multiple regression has to be understood as non-significance when this variable is considered with the other covariates. Considering all the aspects, we focus here on Ridge regularization.

There has been a tremendous amount of research about Ridge adaptation, see (McDonald, 2009) for a general review. Ridge regularization has been adapted to the spatial setting; Wheeler and Páez (Wheeler, 2009) consider Ridge adaptation in the framework of geographically weighted regression; more interesting, in our framework, Fan et al. (2017) considered the context of spatial autoregressive models. They propose "Spatially Filtered Ridge Regression", which we describe in section 4.3.2. However, if the authors use a Ridge procedure for computing the coefficients estimates, they estimate the spatial dependence parameter as if there were no collinearity issues. In this work, we propose new estimation algorithms which take into account that matter for all the parameters.

Furthermore, Ridge regression involves the delicate choice of the regularization parameter γ. Indeed, this parameter plays a crucial role since it controls the strength of the regularization. The appropriate choice of this parameter is a difficult problem. A quite general theory on the properties of the various regularization methods and different parameter choice procedures has been developed over the last decades, starting from the seminal work of Tikhonov and Arsenin (1977). Fan et al. (2017) choose γ according to the Ridge trace criterion proposed by Hoerl and Kennard (1970), that is plotting Ridge coefficients versus γ values, and select the minimum value of γ for which the coefficients start to stabilize. Despite being commonly used in applications, this method suffers from being user-dependent, in the sense that the final choice is done by the user according to

what he observes in the plot. Cross-validation is another common re-sampling method used to tune model parameters. But special care needs to be taken in the presence of spatial dependence; indeed, classical cross-validation techniques are no longer suitable since the assumption of independence between the training and test sets is violated, see Roberts et al. (2017). Here, we want to use an adapted spatial cross-validation procedure to select automatically the regularization parameter; we will make use of spatial leave-one-out (SLOO) which is especially well adapted for spatial autoregressive models.

Building an explanatory model, a crucial point is to determine which variables significantly affect the behaviour of the outcome. Unfortunately, one consequence of Ridge regression is that Student tests or classical F-tests are not appropriate any more. Halawa and El Bassiouni (2000) proposed a t-test based on Ridge estimators; however, other authors showed in intensive simulation studies that it may fail depending on the value of the regularization parameter. We propose permutation F-tests and t-tests.

In Section 4.2, we present the two main spatial autoregressive models that we are going to consider. We develop our new estimation procedures in Section 4.3. The crucial step of selecting the regularization parameter is achieved through a coherent spatial leave-one-out procedure. Finally, we propose in Section 4.4 to run permutation F-tests to assess the importance of the explanatory variables.

To validate the performance of our new procedure, we conduct comprehensive simulations whose results are summarised in Section 4.5; we simulate spatial autoregressive models with 8 highly correlated variables, and estimate their parameters using classical spatial and non-spatial methods, the Spatially Filtered Ridge Regression proposed by Fan et al. (2017), and our procedure.

Finally, for purpose of illustration, we conduct a thorough study on a real data set in Section 4.6. We consider the hospitalization rate due to the COVID-19 pandemic in metropolitan France, to be explained by socio-economic covariates. First we resume the data and include an exploratory analysis which highlights the presence of spatial dependence and multicollinearity. Then we run our estimation procedure and finally determine the core variables affecting the epidemic indicator. Our concluding remarks are provided in Section 4.7.

4.2 Spatial autoregressive models

Let us consider a finite set of sites $S = \{\boldsymbol{s}_i, \quad i = 1, \cdots, n\}$ on a spatial domain $D \subset \mathbb{R}^d$. We assume that we observe some data $\mathbf{Y} = (Y(\boldsymbol{s}_1), \cdots, Y(\boldsymbol{s}_n))^T$ on S and p explanatory variables $\mathbf{X}_1, \ldots, \mathbf{X}_p$.

Simultaneous spatial autoregressive models are common in spatial econometrics; they take into account spatial dependence structures by using a neighbourhood graph on S. In this work, we will consider two main models; the first one, that we shall denote SAR, is a mixed regressive-spatial autoregressive model defined by

$$\begin{aligned} \mathbf{Y} &= \rho W\mathbf{Y} + \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} &\sim N(\mathbf{0}, \sigma^2\mathbf{I}_n) \end{aligned} \tag{4.2.1}$$

where $\mathbf{I}_n$ is the identity matrix of order n and $\mathbf{X} = [\mathbf{X}_1 \cdots \mathbf{X}_p]$ is a $n \times p$ matrix with $\mathbf{X}_i = (X_i(\boldsymbol{s}_1), \cdots, X_i(\boldsymbol{s}_n))^T$, $i = 1, \ldots, p$. W is a deterministic spatial weights matrix; though there's no direct counterpart, it is often considered as the equivalent of the backshift operator B for time series, like a spatial lag operator acting as a shift over space. The spatial weights depend on the definition of a neighbourhood set for each observation. We set $w_{ii} = 0$ and $w_{ij} = 0$ if sites $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ are not neighbours. $(WY)_i$ is interpreted as a weighted average of the neighbouring values of $y(\boldsymbol{s}_i)$. Let us note that there are many specifications of the spatial weights, the usual way is to consider geographical distances (at a negative power), but one can consider economic or social distances. The parameter ρ in model (4.2.1) above reflects the strength of the spatial dependence between the elements of $\mathbf{Y}$.

The second widely used model is the Spatial Error Model, denoted by SEM, and defined by:

$$\begin{aligned} \mathbf{Y} &= \mathbf{X}\boldsymbol{\beta} + \mathbf{u}, \\ \mathbf{u} &= \lambda W\mathbf{u} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I}_n) \end{aligned} \tag{4.2.2}$$

In this model $\mathbf{u}$ corresponds to a noise that is spatially correlated and λ characterizes the strength of the dependence.

These two models are well-defined under assumptions relying on the eigenvalues of the spatial weight matrix W that we will not discuss here; see for instance the works of Anselin (1988) or LeSage (2008).

Let us recall that our objective is to obtain explanatory SAR or SEM models under the assumption of multicollinearity between the covariates $\boldsymbol{X}_i$. Thus, the matrix $[\mathbf{X}^T\mathbf{X}]$ has very small eigenvalues, leading to numerical instability of its inverse. The poor behaviour of the estimator of the coefficients $\boldsymbol{\beta}$ in the presence of multicollinearity is inherited from ordinary linear regression by spatial autoregressive models. Furthermore, the estimation of the parameter ρ or λ is also computed indirectly from the matrix $[\mathbf{X}^T\mathbf{X}]$. The Ridge regularization is a way to overcome the problem, bypassing the multicollinearity issue while preventing from variable removal. But it has to be adapted to the spatial framework. We present new algorithms to derive Ridge estimates for SAR and SEM models in the next paragraph.

4.3 Estimation

To exhibit the novelty of our procedure, we start by recalling the usual estimation procedure to estimate SAR and SEM models defined in (4.2.1) and (4.2.2).

4.3.1 Spatial autoregressive models estimation

Let us recall that the least squares estimation of these models produces biased and inconsistent estimates. Therefore, one considers the following procedure, which is described for example in Anselin (1988). The estimation is based on the so-called concentrated likelihood and is achieved in a few steps.

Let us start with the SAR model. The log-likelihood has the following expression,

$$\begin{aligned} l_{SAR}(\mathbf{y} \mid \boldsymbol{\beta}, \sigma^2, \rho) = & -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) + \ln|\mathbf{I}_n - \rho W| \\ & - \frac{1}{2\sigma^2}\big((\mathbf{I}_n - \rho W)\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\big)^T\big((\mathbf{I}_n - \rho W)\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\big) \end{aligned} \tag{4.3.1}$$

We first focus on parameter $\boldsymbol{\beta}$. Its maximum likelihood estimator is

$$\begin{aligned} \hat{\boldsymbol{\beta}}_{ML} = & \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T(\mathbf{I}_n - \rho W)\mathbf{Y} \\ = & \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y} - \rho\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T W\mathbf{Y} \\ = & \hat{\boldsymbol{\beta}}_O - \rho\hat{\boldsymbol{\beta}}_L \end{aligned} \tag{4.3.2}$$

where $\hat{\boldsymbol{\beta}}_O = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y}$ is the ordinary least squares (OLS) estimator of the coefficients of the regression of $\mathbf{Y}$ on $\mathbf{X}$ and $\hat{\boldsymbol{\beta}}_L = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T W\mathbf{Y}$ is the OLS estimator of the coefficients of the regression of the spatial lag ($W\mathbf{Y}$) on $\mathbf{X}$. So, we obtain $\hat{\boldsymbol{\beta}}_{ML}$ as soon as ρ is known. The idea is then to perform OLS on the two regressions, and compute the residuals $e_O = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_O$ and $e_L = W\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_L$. Then, it also can be shown that the ML estimate of σ^2 is given by

$$\widehat{\sigma^2} = \frac{1}{n}(e_O - \rho e_L)^T(e_O - \rho e_L). \tag{4.3.3}$$

Again, we obtain this estimate as soon as ρ is known. Substituting σ^2 and $\boldsymbol{\beta}$ in the log-likelihood (4.3.1) by their expressions in equations (4.3.3) and (4.3.2), we obtain the concentrated log-likelihood function,

$$l^C_{SAR}(\mathbf{y} \mid \rho) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln\{\frac{1}{n}(e_O - \rho e_L)^T(e_O - \rho e_L)\} + \ln|\mathbf{I}_n - \rho W| \tag{4.3.4}$$

We then find $\hat{\rho}$ maximising (4.3.4), which resumes in a one-parameter non-linear optimization problem. The final step consists in plugging $\hat{\rho}$ in (4.3.2) and (4.3.3) to obtain the final estimators.

Let us now turn to the SEM model. The estimation procedure is analogous to the previous one. We write the log-likelihood as,

$$\begin{aligned} l_{SEM}(\mathbf{y} \mid \boldsymbol{\beta}, \sigma^2, \lambda) = & -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) + \ln|\mathbf{I}_n - \lambda W| \\ & - \frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{I}_n - \lambda W)^T(\mathbf{I}_n - \lambda W)(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \end{aligned} \tag{4.3.5}$$

Let us note the filtered variables $\mathbf{X}_\lambda = \mathbf{X} - \lambda W\mathbf{X}$ and $\mathbf{Y}_\lambda = \mathbf{Y} - \lambda W\mathbf{Y}$. Then, for a fixed λ, we get the maximum likelihood estimators $\widehat{\boldsymbol{\beta}_\lambda} = \left(\mathbf{X}_\lambda^T\mathbf{X}_\lambda\right)^{-1}\mathbf{X}_\lambda^T\mathbf{Y}_\lambda$ and $\widehat{\sigma^2_\lambda} = \frac{1}{n}\mathbf{e}_\lambda^T\mathbf{e}_\lambda$ with $\mathbf{e}_\lambda = \mathbf{Y}_\lambda - \mathbf{X}_\lambda\widehat{\boldsymbol{\beta}_\lambda}$. One can notice that these estimators are those obtained from writing again the SEM model (4.2.2) as

$$\mathbf{Y}_\lambda = \mathbf{X}_\lambda^T\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I}_n) \tag{4.3.6}$$

Substituting the expressions of $\widehat{\boldsymbol{\beta}_\lambda}$ and $\widehat{\sigma^2_\lambda}$ in the log-likelihood (4.3.5), we obtain the concentrated log-likelihood,

$$l^C_{SEM}(\mathbf{y} \mid \lambda) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln\{\frac{1}{n}\mathbf{e}_\lambda^T\mathbf{e}_\lambda\} + \ln|\mathbf{I}_n - \lambda W| \tag{4.3.7}$$

which is a non-linear function of λ. Again, optimizing this function gives $\hat{\lambda}$, then, the final estimates $\widehat{\boldsymbol{\beta}_\lambda}$ and $\widehat{\sigma^2_\lambda}$.

In both models, the procedures will suffer from collinearity issues and produce unstable numerical solution of the estimate of $\boldsymbol{\beta}$. Therefore, the residuals derived from $\widehat{\boldsymbol{\beta}}$ inherit from its instability. Finally, $\widehat{\sigma^2}$ as well as the estimates of ρ or λ cannot be trusted. The impact of collinearity is felt all the way to all the parameters. It seems then important to consider all parameters in the new estimation algorithms to release them from the collinearity burden.

4.3.2 Ridge regression for spatial autoregressive models

In ordinary non-spatial regression, the Ridge estimator is defined by

$$\hat{\boldsymbol{\beta}}_R = (\mathbf{X}^T\mathbf{X} + \gamma\mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{Y}, \tag{4.3.8}$$

$\gamma > 0$ being the so-called regularization parameter. Let us note that in all this paragraph, we consider centred variables, since the regularization concerns the coefficients of the variables without the intercept. It is also very common to scale the variables.

Spatially filtered Ridge regression (SFRR) as named by the authors who first proposed it (Fan et al., 2017) integrates Ridge regression and spatial autoregressive models. This method follows three fundamental steps. For the SAR model, [resp. the SEM model],

1. Estimate ρ [resp. λ].

2. Consider the new response vector to be $\mathbf{Y}_{\hat{\rho}} = (\mathbf{I} - \hat{\rho}W)\mathbf{Y}$ [resp. $\mathbf{Y}_{\hat{\lambda}} = (\mathbf{I} - \hat{\lambda}W)\mathbf{Y}$ and $\mathbf{X}_{\hat{\lambda}} = (\mathbf{I} - \hat{\lambda}W)\mathbf{X}$].

3. Select γ following Ridge trace criterion and estimate $\hat{\boldsymbol{\beta}}_R$ in (4.3.8) with the new response vector (and current or new design matrix).

Fan et al. do not raise it, but the first step is, in fact, an issue, since, as we mentioned above, the ML estimator of ρ (or λ) is obtained using residuals, derived on their own from the ill-conditioned matrix $[\mathbf{X}^T\mathbf{X}]$. We bypass the problem in proposing an iterative algorithm which considers both parameters ρ (or λ) and $\boldsymbol{\beta}$. The estimation of σ^2 automatically follows.

The algorithms we propose for estimating parameters of the SAR and SEM models are similar.

Ridge Regularization for SAR models (RRSAR)

1. Initialization. Consider the ordinary linear regression model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ and estimate $\boldsymbol{\beta}$ by $\hat{\boldsymbol{\beta}}_{0,R} = (\mathbf{X}^T\mathbf{X} + \gamma_0\mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{Y}$ defined in (4.3.8) for $\gamma_0 > 0$. Similarly, for $\gamma_L > 0$, compute $\hat{\boldsymbol{\beta}}_{L,R} = (\mathbf{X}^T\mathbf{X} + \gamma_L\mathbf{I}_p)^{-1}\mathbf{X}^TW\mathbf{Y}$ the ridge estimate of $\boldsymbol{\beta}$ in the ordinary regression of $(W\mathbf{Y})$ on $\mathbf{X}$.

2. Compute $\mathbf{e}_0 = \mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{0,R}$ and $\mathbf{e}_L = W\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{L,R}$ and estimate ρ by using the concentrated ML (4.3.4).

3. Consider the filtered $\mathbf{Y}_{\hat{\rho}} = (\mathbf{I} - \hat{\rho}W)\mathbf{Y}$ and compute $\hat{\boldsymbol{\beta}}_R^{SAR} = (\mathbf{X}^T\mathbf{X} + \gamma\mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{Y}_{\hat{\rho}}$ for $\gamma > 0$.

Ridge Regularization for SEM models (RRSEM)

1. Initialization. Consider the ordinary linear regression model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ and estimate $\boldsymbol{\beta}$ by $\hat{\boldsymbol{\beta}}_R$ defined in (4.3.8).

2. Consider $\mathbf{Y}_\lambda = \mathbf{X}_\lambda\hat{\boldsymbol{\beta}}_R + \boldsymbol{\varepsilon}$ and $\mathbf{e}_\lambda = \mathbf{Y}_\lambda - \mathbf{X}_\lambda\hat{\boldsymbol{\beta}}_R$ and estimate λ by using the concentrated ML (4.3.7).

3. Consider the filtered $\mathbf{Y}_{\hat{\lambda}} = (\mathbf{I} - \hat{\lambda}W)\mathbf{Y}$, the filtered matrix $\mathbf{X}_{\hat{\lambda}} = (\mathbf{I} - \hat{\lambda}W)\mathbf{X}$, and compute $\hat{\boldsymbol{\beta}}_R^{SEM} = (\mathbf{X}_{\hat{\lambda}}^T\mathbf{X}_{\hat{\lambda}} + \gamma\mathbf{I}_p)^{-1}\mathbf{X}_{\hat{\lambda}}^T\mathbf{Y}_{\hat{\lambda}}$ for $\gamma > 0$.

4. Consider $\mathbf{Y}_\lambda = \boldsymbol{X}_\lambda\hat{\boldsymbol{\beta}}_R^{SEM} + \boldsymbol{\varepsilon}$ and $\mathbf{e}_\lambda = \mathbf{Y}_\lambda - \mathbf{X}_\lambda\hat{\boldsymbol{\beta}}_R^{SEM}$ and estimate new λ by using the concentrated ML (4.3.7).

5. Repeat step 3 to obtain the final $\hat{\boldsymbol{\beta}}_R^{SEM}$.

Let us note that the RRSAR algorithm follows the original estimation algorithm of SAR models; Ridge estimates replace OLS estimates in the first step, and $\hat{\boldsymbol{\beta}}^{SAR}$ is changed to $\hat{\boldsymbol{\beta}}_R^{SAR}$. Let us point out that the regularization parameters γ in the first step are not identical, we get different values whether considering the dependent variable to be $\mathbf{Y}$ or $W\mathbf{Y}$. This is crucial to get a correct estimation of ρ. Then we cannot just plug in the obtained estimates in (4.3.2) but we need to regularize globally $\boldsymbol{\beta}$ considering the new dependent variable $\mathbf{Y}_{\hat{\rho}}$.

The RRSEM algorithm has two more steps than the RRSAR because the first estimate of λ in step 2 is obtained after an OLS procedure and has to be refined.

In these algorithms, the computation of the Ridge estimates necessitate sub-steps to determine a good regularization parameter γ. This is done by spatial cross-validation. This step is the subject of the next section 4.3.3.

4.3.3 Spatial cross-validation for selecting the Ridge parameter

The parameter γ has a key role in Ridge regularization ; assigning γ to 0 is considering ordinary least squares (OLS); on the contrary, high values of γ increase the penalty term and thus pull the regression coefficients β_i towards zero. Each value leads to different estimates of $\boldsymbol{\beta}$, crushing them more or less towards zero. In their algorithm, Fan et al. (2017) follow the Ridge trace criterion (Hoerl and Kennard, 1970) to choose γ. They plot the coefficients estimates versus γ and look for the value of γ for which the coefficients stabilize. Another common method is selecting the parameter γ that minimizes the mean square error estimated using cross-validation techniques.

Spatial data, and more accurately, spatial auto-correlation, challenges classical cross-validation techniques. In this situation, random splitting of the data into training and testing sets does no longer simulates the original structure of the data and therefore the key assumption of independent data samples behind cross-validation is violated. There exist some extensions of classical cross-validation to deal with spatial dependence. Detailed works and applications have been conducted on spatial leave-one-out (SLOO) (Le Rest et al., 2013), spatial k-fold cross-validation (Pohjankukka et al., 2017) and blocked cross-validation (Brenning, 2012). The majority of these modifications correspond to the idea of achieving independence between the training set and the test set by the means of "point separation". This is done by deleting points from the training set within a distance h of the test set. The area of deleted points is called "buffer" or dead zone. The selection criterion for h is still under debate. In the case of spatial autoregressive models, SLOO rises naturally because the determination of h is beneficially replaced by taking out the first order neighbours designed by W.

Spatial leave-one-out follows the usual leave-one-out procedure; at the $m-$th iteration of SLOO, one observation becomes the validation set $\mathcal{V}_m = \{(y(\boldsymbol{s}_m), \boldsymbol{x}(\boldsymbol{s}_m))\}$, as in the classical method. Then, we define the buffer around $\boldsymbol{s}_m$ as the set of its neighbours $\partial \boldsymbol{s}_m = \{\boldsymbol{s}_i \in S : \ w_{mi} \neq 0\}$. The training set is then $\mathcal{T}_m = \{(y(\boldsymbol{s}_j), \boldsymbol{x}(\boldsymbol{s}_j)) : w_{mj} = 0\}$. If needed, one can easily extend the buffer to take out the k-th order neighbouring observations, k being chosen for instance following a Moran test (Moran, 1950) based on the k-th order neighbours.

We describe now the SLOO algorithm that we use to select the regularization parameter γ. Let us consider, for example, the SAR model; the procedure is similar for the SEM model. Our criterion is based on the maximisation of the likelihood rather than MSE; indeed, MSE is mostly used for prediction purposes, while we are interested in an explicative model.

For a fixed γ, we run the following procedure. For $m = 1$ to n, we compute $\hat{\boldsymbol{\beta}}^{SAR}_{R,m}(\gamma) = (\mathbf{X}_m^T\mathbf{X}_m + \gamma \mathbf{I}_p)^{-1}\mathbf{X}_m^T\mathbf{Y}_{\hat{\rho},m}$; the subscript m states for the calculations are made on the training set $\mathcal{T}_m$. Note that $\hat{\rho}$ has been obtained from computations on the whole data set at the previous step of the estimation algorithm and is fixed at this point. The same applies for $\hat{\sigma}^2$.

Next we compute the conditional log-likelihood on the validation set, conditionally to the neighbouring values,

$$l_m(\gamma) = -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln(\hat{\sigma}^2) - \frac{1}{2\hat{\sigma}^2}\left(y(\boldsymbol{s}_m) - \hat{\rho}\sum_{\boldsymbol{s}_j \in \partial \boldsymbol{s}_m} w_{mj-} y(\boldsymbol{s}_j) - \boldsymbol{x}(\boldsymbol{s}_m)^T \hat{\boldsymbol{\beta}}^{SAR}_{R,m}(\gamma)\right)^2.$$

Finally, the regularization parameter $\hat{\gamma}$ selected in the estimation algorithm is defined by

$$\hat{\gamma} := \underset{\gamma \in \{\gamma_k\}_{k=1}^K}{\arg\max} \frac{1}{n}\sum_{m=1}^{n} l_m(\gamma). \tag{4.3.9}$$

where $\{\gamma_k\}_{k=1}^K$ is a sequence of possible values of γ; this sequence is created using the path-wise coordinate descend method (Friedman et al., 2010); it first selects γ_{max}, the smallest value that draws $\hat{\boldsymbol{\beta}}$ to $\mathbf{0}$; then the sequence of K values decreases on the log-scale from γ_{max} to $\gamma_{min} = c * \gamma_{max}$. The most common choices are $K = 100$ and $c = 0.001$.

4.4 Importance of the explanatory variables

When building an explanatory model, one major point is to determine how important is the influence of the explanatory variables on the dependent variable $\boldsymbol{Y}$. For spatial autoregressive models, the interpretation of coefficients is not classical, it is made in terms

of "impacts" rather than exploiting coefficients values. Indeed, if we consider a SAR model, for example, the presence of the spatially lagged dependent variable entails, as a logical consequence, that a change in an explanatory variable at a single location can affect several values of $\boldsymbol{Y}$ in other sites. These impacts depend directly on ρ and the components of $\boldsymbol{\beta}$, see LeSage (2008). Furthermore, in the context of Ridge regression, the coefficients suffer a steep fall due to the regularization; then, classic tests are not valid any more.

Halawa and El Bassiouni (2000) propose to use a statistics that imitates the classic Student ratio to test individual regression coefficients in the case of ordinary regression (without spatial dependence).

Let us recall that the bias and variance of the Ridge estimator of the regression coefficients $\boldsymbol{\beta}$ are defined by respectively,

$$\begin{aligned} bias(\hat{\boldsymbol{\beta}}_R) &= E[\hat{\boldsymbol{\beta}}_R] - \boldsymbol{\beta} = -\gamma \left(\mathbf{X}^T\mathbf{X} + \gamma\mathbf{I}\right)^{-1} \boldsymbol{\beta} \\ Var(\hat{\boldsymbol{\beta}}_R) &= \sigma^2 \left(\mathbf{X}^T\mathbf{X} + \gamma\mathbf{I}\right)^{-1} \mathbf{X}^T\mathbf{X} \left(\mathbf{X}^T\mathbf{X} + \gamma\mathbf{I}\right)^{-1} \end{aligned} \tag{4.4.1}$$

To test the null hypothesis $(H_0) : \beta_{R,j} = 0$, Halawa and El Bassiouni (2000) propose to consider $T_j = \dfrac{\hat{\beta}_{R,j}}{S(\hat{\beta}_{R,j})}$, where $\hat{\beta}_{R,j}$ is the j-th element of $\hat{\boldsymbol{\beta}}_R$ and $S(\hat{\beta}_{R,j})$ is the square root of the j-th diagonal element of $Var(\hat{\boldsymbol{\beta}}_R)$. They state that this statistics follows approximately a Student's t distribution with $n-p$ degrees of freedom. In practice, σ^2 in (4.4.1) is replaced by its estimation based on the residuals. However, Perez-Melo and Kibria (2020) show that the test behaves differently according to the regularization parameter; in his thesis Bécu (2016) also states that the test fails especially for large values of γ.

Permutation tests offer a suitable alternative to determine statistical significance. They are based on the assumption that under the hypothesis of no relationship between the dependent variable $\boldsymbol{Y}$ and an explanatory variable X_j, the observations are exchangeable and the joint probability distribution of the permuted samples coincides with the joint probability distribution of the original sample. There are numerous works proposing permutation tests for OLS with the permutation done to the response (Manly, 2006), or the residuals (Kennedy, 1995; Anderson and Robinson, 2001), or the predictor (Hastie and Tibshirani, 1995). Thus, Bécu et al. (2017) propose to replace the classic F-test by a permutation F-test, which is asymptotically exact when the tested variable is independent of the other explanatory variables, and is approximate in the general case. Though the assumption of independence is violated, this test might perform well in the case of not so high correlations.

The permutation F-test is defined as follows. First, let us note $\mathcal{M}_0$ the small model (without the variable to be tested) nested in $\mathcal{M}_1$ the larger model (including all the variables), $\hat{\boldsymbol{Y}}_0$ and $\hat{\boldsymbol{Y}}_1$ the respective predictors of $\mathbf{Y}$ estimated in these models. The classic

F-statistic is defined by

$$F = \frac{\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}_0\|^2 - \|\boldsymbol{Y} - \hat{\boldsymbol{Y}}_1\|^2}{\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}_1\|^2/(n-p)} \tag{4.4.2}$$

Now, let us permute B times the n values of X_j, and let us note $X_{j(b)}$ the b-th permutation. We consider the estimator of the parameters of the spatial autoregressive model with the explanatory variables $(X_1, \cdots, X_{j-1}, X_{j(b)}, X_{j+1}, \cdots, X_p)$. $\hat{\boldsymbol{Y}}_{j(b)}$ is the corresponding predictor of $\boldsymbol{Y}$. We define the permutation test statistic $F_{j(b)}$ by

$$F_{j(b)} = \frac{\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}_0\|^2 - \|\boldsymbol{Y} - \hat{\boldsymbol{Y}}_{j(b)}\|^2}{\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}_{j(b)}\|^2/(n-p)} \tag{4.4.3}$$

The estimated p-value is

$$\mathrm{p_v} = \frac{1}{B}\sum_{b=1}^{B} \mathbf{1}\left\{F_{j(b)} \geq F_j\right\} \tag{4.4.4}$$

where F_j is the usual F-statistic defined in equation (4.4.2) above. This resumes to

$$\mathrm{p_v} = \frac{1}{B}\sum_{b=1}^{B} \mathbf{1}\left\{\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}_1\|^2 \geq \|\boldsymbol{Y} - \hat{\boldsymbol{Y}}_{j(b)}\|^2\right\} \tag{4.4.5}$$

Similarly, we can derive a permutation t-test considering the following statistics,

$$T_{j(b)} = \frac{\hat{\beta}_{R,j}}{S(\hat{\beta}_{R,j})} \tag{4.4.6}$$

associated to the p-value

$$\mathrm{p_v} = \frac{1}{B}\sum_{b=1}^{B} \mathbf{1}\left\{T_{j(b)} \geq T_j\right\} \tag{4.4.7}$$

Note that these permutation tests are considered to be more conservative tests than parametric tests; moreover in our case, they may fail in case of very large values of γ too, and most of all, they do not take into account spatial dependence in the model and potentially spatially correlated covariates. We point out that in all the tests above, γ is chosen by the user and in our case, results from SLOO (each permutation gives its own optimal value).

4.5 Simulation experiments

In this section, we present simulation results to assess the performance of our methodology. We consider various scenarios, always in a multicollinearity framework. We simulate SAR and SEM models, with 8 highly correlated covariates, as described further, and for various values of ρ or λ in order to consider weak to strong spatial dependence. For each model,

we explore two settings; in the first one, we simulate the covariates once for all and only the error term is renewed at each simulation. In the second scenario, we simulate the covariates each time. Then, each resulting dataset is estimated according to different methods regardless whether the spatial feature and the Ridge regularization are taken into consideration. The code for our simulations is available in our GitHub repository `https://github.com/c0ra/RRSARMMI` along with the full results.

We consider a 30x30 grid over which we generate 8 covariates with multicollinearity issues. Here, we use the function RFsimulate from the package RandomFields in R.

The first two variables are generated as stationary and isotropic Gaussian random fields, with the exponential and Gaussian spatial covariance functions defined respectively as follows:

$$C_E(h) = \sigma^2 e^{-h/\theta} \text{ and } C_G(h) = \sigma^2 e^{-h^2/\theta}$$

where h is the distance between two points, σ^2 is the variance, and θ is the scale parameter that controls the strength of the spatial dependence.

X_1 is generated with an exponential covariance function with variance equal to 1 and scale set to 0.5, while X_2 has a Gaussian covariance function with variance 1 and scale set to 0.4. The remaining six variables are generated as follows,

$$X_3 = \exp(X_1) - |X_2| \qquad X_4 = |X_2| + (\frac{X_1 + X_2}{2} - 4)^2$$
$$X_5 = X_1 + X_1 X_2 \qquad X_6 = \log(X_4) - \frac{X_1}{12} + X_1^2$$
$$X_7 = X_1 + 2X_2 + \sqrt{X_4} \qquad X_8 = X_5 + \frac{X_2}{2} + X_2^2$$

Afterwards, all covariates are standardized to build the matrix $\mathbf{X}$. In the deterministic scenario, this matrix is the same for all simulations; the high multicollinearity is expressed by very large values of matrix $\mathbf{X}^T\mathbf{X}$'s condition number, 138 473.1, and by the variance inflation factor (VIF) for the covariates, given in Table 4.1; moreover we plot the correlation matrix of the covariates in Figure 4.1.

Table 4.1: Variance inflation factor (VIF) of the simulated covariates

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
4821.923	16507.146	13.756	3804.250	13.649	29.921	10690.157	74.898

The spatial weight matrix W was constructed using a row-standardized contiguity matrix with the Rook neighbourhood (which is similar on a regular grid to the four nearest neighbours system). Then we simulate the dependent variable $\mathbf{Y}$ according to the SAR or

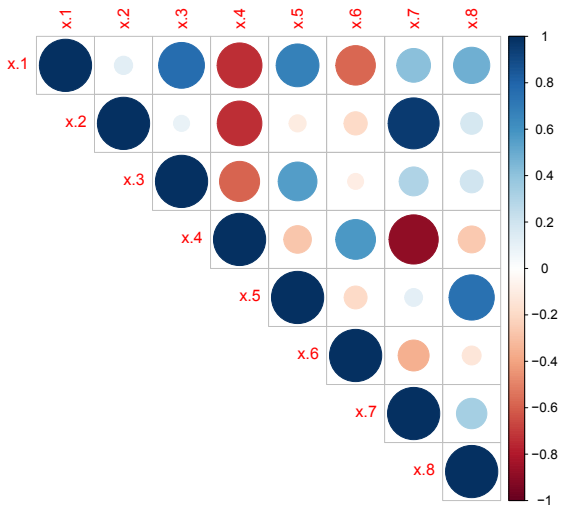


Figure 4.1: Correlation matrix of the 8 simulated covariates. This matrix visualizes the correlations between different variables used in the study. Each cell represents the correlation coefficient between two variables, the scale on the right provides a reference for interpreting the correlation coefficients, where blue indicates a positive correlation and red indicates a negative correlation. The intensity of the colour corresponds to the magnitude of the correlation, with darker shades signifying stronger relationships.

the SEM model, with 5 different values of the spatial autoregressive coefficient reflecting weak to strong dependence, ρ $(resp.\lambda) \in (0.1, 0.3, 0.5, 0.7, 0.9)$. The dependent variable $\mathbf{Y}$ was obtained by either

$$\mathbf{Y} = (I_n - \rho W)^{-1}\mathbf{X}\boldsymbol{\beta} + (I_n - \rho W)^{-1}\boldsymbol{\varepsilon}, \tag{4.5.1}$$

or

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + (I_n - \lambda W)^{-1}\boldsymbol{\varepsilon}, \tag{4.5.2}$$

where $\mathbf{X}$ is the covariates matrix, $\boldsymbol{\beta}$ is a vector of ones, and $\boldsymbol{\varepsilon}$ follows a Gaussian distribution with mean 0 and standard deviation 1. Then we centre $\mathbf{Y}$ but we do not scale it.

Each model is simulated 500 times, and then estimated. We consider the following estimation procedures, whose abbreviation that we shall use in the summaries is given between parenthesis: Ordinary least squares regression (OLS), ordinary Ridge regression (RR), ordinary SAR or SEM (SAR or SEM) without regularization, Spatially Filtered Ridge Regression (SFRR) proposed by Fan et al. (2017), and our estimation procedure, named as Ridge Regression for SAR models (RRSAR) or Ridge Regression for SEM models (RRSEM). In this regular grid setting, the buffer around each site $\mathbf{s}$ during the SLOO procedure is defined by the Rook neighbourhood.

For comparability purpose we compute the bias, variance and mean square error (MSE) of the regression coefficients β_1 to β_8 as well as for ρ or λ.

We summarize the results in tables 4.2, 4.3 and 4.4. We only display the results for the case of "stochastic" covariates simulated each time; those obtained for the case where we simulate them once for all are very similar. Moreover, for sake of place, we gathered the results for the regression coefficients in presenting the average of their bias, variance and MSE. We do not present the results between SFRR, RRSAR and RRSEM for the dependence parameters ρ and λ. Indeed, the bias, variances, and MSE of λ estimates are identical up to the third digit for SFRR and RRSEM, equal to ± 0.000; they are similar for the SAR framework, with a small difference in the bias; we observe a small bias of -0.001 in our procedure for ρ equal to 0.3 to 0.7, while it is -0.000 following the SFRR procedure. Detailed results for each regression coefficient and for each value of ρ or λ are retrievable from our GitHub repository.

Let us note that if we are going to compare the MSE between the different estimation procedures, one has to keep in mind that other diagnostics are ignored here, but obviously, SAR and SEM lead to poor estimates due to the multicollinearity issue, we can get low variances from RR, but it suffers from the absence of spatial dependence in the model, and OLS endures both inconveniences. Our real challenge is to compare our procedure with SFRR.

When examining the average bias, variance and MSE of regression coefficient estimates (Tables 4.2, 4.3, 4.4), a clear pattern emerges as we vary the value of the dependence parameter. Bias and variance (and thus MSE) consistently escalate with the spatial dependence parameter for OLS, occasionally resulting in extremely high values (e.g., an average variance of 656.4 is observed for $\rho = 0.9$). This illustrates the failure of this model in our framework and the strong need of regularization. SAR and SEM estimation procedures help to reduce the bias, which is almost constant whatever the strength of the spatial dependence; on the other hand, they lead to high (stable) variances of order 5 or 6. Unsurprisingly, the RR estimation procedure helps a lot to reduce the variances, but may lead to a large bias (2.5 for $\rho = 0.9$); it is interesting to note that the variances increase with ρ and λ denoting the lack of spatial feature inclusion. To summarize, as we transition from 0.1 to 0.9, OLS experiences a significant surge in MSE, multiplied by more than 100. In contrast, RR's MSE average is multiplied by "only" 25 for the SAR simulation, and increase by 40% over the same range for the SEM one.

As expected, the best results are obtained for SFRR and our algorithms. In both cases, the bias is more important than for ordinary SAR and SEM estimation, but this is expected since these techniques force the coefficients towards zero. Let us note that in all cases, the bias is more important after SFRR estimation than after our procedures. Especially in the SEM simulation, SFRR leads to increasing larger bias (-1 for $\lambda = 0.9$ and SFRR versus -0.4 for RRSEM). SFRR estimation leads to very low variances, lower than ours. Interestingly,

though they stay constant for the SAR simulation over the ρ variation, they increase with λ in the SEM framework for both methods, SFRR and RRSEM. Considering the MSE, our method is superior to SFRR, for both models. Our conclusion is that SFRR leads to an excessive decrease of the regression coefficients towards zero, raising the bias; indeed, our regularization procedure is sufficient to stabilise the coefficients, leading to lower bias.

Table 4.2: Average regression coefficient bias

	SAR simulation					SEM simulation				
ρ/λ	OLS	SAR	RR	SFRR	RRSAR	OLS	SEM	RR	SFRR	RRSEM
0.1	0.044	-0.236	-0.521	-0.551	-0.445	0.164	0.161	-0.533	-0.576	-0.422
0.3	0.746	-0.233	-0.360	-0.550	-0.444	0.166	0.165	-0.533	-0.703	-0.413
0.5	1.803	-0.229	-0.112	-0.547	-0.443	0.176	0.168	-0.547	-0.810	-0.406
0.7	3.714	-0.224	0.441	-0.548	-0.443	0.199	0.168	-0.559	-0.933	-0.402
0.9	9.092	-0.220	2.567	-0.543	-0.443	0.280	0.167	-0.551	-1.051	-0.400

Table 4.3: Average regression coefficient variance

	SAR simulation					SEM simulation				
ρ/λ	OLS	SAR	RR	SFRR	RRSAR	OLS	SEM	RR	SFRR	RRSEM
0.1	6.739	5.788	0.053	0.053	0.109	6.277	6.201	0.068	0.070	0.131
0.3	11.702	5.778	0.078	0.053	0.109	6.968	6.227	0.074	0.079	0.143
0.5	26.719	5.769	0.083	0.052	0.109	8.484	6.126	0.075	0.110	0.155
0.7	83.771	5.765	0.167	0.052	0.109	12.301	5.909	0.086	0.141	0.166
0.9	656.366	5.776	1.519	0.054	0.109	32.018	5.598	0.194	0.209	0.170

Table 4.4: Average regression coefficient MSE

	SAR simulation					SEM simulation				
ρ/λ	OLS	SAR	RR	SFRR	RRSAR	OLS	SEM	RR	SFRR	RRSEM
0.1	6.741	5.844	0.323	0.356	0.307	6.304	6.226	0.352	0.401	0.309
0.3	12.258	5.832	0.207	0.355	0.307	6.996	6.254	0.358	0.573	0.313
0.5	29.971	5.822	0.095	0.351	0.306	8.514	6.154	0.374	0.766	0.320
0.7	97.565	5.815	0.361	0.353	0.306	12.341	5.938	0.399	1.013	0.328
0.9	739.036	5.824	8.107	0.349	0.305	32.096	5.625	0.497	1.314	0.330

4.6 Application

We apply our methodology to a real data set. The data is related to the COVID-19 epidemic in metropolitan France in 2020. The observations are collected from the 96 "départements", which are administrative units. Their geographic distribution is illustrated in Figure 4.2, including the Corsica island, and we will call them "departments" from now. We consider the hospitalization rate due to the COVID-19, reflecting the strength of the

epidemic, and a set of socio-economic covariates, which are highly correlated. The aim of the study is to determine how these covariates impact on the dependent variable. First, we conduct an exploratory analysis of the data to bring out spatial dependence as well as multicollinearity issues. Then we consider both SAR and SEM models; we estimate the coefficients of each model using Ridge regularization, according to one or the other of the iterative algorithms RRSAR and RRSEM derived in section 4.3.2; the Ridge parameter is chosen following a spatial leave-one-out procedure, as described in paragraph 4.3.3. Once the model is estimated, the significance of the explanatory variables is determined by running permutation F-tests presented in section 4.4. Finally, we compare our results with those obtained by classical estimation of SAR and SEM models. Our code and results are available on our GitHub repository `https://github.com/c0ra/RRSARMMI`

4.6.1 Data description

The COVID-19 epidemic in France, upon its arrival at the end of February 2020, did not affect the entire territory equally. The "Grand Est", "Hauts-de-France" and "Ile-de-France" regions, corresponding roughly to East France, North, and Paris and its suburbs, concentrated the highest number of cases in the initial months of 2020. The spatial study of the state of the epidemic before the introduction of vaccination campaigns is a crucial tool to provide information on the behaviour of future epidemics, as well as to suggest in which sites more resources should be invested to prepare facilities and personnel capacities to face this kind of health emergencies.

Amdaoud et al. (available at EconomiX Working Papers `https://ideas.repec.org/p/drm/wpaper/2020-4.html`) studied the spatial distribution of three indicators of the intensity of the COVID-19 epidemic. These indicators are the hospitalization rate, the mortality rate, and the excess of mortality rate; the data are collected from 19 March 2020 to 12 May 2020 and aggregated at the scale of the department. Even though we studied the three indicators, we will focus here on the hospitalization rate; the results for the two others are analogous. The exploratory analysis revealed a heterogeneous spatial distribution, as can be seen in Figure 4.2, and the authors proposed to use spatial autoregressive models involving some socio-economic covariates to explain the spatial behaviour of the epidemic. We extend their study, including more updated data and more importantly, a larger number of socio-economic variables; actually, Amdaoud et al. removed some covariates to avoid multicollinearity problems.

For better linearisation fitting, we consider the logarithm of some variables; thus, the response variable, denoted by `LnHosp`, is the logarithm of the hospitalization rate, attributed to the COVID-19. We consider 8 covariates; some are related to the general population characteristics: population density (`LnPop`), proportion of people aged 65 and

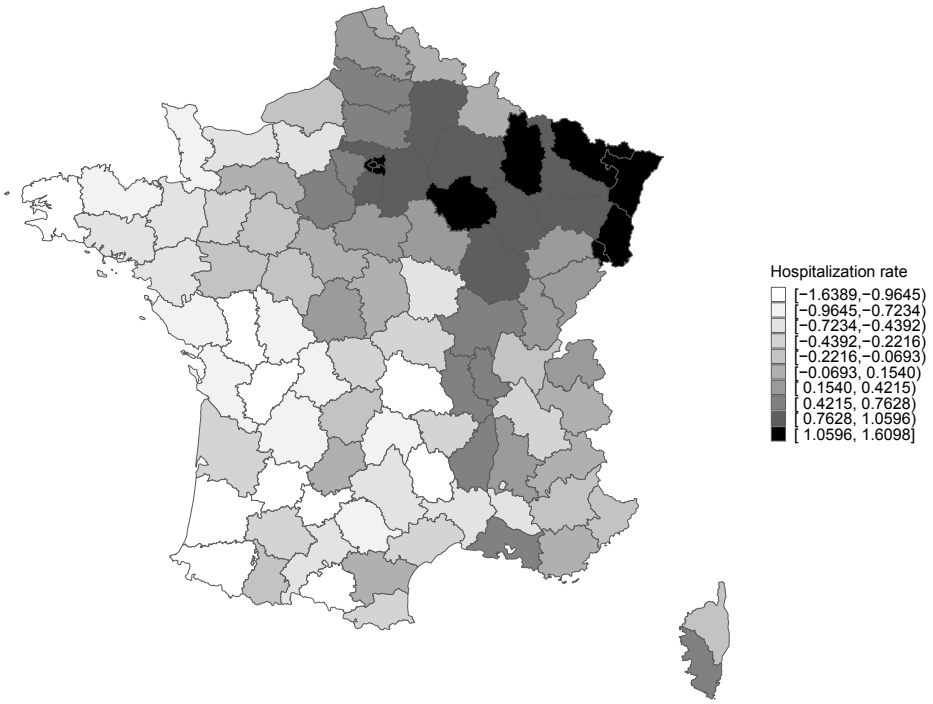


Figure 4.2: Hospitalization rate by French department (by decile)

over (`A65pls`), share of the three main municipalities in the population (`C3`). Three variables are linked to social aspects: the proportions of workers (`Work`), inactive people (`Inac`) and the disparities of income (`SLiveR`); finally two variables reflect the professional health care facilities, the rate of doctors (`FDoc`) and the rate of emergency services (`Emer`). The exact definition of the variables are listed in table **??** in the Appendix. Let us note that these covariates are considered as explanatory factors of the health status of a population and its mortality rate (Link and Phelan, 1995).

4.6.2 Exploratory analysis

First, we run a PCA on the covariates, adding the hospitalization rate as a supplementary variable; PCA is commonly used before modelling real data. It helps to better understand our data and the relationships between the variables. Full results are displayed in Appendix **??**. To summarize, it shows some spatial behaviour. First by opposing rural and urban departments, second by observing that departments that are close on the map of individuals are also geographically close. PCA also shows that the hospitalization rate is mostly characterized by the variables featuring the first component, `LnPop, Inac, SLivR, A65pls, Emer` and `Work`.

Then we turn to the spatial auto-correlation, classically measured by the Moran's in-

dex, see (Moran, 1950). According to the map of French departments, we choose the classic Queen neighbourhood, displayed in Figure 4.3, and in a standard way we choose the weights of the associated matrix W to be equal to the inverse distance between adjacent departments and 0 elsewhere. However, contrary to the usual row-standardization, we normalize W by its spectral radius. The Moran's one-sided randomization tests confirms the presence of a strong positive spatial autocorrelation for the hospitalization rate, for neighbours up to the second order, see Table 4.5.

Table 4.5: Moran's one-sided randomization tests with alternative "greater"

Variable	**Neigh.Order**	**Moran I**	**Mean**	**SD**	**Statistic**	**p-value**
`LnHosp`	1	0.988	-0.0105	0.0069	12.13	3.65e-34
	2	0.576	-0.0107	0.0023	11.98	2.20e-33

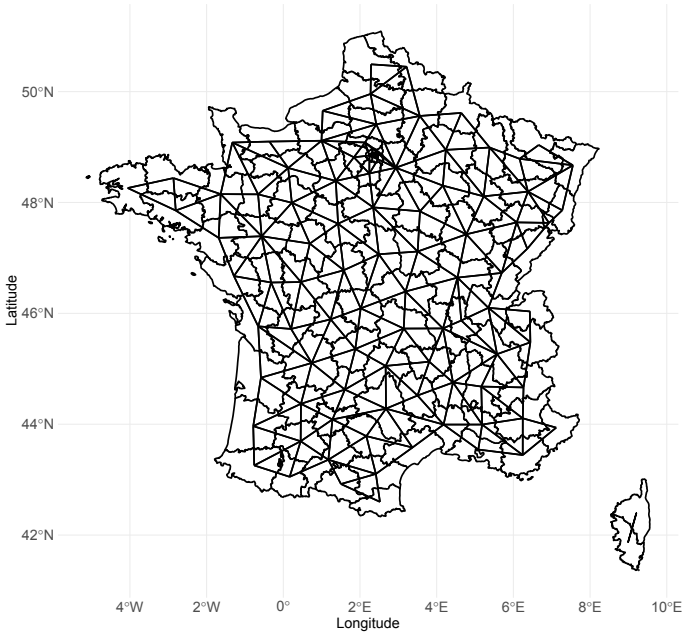


Figure 4.3: Queen neighbourhood graph for French departments

Multi-collinearity

We compute the correlation matrix between the variables; the correlation matrix plot displayed in Figure 4.4 shows that the pairwise correlation is high for several pairs of variables, with the highest values obtained for the pairs `LnPop` - `SLivR`, `LnPop` - `A65pls`, `Work` - `SLivR` and `LnPop` - `Emer`.

Furthermore, some multicollinearity tests and individual diagnostic tests were performed to detect problematic variables. When performing the overall multicollinearity diagnostic tests, a small value of the determinant of the correlation matrix ($|\mathbf{X}^T\mathbf{X}| = 0.0199$) was found, accompanied by a high value of the Farrar's statistic (358.3260). The individual diagnostic tests for multicollinearity point to the variables `LnPop`, `SLivR`, `Work`, `Inac` and `A65pls` as probable sources of multicollinearity.

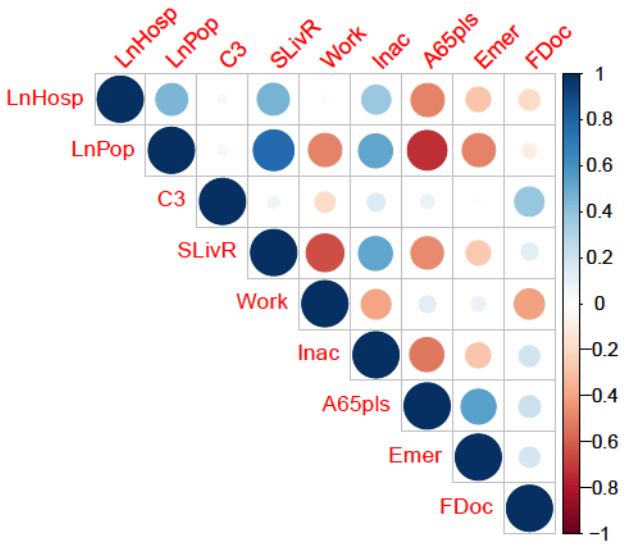
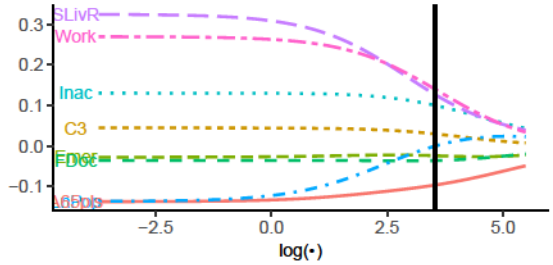


Figure 4.4: Correlation matrix of study variables. This matrix visualizes the correlations between different variables used in the study. Each cell represents the correlation coefficient between two variables, the scale on the right provides a reference for interpreting the correlation coefficients, where blue indicates a positive correlation and red indicates a negative correlation. The intensity of the colour corresponds to the magnitude of the correlation, with darker shades signifying stronger relationships.

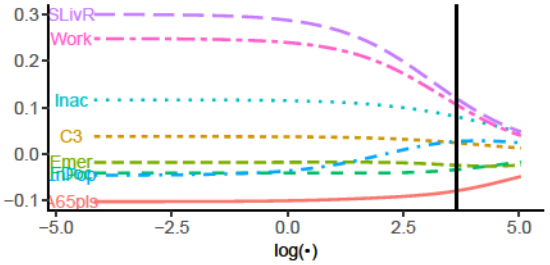
4.6.3 Results

In all what follows, the explanatory variables are centred and scaled, and `LnHosp` is centred. SAR and SEM models are estimated under their ordinary procedure (without regularization) and following our algorithms RRSAR and RRSEM derived in Section 4.3.2. Let us recall that in each Ridge estimation procedure, the regularization parameter γ is determined by SLOO; to this aim, the dead zone is defined for each department by its first neighbours according to the Queen neighbourhood graph.

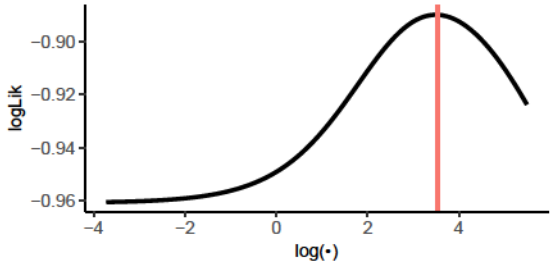
Figure 4.5 displays the evolution of the coefficients values (4.5a and 4.5b) as well as



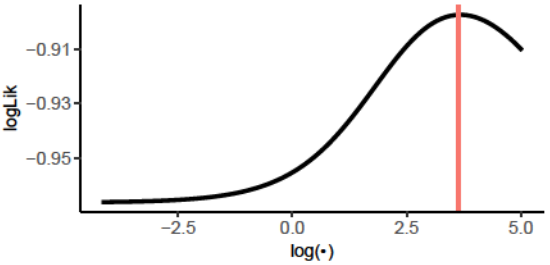
(a) SAR β_{ridge} coefficients paths as a function of $\log(\gamma)$



(b) SEM β_{ridge} coefficients paths as a function of $\log(\gamma)$



(c) SAR Ridge log-likelihood as a function of $\log(\gamma)$



(d) SEM Ridge log-likelihood as a function of $\log(\gamma)$

Figure 4.5: Coefficients paths and average log-likelihood obtained in the SLOO cross-validation study as a function of the logarithm of the regularization parameter. Optimal regularization parameter for Ridge SAR and Ridge SEM is represented by a vertical line.

the average log-likelihood obtained in the SLOO cross-validation study (4.5c and 4.5d) in function of $\log(\gamma)$. The coloured lines are paths of regression coefficients, and the vertical line gives the set of coefficients corresponding to the value of γ selected via SLOO. We observe as expected the classic behaviour of the coefficients decreasing towards zero as the regularization parameter γ increases, but the behaviour is different between the variables. While the coefficients of `Inac, C3, Emer` are very stable, those of `SLivR, Work, A65pls` head towards zero; the coefficient of `LnPop` is very particular as it changes its sign in the SEM framework. We also note that the selected γ values according to one or the other model are similar; indeed, the values are about 34.26 for the SAR model and 38.02 for the SEM (see Table 4.7). Finally, we emphasize that the maximum log-likelihood coincides with the selected γ.

Once the model is estimated, we want to assess the importance of the explanatory variables. When the estimation has been performed under ordinary SAR and SEM modelling, we report the p-values corresponding to a two-sided z-test. For the Ridge versions, we conduct different tests: the t-test for Ridge estimators (Halawa and El Bassiouni, 2000) (here with 88 degrees of freedom), the permutation t-test, and the permutation F-test, as

described in Section 4.4. Here, we chose to run 100 permutations for each of the 8 variables in the permutation versions. We present the results in Table 4.6 and Table 4.7. Note that the values of the regression coefficients in this table have been multiplied each one by the sample standard deviation of the corresponding covariate (but the tests statistics are computed with the original coefficient).

Results are quite coherent. Before analysing them, we point out that here, of course, we do not know the true model. The absolute value of the regression coefficients are without interest, while we are mostly interested in their p-value.

We see how collinearity affects the results for the ordinary SAR and SEM models; for instance in the SAR and SEM estimation, the coefficients of `LnPop` is negative, though this variable is positively correlated with `LnHosp`. The most correlated variables with `LnHosp` are `LnPop, SLivR, Inac` and `A65pls,` but `LnPop` and `A65pls` appear to be non-significant; `Inac` is also non-significant in the SEM model.

The regularized estimation of those models suppress most of these hassles.

	SAR		RRSAR			
				R. Perm. F-test	R. t-test	R. Perm. t-test
	Coef	p-value	Coef	p-value	p-value	p-value
`LnPop`	-0.1256	0.2297	-0.0006	0.8900	0.4910	0.4000
`C3`	0.5337	0.4365	0.3333	0.6500	0.5157	0.2700
`SLivR`	0.6277	0.0029	0.2593	0.0000	0.0031	0.0000
`Work`	12.8074	0.0019	6.7131	0.0000	0.0027	0.0100
`Inac`	6.7543	0.0742	5.1264	0.0800	0.0184	0.0300
`A65pls`	-0.0357	0.1713	-0.0257	0.0800	0.0108	0.9900
`Emer`	-6.6582	0.6206	-4.8059	0.7300	0.2132	0.7300
`FDoc`	-0.1584	0.7012	-0.2118	0.9700	0.4542	0.7200
γ			34.2557			
ρ	0.8489	0.0001	0.7282	0.0003		

Table 4.6: Coefficients and p-values of variable importance tests for hospitalization rate models. For SAR, we report the p-values corresponding to a two-sided z-test. For RRSAR, importance is determined by tests described in Section 4.4 with 100 permutations for each of the 8 variables in the case of the permutation tests.

	SEM		RRSEM			
				R. Perm. F-test	R. t-test	R. Perm. t-test
	Coef	p-value	Coef	p-value	p-value	p-value
LnPop	-0.0363	0.7775	0.0211	0.5300	0.2019	0.2400
C3	0.4144	0.5608	0.2672	0.7500	0.5964	0.2900
SLivR	0.6293	0.0036	0.2455	0.0000	0.0065	0.0000
Work	12.0400	0.0101	5.1042	0.0100	0.0218	0.0100
Inac	6.0018	0.1553	4.1157	0.2100	0.0586	0.0300
A65pls	-0.0284	0.3336	-0.0213	0.1300	0.0393	0.9800
Emer	-3.4274	0.8062	-4.9290	0.7300	0.2242	0.7100
FDoc	-0.2583	0.5795	-0.2010	0.6300	0.5110	0.7800
γ			38.0189			
λ	0.8549	0.0047	0.8996	0.0000		

Table 4.7: Coefficients and p-values of variable importance tests for hospitalization rate models. For SEM, we report the p-values corresponding to a two-sided z-test. For RRSEM, importance is determined by tests described in Section 4.4 with 100 permutations for each of the 8 variables in the case of the permutation tests.

First, the coefficient of LnPop becomes positive in RRSEM, and is very close to zero in RRSAR (we have seen that it would become positive with a larger γ in Figure 4.5a), which is an improvement. We note that the variables SlivR, Work that were significant, stay significant. On the other side, we see that variables Emer, FDoc are non-significant. This non-significance could result from the fact that these variables are considered among others that could better capture the relationships with the hospitalization rate, or their direct impact could be too small to be detected by tests. Then, very interestingly, the three tests do not provide identical results. Globally, the permutation F-test appears to be more conservative than the t-tests. The variable Inac which was barely significant in the SAR model (p-value equal to 0.0742) keeps the same level of significance for the F-test but becomes important according to the two t-tests; it also becomes significant in the SEM framework according to the same t-tests. One other interesting variable is A65pls; it is not significant in the SAR and SEM models after the ordinary estimation procedure; but it becomes significant after the Ridge regularization following the t-test in both models, and barely significant (at level 10%) for the F-test in the SAR framework (note that its p-value has jumped from 33% down to 13% in the SEM model). The result of the permutation t-test for this variable is surprising, giving a very large p-value of about 98% for both models. To summarize, it is clear that variables Work and SLivR are very significant to explain the hospitalization rate due to COVID-19, which reflects the importance of inequalities and low income, before all. The results obtained for Inac support this idea. Then, the proportion

of elder people is also determinant, illustrated by the p-values of `A65pls`.

4.7 Conclusions

Multi-collinearity is a common feature of real life data; it also affects the class of simultaneous spatial autoregressive models in spatial econometrics.

We propose estimation algorithms that consider multicollinearity in the estimation of all the parameters using a regularization technique of type Ridge; then, the regularization parameter is obtained via a spatial ad-hoc cross-validation procedure; SLOO is particularly well adapted to spatial autoregressive models since there's no issue defining the size of the buffer surrounding the validation set, it is naturally determined by the first order neighbours. A drawback of SLOO is that it is computationally expensive in the case of a large number of observations. Some users propose to choose a large proportion of observations to serve as validation sets in the SLOO, instead of all the observations. These points can be chosen either randomly or regularly spaced. After Ridge regularization, the question of the importance of the covariates rises. We considered three different tests adapted to ridge regression; ran on the application, they provided coherent and different results. The question of a test adapted to both regularization and spatial dependence, including also the spatial dependence of the covariates themselves, is still an open problem.

The code for reproducing the results of this chapter appears in the GitHub project https://github.com/c0ra/RRSARMMI/.

Our methodology can be extended in several directions. First, for variable selection purpose, the procedure can be easily adapted to Lasso regression and Elastic net. Next, it can also be further developed to consider other models like the general spatial model (which roughly speaking integrates both SAR and SEM in one model) or the Durbin model, which extends the SAR model to include spatially lagged explanatory variables.

Conclusions

Chapter 1. Chapter 1 presents the importance of adapting classical cross-validation procedure for reliably estimating the risk of an algorithm when data exhibit spatial structure. We present various spatial cross-validation techniques and study their behaviour in situations of low to strong spatial dependence. The main take away from this chapter is that the methods that maintain sufficient separation between training and validation sets, while retaining enough data for modeling, provide more accurate empirical risk estimates. In particular, spatial blocked cross-validation (SBCV) and spatial leave-one-out (SLOO) accomplish improved performance evaluations.

Chapter 2. In Chapter 2, we define the optimal buffer size as the value that minimizes the gap between the "ideal" risk and the corresponding cross-validation risk. We also compare three practical methods for buffer-size selection: two from the literature and one that we propose, which is particularly relevant in real-world scenarios with a single realization of the spatial process. While the variogram-based approach can be effective under certain conditions, the results for both SLOO and SBCV indicate that it may still produce higher discrepancy values, particularly at larger spatial dependencies or block sizes. This highlights the method's sensitivity to both the assumed variogram model and the scale of spatial dependence. For SLOO, the SAC-0.1 threshold and $\text{Moran}_{0.05}$ yield similarly low discrepancies, whereas under SBCV, the SAC-0.1 method consistently matches or surpasses this performance across various spatial dependencies and block sizes.

Chapter 3. In Chapter 3, we introduce a parametric bootstrap approach for selecting the optimal buffer size. After fitting a spatial covariance model to the residuals, we generate multiple bootstrap replicates to approximate the ideal predictive risk, an otherwise unobservable quantity in single, realization studies. Simulation experiments confirm that this method tracks the true optimal buffer size, even under varying strengths of spatial autocorrelation. Additionally, a subsampling scheme (SSLOO) is proposed to reduce computational overhead occurring in leave-one-out demonstrating that one can substantially lower runtime without compromising accuracy. The chapter concludes by applying these techniques to a benchmark simulation inspired by species-abundance data, showcasing the

robustness of the bootstrap method across diverse settings.

Chapter 4. In Chapter 4 we introduce a new estimation procedure that addresses the impact of multicollinearity in spatial autoregressive models through a Ridge-type regularization. Crucially, the regularization parameter is chosen via an ad-hoc spatial cross-validation strategy, where SLOO naturally defines the buffer by first-order neighbors. This built-in spatial structure obviates much of the complexity around buffer size determination. Simulation studies validate the new estimation approach, and an application to COVID-19 data in France illustrates how it can provide explanatory models for multiple epidemic indicators under real-world conditions.

Limitations and perspectives

1. The simulation techniques presented in this thesis assume regularly distributed observations, thus guaranteeing uniform measurement density. In practice, perfectly uniform sampling is often unattainable, and clustered sampling designs are commonplace in areas such as ecology Ploton et al. (2020), air quality research Xiao et al. (2018), or soil analysis Hengl et al. (2015). This phenomenon of clustered sampling is especially prevalent on a global scale as mentioned by Meyer and Pebesma (2022). When data are clustered, the regions from which samples are collected tend to be overly represented. This leads to over-representation of certain regions and under-representation of others in prediction locations (de Bruin et al., 2022). For such settings, methods like the Nearest Neighbour Distance Matching Leave-One-Out Cross-Validation proposed by Mila et al. (2022) can outperform buffered approaches if the expected distance between training data and predictions is known. While our results emphasize the benefits of spatial CV under regular sampling, we acknowledge that no single CV method is universally optimal for all sampling designs.

   Regarding block-based CV (BCV and SBCV), we assigned blocks purely by spatial location and did not consider strategies that preserve the distribution of outcomes or predictors in the validation set. Depending on the structure of the dependent variable, simplistic blocking may yield validation sets with systematically different outcome distributions from the training data, potentially introducing bias in performance estimates.

2. Throughout this work, we adopted a deterministic explanatory variables framework. This common assumption treats the covariates as non-random, thereby evaluating predictive performance for a specific spatial configuration. Although this framework facilitates precise model assessment, it limits generalizability when environmental or

other predictors vary across space. Consequently, the predictive risk estimates may not transfer seamlessly to new spatial arrangements. Despite these drawbacks, the fixed-$\mathbf{X}$ framework remains a valuable tool in controlled settings, allowing a focused analysis of model performance under known conditions. Likewise, transitioning from a fixed-$\mathbf{X}$ to a random-$\mathbf{X}$ design framework may capture wider variability in real applications.

3. Our study employed a linear regression model with ordinary least squares estimation. However, future work could explore alternative approaches, such as generalized least squares or more flexible models like random forests, which may provide greater flexibility and robustness in handling spatial dependencies. That said, the use of OLS circumvents the need to fit the empirical covariance matrix $\hat{\Sigma}$, which is required for feasible generalized least squares. As demonstrated in Chapter 2, feasible generalized least squares estimation introduces challenges, particularly the variability of variogram parameters, which becomes more pronounced as spatial dependence increases.

4. In future work, penalized empirical risk criteria could help address scenarios of very strong dependence or highly uneven sampling designs. Ongoing theoretical studies aim to refine these bootstrap-based techniques by incorporating suitable penalties and extending the framework to more complex spatial models. A promising direction is to incorporate penalty functions over the parameter space, thus building estimators via data-driven projection methods. Such an approach, coupled with concentration inequalities (as in Fermín and Ludena (2008)), could yield more theoretically grounded solutions to choosing optimal buffer sizes based on observed spatial structure.

5. The methodology of Chapter 4 can be easily adapted for variable selection using Lasso or Elastic Net penalties. Extending it to more general spatial models—such as those integrating spatial error components or the Durbin model—would broaden its applicability to a wider array of real-world spatial processes. By incorporating spatially lagged predictors, these extensions further refine the model's capacity to capture complex dependence structures.

Bibliography

Aldworth, J. and Cressie, N. (1999). Sampling designs and prediction methods for gaussian spatial processes. In *Multivariate analysis, design of experiments, and survey sampling*, pages 25–78. CRC Press.

Alin, A. (2010). Multicollinearity. *WIREs Computational Statistics*, 2:370–374.

Anderson, M. J. and Robinson, J. (2001). Permutation tests for linear models. *Australian & New Zealand Journal of Statistics*, 43:75–88.

Anselin, L. (1988). *Spatial econometrics: methods and models*. Dorddrecht: Kluwer Academic Publishers, Dordrecht.

Arlot, S., Celisse, A., et al. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79.

Bach, F. R. (2020). Learning theory from first principles lecture 3: Empirical risk minimization.

Bates, S., Hastie, T. J., and Tibshirani, R. (2021). Cross-validation: What does it estimate and how well does it do it? *Journal of the American Statistical Association*, 119:1434 – 1445.

Bécu, J.-M. (2016). *Contrôle des fausses découvertes lors de la sélection de variables en grande dimension*. Theses, Université de Technologie de Compiègne.

Breiman, L. (1992). The little bootstrap and other methods for dimensionality selection in regression: X-fixed prediction error. *Journal of the American Statistical Association*, 87:738–754.

Breiman, L. and Spector, P. (1992). Submodel selection and evaluation in regression. the x-random case. *International statistical review/revue internationale de Statistique*, pages 291–319.

Brenning, A. (2012). Spatial cross-validation and bootstrap for the assessment of prediction rules in remote sensing: The r package sperrorest. Paper presented at the 2012 IEEE

international geoscience and remote sensing symposium, IEEE, Munich, 5372–5375 July 2012.

Burman, P. (1989). A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76(3):503–514.

Burman, P., Chow, E., and Nolan, D. (1994). A cross-validatory method for dependent data. *Biometrika*, 81(2):351–358.

Bécu, J. M., Grandvalet, Y., Ambroise, C., and Dalmasso, C. (2017). Beyond support in two-stage variable selection. *Statistics and Computing*, 27:169–179.

Cornes, R. C., van der Schrier, G., van den Besselaar, E., and Jones, P. D. (2018). Journal of geophysical research: Atmospheres an ensemble version of the e-obs temperature and precipitation data sets.

Cressie, N. (2015). *Statistics for Spatial Data*. John Wiley & Sons.

de Bruin, S., Brus, D. J., Heuvelink, G. B., van Ebbenhorst Tengbergen, T., and Wadoux, A. M.-C. (2022). Dealing with clustered samples for assessing map accuracy by cross-validation. *Ecological Informatics*, 69:101665.

Dhingra, M. S., Artois, J., Robinson, T. P., Linard, C., Chaiban, C., Xenarios, I., Engler, R., Liechti, R., Kuznetsov, D., Xiao, X., von Dobschuetz, S., Claes, F., Newman, S. H., Dauphin, G., and Gilbert, M. (2016). Global mapping of highly pathogenic avian influenza h5n1 and h5nx clade 2.3.4.4 viruses with spatial cross-validation. *eLife*, 5.

Diggle, P. J. and Ribeiro, P. J. (2007). Generalized linear models for geostatistical data. *Model-based geostatistics*, pages 79–98.

Evans, J. S., Murphy, M. A., Holden, Z. A., and Cushman, S. A. (2011). Modeling species distribution and change using random forest. In Drew, C. A., Wiersma, Y. F., and Huettmann, F., editors, *Predictive Species and Habitat Modeling in Landscape Ecology: Concepts and Applications*, pages 139–159. Springer New York, New York, NY.

Fan, C., Rey, S. J., and Myint, S. W. (2017). Spatially filtered ridge regression (sfrr): A regression framework to understanding impacts of land cover patterns on urban climate. *Transactions in GIS*, 21:862–879.

Fermín, A. K. and Ludena, C. (2008). A statistical view of iterative methods for linear inverse problems. *Test*, 17:381–400.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1–22.

Halawa, A. and El Bassiouni, M. (2000). Tests of regression coefficients under ridge regression models. *Journal of Statistical Computation and Simulation*, 65(1-4):341–356.

Hastie, T. and Tibshirani, R. (1995). Generalized additive models for medical research. *Statistical Methods in Medical Research*, 4:187–196.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, New York.

Hengl, T., Heuvelink, G. B. M., Kempen, B., Leenaars, J. G. B., Walsh, M. G., Shepherd, K. D., Sila, A., MacMillan, R. A., Mendes de Jesus, J., Tamene, L., and Tondoh, J. E. (2015). Mapping soil properties of africa at 250 m resolution: Random forests significantly improve current predictions. *PLOS ONE*, 10(6):1–26.

Henry Crosby, T. D. and Jarvis, S. A. (2020). Road and travel time cross-validation for urban modelling. *International Journal of Geographical Information Science*, 34(1):98–118.

Hoerl, A. and Kennard, R. (1988). Ridge regression. In Kotz, S., Johnson, N. L., and Read, C. B., editors, *Encyclopedia of Statistical Sciences*, volume 42, pages 129–136. Wiley, New York.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hoffimann, J., Zortea, M., de Carvalho, B., and Zadrozny, B. (2021). Geostatistical learning: Challenges and opportunities. *Frontiers in Applied Mathematics and Statistics*, 7.

Holland, J. D., Bert, D. G., and Fahrig, L. (2004). Determining the Spatial Scale of Species' Response to Habitat. *BioScience*, 54(3):227–233.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.

Jemeļjanova, M., Kmoch, A., and Uuemaa, E. (2024). Adapting machine learning for environmental spatial data - a review. *Ecological Informatics*, 81:102634.

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

Kennedy, F. E. (1995). Randomization tests in econometrics. *Journal of Business & Economic Statistics*, 13:85–94.

Killeen, P., Kiringa, I., Yeap, T. H., and Branco, P. (2024). Corn grain yield prediction using uav-based high spatiotemporal resolution imagery, machine learning, and spatial cross-validation. *Remote. Sens.*, 16:683.

Lahiri, S. N. (2003). Resampling methods for dependent data.

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

Le Rest, K., Pinaud, D., Monestiez, P., Chadoeuf, J., and Bretagnolle, V. (2014). Spatial leave-one-out cross-validation for variable selection in the presence of spatial autocorrelation. *Global Ecology and Biogeography*, 23(7):811–820.

LeSage, J. P. (2008). An introduction to spatial econometrics. *Revue d'économie industrielle*, 123:19–44.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2:18–22. https://journal.r-project.org/articles/RN-2002-022/.

Link, B. G. and Phelan, J. (1995). Social conditions as fundamental causes of disease. *Journal of Health and Social Behavior*, pages 80–94.

Mahoney, M., Johnson, L., Silge, J., Frick, H., Kuhn, M., and Beier, C. (2023a). Assessing the performance of spatial cross-validation approaches for models of spatially structured data.

Mahoney, M. J., Johnson, L. K., Silge, J., Frick, H., Kuhn, M., and Beier, C. M. (2023b). Assessing the performance of spatial cross-validation approaches for models of spatially structured data. *arXiv preprint arXiv:2303.07334*.

Manly, B. F. J. (2006). *Randomization, bootstrap and Monte Carlo methods in biology*, volume 70. CRC press, Boca Raton.

Massart, P. (2007). *Concentration inequalities and model selection: Ecole d'Eté de Probabilités de Saint-Flour XXXIII-2003*. Springer.

McDonald, G. C. (2009). Ridge regression. *WIREs Computational Statistics*, 1:93–100.

Melton, J. R., Chan, E., Millard, K., Fortier, M., Winton, R. S., Martín-López, J. M., Cadillo-Quiroz, H., Kidd, D., and Verchot, L. V. (2022). A map of global peatland extent created using machine learning (peat-ml). *Geoscientific Model Development*, 15(12):4709–4738.

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

Mila, C., Mateu, J., Pebesma, E., and Meyer, H. (2022). Nearest neighbour distance matching leave-one-out cross-validation for map validation. *Methods in Ecology and Evolution*, 13(6):1304–1316.

Misiuk, B., Diesing, M., Aitken, A., Brown, C. J., Edinger, E. N., and Bell, T. (2019). A spatially explicit comparison of quantitative and categorical modelling approaches for mapping seabed sediments using random forest. *Geosciences*, 9(6):254.

Moran, P. A. P. (1950). A test for the serial independence of residuals. *Biometrika*, 37:178–181.

Perez-Melo, S. and Kibria, B. M. G. (2020). On some test statistics for testing the regression coefficients in presence of multicollinearity: A simulation study. *Stats*, 3(1):40–55.

Ploton, P., Mortier, F., Réjou-Méchain, M., Barbier, N., Picard, N., Rossi, V., Dormann, C., Cornu, G., Viennois, G., Bayol, N., et al. (2020). Spatial validation reveals poor predictive performance of large-scale ecological mapping models. *Nature communications*, 11(1):4540.

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

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, 566:195 – 204.

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

Rosset, S. and Tibshirani, R. J. (2020). From fixed-x to random-x regression: Bias-variance decompositions, covariance penalties, and prediction error estimation. *Journal of the American Statistical Association*.

Rubinstein, R. Y. (1981). Simulation and the Monte Carlo method. In *Wiley series in probability and mathematical statistics*.

Stock, A., Haupt, A. J., Mach, M. E., and Micheli, F. (2018). Mapping ecological indicators of human impact with statistical and machine learning methods: Tests on the california coast. *Ecol. Informatics*, 48:37–47.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Tikhonov, A. N. and Arsenin, V. I. (1977). *Solutions of Ill-posed Problems*. Wiley, New York.

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

Urbazaev, M., Cremer, F., Migliavacca, M., Reichstein, M., Schmullius, C., and Thiel, C. (2018). Potential of multi-temporal alos-2 palsar-2 scansar data for vegetation height estimation in tropical forests of mexico. *Remote Sensing*, 10(8).

Valavi, R., Elith, J., Lahoz-Monfort, J. J., and Guillera-Arroita, G. (2019). blockcv: An r package for generating spatially or environmentally separated folds for k-fold cross-validation of species distribution models. *Methods in Ecology and Evolution*, 10(2):225–232.

Vapni, V. N. (2000). The nature of statistical learning theory. In *Statistics for Engineering and Information Science*.

Wadoux, A. M.-C., Heuvelink, G. B., de Bruin, S., and Brus, D. J. (2021). Spatial cross-validation is not the right way to evaluate map accuracy. *Ecological Modelling*, 457:109692.

Wang, J., Hopkins, L., Hallman, T., Robinson, W. D., and Hutchinson, R. (2023a). Cross-validation for geospatial data: Estimating generalization performance in geostatistical problems. *Transactions on Machine Learning Research*.

Wang, Y., Khodadadzadeh, M., and Zurita-Milla, R. (2023b). Spatial+: A new cross-validation method to evaluate geospatial machine learning models. *International Journal of Applied Earth Observation and Geoinformation*, 121:103364.

Wheeler, D. C. (2009). Simultaneous coefficient penalization and model selection in geographically weighted regression: The geographically weighted lasso. *Environment and Planning A: Economy and Space*, 41:722–742.

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

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society*, 67:301–320.