Voronoi tessellation-based lifting scheme in bounded regions

Fatih Gezer

Department of Statistics

University of Leeds

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

December 2021

Declaration

The candidate confirms that the work submitted is his own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

The findings in Chapter 2 of the thesis are published as Gezer, F., Aykroyd, R. G., & Barber, S. (2021). "Statistical properties of Poisson-Voronoi tessellation cells in bounded regions". *Journal of Statistical Computation and Simulation*, 91(5), 915-933.

The motivation of the investigation of Poisson-Voronoi tessellation cells and their statistical properties in bounded regions come from discussions with Stuart Barber and Robert Aykroyd. Fatih Gezer (1) conducted the simulation study, (2) demonstrated the differences in the statistical properties of Voronoi cells for the infinite plane, unit square, and convex hull boundary cases, and (3) approximated the distributions of cell properties by parametric distributions.

To my mother and father, for their endless and unconditional support...

Acknowledgements

I would like to express my sincere gratitude and appreciation to my supervisors Dr. Stuart Barber and Dr. Robert G. Aykroyd for their continuous support and guidance with their immense knowledge during my PhD. I am grateful for their kindness and approachability whenever I needed their help. It was an exceptional experience for me to work with them. I am also grateful for the scholarship from the Ministry of National Education, Republic of Turkey during my PhD.

I would like to thank Dr. John Paul Gosling and Prof. Charles Taylor for their useful comments at my annual reviews, and Dr. Matthew Aldridge and Prof. Janine Illian for being my PhD viva examiners. I am also thankful to Prof. Peter Diggle for providing suggestions for a future use of the methods we devised. Special thanks to all my PGR friends who made this journey meaningful and enjoyable. I would also like to thank the staff of the School of Mathematics at the University of Leeds for making things easier and quick for us.

Last but not least, I would like to thank my mother Saliha, my father Bayram and my sisters Betül and Tuba, and all my family members and friends for their unconditional and emotional support during my 7.5 years being far from them. Despite the distances, I always felt their presence as they are beside me. Finally, I would like to thank my wife Ayşenur who joined me at the middle of this journey and gave me an incredible support. Words are not enough to describe her kindness and patience. I will always be in debt to her.

Abstract

We study the Voronoi tessellation-based lifting scheme in two-dimensional regions where the spatial data is available in a finite and bounded two-dimensional region. The lifting scheme is a second-generation wavelet method that is used for the analysis of spatial data which we model as being an underlying 'true' surface corrupted by noise. On the other hand, Voronoi tessellation is a standard technique to partition the space into smaller sub-regions called Voronoi cells that are used as an ingredient in the lifting scheme.

We investigate the statistical properties of Voronoi cells for homogeneous Poisson points in the infinite plane and bounded regions. The properties are the cell area, perimeter, and the number of cell edges. Our findings show that the distributions of cell properties differ substantially when boundaries are imposed. These differences are affected by proximity.

We emphasize the consequences of the boundaries on the Voronoi cells, and we devise a method that treats the spatial data in the finite region as if it is a subset of a larger region or an infinite plane. This approach predicts the true cell area that is actually clipped by a boundary line using regression-based models. The models are updated for general data cases, and have an overall promising performance.

Lifting scheme uses the features of Voronoi tessellation and the information obtained from the Voronoi cells. The ultimate goal of this thesis is to implement the devised method, which adjusts the cell area near boundaries, into the lifting scheme framework and compare its performance to the standard approaches. Various configurations are considered; standard and proposed weight methods, noisy test functions with different spatial characteristics, and randomly distributed, regular, and clustered point patterns. The proposed approach over-perform the existing options and even gives better performance over the standard spatial prediction techniques such as kriging in certain cases.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

In this thesis, we study Voronoi tessellations and the lifting scheme, and how these two topics combine in situations where the estimation of an underlying function from noisy spatial data is disrupted by artificially imposed boundaries. The idea of Voronoi tessellation is the division of the space into smaller sub-regions called Voronoi cells, and the lifting scheme is used for denoising irregularly spaced data in multidimensions. The focus of this thesis is to study the Voronoi tessellation-based lifting scheme in two-dimensional space, and investigate what happens if the infinite plane is disrupted by a boundary.

When the spatial data is constrained by the boundaries, there are certain limiting circumstances since the boundary act as a cutoff point of the data. Boundaries also change the shapes of the Voronoi cells. Hence, it is important to investigate and understand the effects of the boundaries on the Voronoi cells. Furthermore, we propose methods that behave as if there is an infinite plane which the bounded region is a subset of. For the spatial data observed within a bounded region, this approach has an implicit assumption that the bounded region in which we observe data is actually a subset of a larger region or an infinite plane. Therefore, the data in a finite region is treated as if there is no boundary.

Voronoi tessellation has a wide usage in many disciplines as well as its links to lifting scheme. The lifting framework in (Jansen *et al.*, 2009) uses the cell area and the neighbourhood structure provided by the Voronoi tessellation of data locations. The lifting scheme is a denoising method for irregularly spaced data and has advantages over conventional wavelet methods, and other well-known spatial prediction

methods. The flexibility and applicability of lifting on general data situations is one of its key strengths. Also the theoretical properties of lifting allow it to deal with functions that are smooth or with discontinuities, or even in the case of uncertainty of either cases.

1.2 Voronoi tessellation

Voronoi tessellation is a standard space subdivision method. In one dimension, the real line is divided into intervals, whereas the two-dimensional space is divided into non-overlapping convex cells or polygons in two-dimensional case, and three dimensional case is also possible that the partitions are referred to as the polyhedron. In this thesis, the particular focus is on the two-dimensional case where we investigate the statistical properties of *Voronoi cells* in the absence and presence of boundaries. There is a vast literature on Voronoi tessellation regarding its theoretical aspects and its applications to many different areas which will be discussed in Chapter 2.

Consider a set of n finite number of points $x_1, x_2, \ldots, x_n \in \mathbb{R}^2$ within some finite region $\Omega \subset \mathbb{R}^2$ where Ω is a suitable region that contains all the points, Voronoi tessellation subdivides the two-dimensional Euclidean space into a collection of non-overlapping convex polygons or mosaics $V = \{V_i; i = 1, \ldots, n\}$ called *Voronoi cells*. This is done by associating each point x_i with all the closest points x in that space based on the Euclidean distance. Each Voronoi cell V_i associated with the point x_i is defined as

$$V_i = \{x \in \mathbb{R}^2 \big| \|x - x_i\| \leq \|x - x_j\| \quad \text{for} \quad j = 1, 2, ..., i-1, i+1, ..., n\} \tag{1.1}$$

where $\|.\|$ denotes Euclidean distance. Each cell V_i is defined to be that segment of Ω which is closer to the corresponding point x_i than any other point. The edges of the Voronoi cells may consists of line segments, half lines or infinite lines. We consider the line that separates two cells has nearly zero thickness hence the intersection of two cells $V_i \bigcap V_j$ is nearly nonempty. Therefore, Voronoi cells satisfy $\Omega = \bigcup_{i=1}^{n} V_i$ and $\bigcap_{i=1}^{n} V_i = \emptyset$ up to a measure zero and the statement can be generalized for $d-$dimensional cases as explained in Okabe *et al.* (2000) and Møller (2012).

1.3 A motivating example for boundaries

Given a set of points, Voronoi tessellation can be constructed based on the locations of the points. A simple example is illustrated in Figure 1.1. The left plot shows

the Voronoi tessellation of randomly distributed points in a continuous region but only a part of the region is shown. The solid lines continue outside the window based on the locations of other points. The geometric structure created by Voronoi tessellation is non-overlapping convex polygons or the Voronoi cells where each cell edge is the perpendicular bisector between two points.

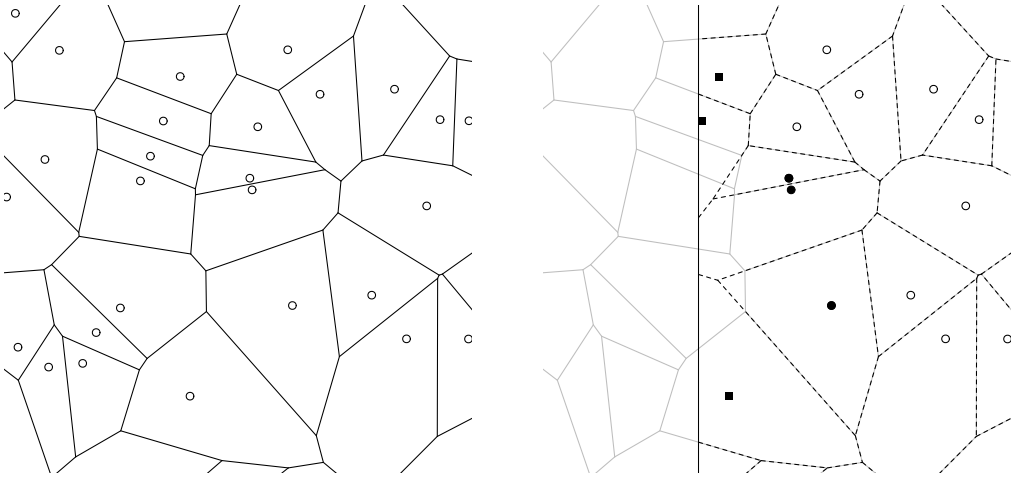


Figure 1.1: A zoomed in version of Voronoi tessellation of points with $\rho = 200$ in an infinite plane. Gray lines are the original tessellation lines before any boundary is used. Dashed lines are the tessellation lines after the vertical solid line is imposed as a boundary to the points on the right side. Cells with the (■) had an intersected the boundary, and cells with (•) did not intersect the boundary but also had a changes in their shapes. Gray and black circle points are the points of remaining cells.

In this thesis, we are also interested in the cases where the spatial region is disrupted by a boundary. The right plot shows an example of this situation. Voronoi tessellation of the same set of points subject to a boundary line is given. The objective of this illustration is to demonstrate the consequences of the boundaries. Consider the vertical solid line is an imposed boundary, and the Voronoi tessellation of points on the right side is performed again. Changes on the shape of cells are observed for the cells that are closer to the boundary and most of the cells far from the boundary remains the same.

There are interesting features of the cells in the presence and absence of the boundary. The cells with a (■) point have vertices on the boundary line and the boundary line clipped a part of the cell at the top. The remaining two cells with (■) are both clipped and expanded after the boundary is imposed. More importantly, although some cells with (•) point did not have a vertex on the boundary, their shapes are

also affected by the boundary. Therefore, a statement 'only the cells that have a vertex on the boundary are likely to be affected by the boundary' be inefficient.

The boundaries and the boundary types is an important context in spatial statistics and a clear explanation of their functionality and effect is necessary. A boundary is a real or artificial line or point based on the dimension of the space that separates two things or acts as an end point of an existing space. In Figure 1.1 (right), we visualised the Voronoi tessellation of points and draw an artificial boundary line. The vertical line acts as an end point of the tessellation and has an effect on the existing structure. However, the dashed lines can also be considered as the boundaries of polygons which are perpendicular bisectors that separates two neighbour points. In this thesis, we focus on boundaries for simulated data, and real data examples where the boundary is an artificially imposed boundary or a study region.

In the real life, physical boundaries occur due to the existence of a natural factor, for instance, a coastline, river, or the starting point of a desert. The political boundaries are another example of real boundaries such as the border between two states or countries. Occasionally, the political boundaries are determined based on the natural factors such as a river may be referenced to separate the states. Also, a smaller sampling regions may be defined on a large geographical region to study the features of the plants or the soil. The defined sampling region may be a suitable rectangular window and acts as the boundary. The natural boundaries have different effects compared to the boundaries such as the sampling regions. For instance, proximity to a natural boundary may have a negative effect on the fertility of the soil or the existence of the trees. However, if a rectangular region is sampled from a larger region, the observations in the sampled region are related to the ones outside the boundary, hence it is important to consider ways to understand and reduce the bias near the boundaries. Throughout the thesis, we give examples of these kind of induced boundary types, discuss the issues that may occur, and propose ways to reduce the boundary effects.

1.4 Spatial point patterns and Poisson point process

A spatial point pattern is a set of randomly located points on a specified region that is designated as the two-dimensional Euclidean space in this thesis but one and three-dimensional cases are also likely. The locations of trees in a forest, cell

nuclei of a tissue, earthquake centres, bird nests, particles, and the positions of the galaxies in the universe are the examples of point patterns. The locations of points are also referred to as the *events*, and the information carried at the locations are called *marks* such as the tree diameter. These type of data are called the *marked point pattern* data.

On the other hand, the point processes are stochastic mechanisms, and are useful to understand, describe and analyse the point patterns. It is mostly used for the identification of the short-range relationship between the points that characterizes whether a spatial randomness, regularity, or clustering exist. Our aim is not to give a complete treatment about point process statistics, it is rather to explain the methods which are useful to implement the Voronoi tessellation and the lifting framework that are the main focus of this thesis.

The point patterns are assumed to have an underlying mechanism that can be formulated by point processes. Voronoi tessellation explained in Section 1.2 require points $x_1, x_2, \ldots, x_n \in \mathbb{R}^2$, however, have not mentioned yet whether the points rely on a mathematical concept.

In this thesis, we consider several types of geometrical structures of the point patterns that are realisations from point processes. Hence, we use the point processes to generate point patterns that obey the parameters of certain point processes. A point process $\mathcal{N}$ is described as a random counting measure or a function that is operating on sets in (Illian *et al.*, 2008). For instance, for any bounded region $B \in \mathbb{R}^2$, $\mathcal{N}(B)$ stands for the number of events or points within the region B.

One of the simplest but a fundamental point process is the homogeneous Poisson process whose realisations exhibit complete spatial randomness. A point process $\mathcal{N}$ is a homogeneous Poisson process if it has the following properties:

i The number of points in any bounded region B follows a Poisson distribution with mean $\rho|B|$. In the formal way,

$$\Pr\{\mathcal{N}(B) = n\} = \frac{(\rho|B|)^n}{n!} e^{-\rho|B|}$$

where $|B|$ denotes the area of B,

ii and given n points x_i, those points form an independent random sample with a uniform distribution on B.

The parameter ρ in (i) is called the intensity that refers to the mean number of points per unit area. The second property constitutes the complete spatial randomness of the points. We will use the homogeneous Poisson points essentially in Chapter 2 and occasionally in the other chapters.

1.5 Regular and clustered points

The regularity and clustering of points or events happen towards departure from complete spatial randomness. In the parts of the thesis, we are interested in the usage of examples of regular and clustered points, in addition to the homogeneous Poisson points. The reason of the consideration of regular and clustered points is to see how the core methods we develop throughout the thesis that rely on point patterns are affected by the departure from complete spatial randomness. In other words, the intention is to examine the performance of the methods when the regularity and clustering approaches to its extreme forms.

Regularly spaced and clustered data locations are the two important cases we consider. These point patterns are frequently seen in real life data. Regular and clustered point processes are also called inhibition or repulsion, and clumping or attraction respectively in the literature. These two processes can be expanded to examples where different levels of regularity or clustering is observed such as departures from homogeneity to highly clustered and regular points.

There is a convenience of generating point patterns that are examples of clustering and regularity and control this process with a single parameter. The saturation process by Geyer (1999) permits both the attraction and repulsion processes for spatial data. Geyer's saturation process is an extension to the Strauss process Strauss (1975) that is a method for repulsion within a fixed radius. The saturation process of Geyer modifies the Strauss process by constraining the overall contribution of each point to a maximum value (Goldstein *et al.*, 2015). The probability density of the saturation model is

$$f(\mathbf{x};\beta,\gamma,r,s) = c\beta^n \prod_{i=1}^{n} \gamma^{\min(\sum_{j\neq i} \mathbb{1}_{\|x_i - x_j\|\leq r}, s)} \tag{1.2}$$

where c is a constant, β, γ, r, s are the parameters, $\sum_{j\neq i} \mathbb{1}_{\|x_i - x_j\|\leq r}$ denotes the number of neighbours of the point x_i within a distance r. The saturation threshold $s \geq 0$ prevents each term in the product from being larger than γ^s and hence the product is never larger than γ^{sn}. This prohibits the attraction from becoming very strong,

which discourages highly clustered patterns. If $s = 0$, the model becomes a Poisson point process, if $s > 0$, the interaction parameter γ can take any values such that $\gamma > 1$ indicates attraction or clustering and $\gamma < 1$ indicates repulsion or inhibition. Also, in the case of $s = \infty$, the model reduces to the Strauss process.

This model will also be used for generating realisations of regular and clustered point patterns as is aimed for the Poisson point process. The method is useful in terms of controlling the departure from homogeneous pattern with a single parameter γ. Hence, it allows flexibility to decide on the degree of clustering and regularity which we are especially interested in.

1.6 Lifting scheme

The lifting scheme transforms a noisy function at irregularly spaced data locations into the lifting domain where the data is represented by a set of coefficients. Then the coefficients are modified by a thresholding rule that aims to separate the noise and preserve the important features in the data such as the step changes or spikes in a function. The inverse transform of the thresholded coefficients gives an estimate of the true function. We adopt the *lifting one coefficient at a time* technique proposed by Jansen *et al.* (2009), which iteratively transforms the data into coefficients in the lifting domain by starting with localised or fine-scale details and working up to broader or coarse-scale patterns.

Within the stages of the lifting transform, the weights, which are obtained from the areas of Voronoi cells are used. This is the part where the Voronoi tessellation and the lifting scheme merge. Voronoi tessellation is used for the detection of the neighbourhood of the points, and the cell area is used for the calculation lifting coefficients. The standard choice for the weights is using the observed cell area that is calculated using the boundaries. We alternatively use adjusted cell area as the weights that are attained from the a method we will devise later.

The noise is a usual and an unavoidable issue which happens during data collection or the recording of the data by measurement tools. In a general sense, noisy data in real life case may be an image of a person or an ultrasound image that contain noise, or a recorded noisy signal from sound. The underlying true patterns are the true functions in this case which we would like to estimate by separating the noise from the data. In this thesis, we are more interested in developing some aspects of existing denoising methods and propose alternative ways that can improve the estimation of the underlying true patterns. Therefore, we use the two-dimensional

analogues of some well known functions and artificially add noise. The functions which are treated as the *true functions* are explained in Section 6.1. The noise-added test functions are the noisy functions which we apply the lifting scheme to separate the noise from the data.

Function estimation using the lifting scheme has the standard model $y_i = f(x_i) + \epsilon_i$ where we consider $\{x_i\}_{i=1}^n \in \mathbb{R}^2$ as the irregularly spaced data locations, y_i are the observed noisy data, $f(x_i)$ are the values of some underlying true function corrupted by independent and identically distributed Gaussian noise such that $\epsilon_i \sim N(0, \sigma^2)$. However, we are interested in the estimation of the function f_i when only the noisy observations y_i are available. In this situation, we use the lifting scheme to obtain an estimate of $\hat{f}_i$.

Voronoi tessellation-based lifting scheme aims to estimate the underlying true function f_i using the Voronoi tessellation cell area as weights during the process. Lifting is a linear transformation that transforms the observed noisy data y_i into the lifting domain, and the transform can be represented by a transform matrix $\mathbf{L}$. Hence, the resulting representation of the noisy data can be shown as $\mathbf{d} = \mathbf{L}\mathbf{y}$ where the vector $\mathbf{d}$ consists of lifting coefficients. These coefficients in $\mathbf{d}$ are usually a sparse representation of the observed data $\mathbf{y}$ that explains the data by a small number of non-zero coefficients. The zero or small coefficients indicate small deviations in the data that are due to the noise and the larger coefficients are attributed to the real features in the function. The transform matrix $\mathbf{L}$ is independent of the observed data values and only depends on the data locations hence it has a reusable feature for other data observed at the same locations. The process of estimation of $\hat{f}_i$ includes the adjustment or thresholding of the coefficients in $\mathbf{d}$ that aims to shrink the small coefficients to zero, and keep the larger ones, obtaining a vector of adjusted coefficients $\mathbf{d}'$. Finally, the inverse transform is performed on the thresholded coefficients to estimate the underlying true function as $\hat{\mathbf{f}} = \mathbf{L}^{-1}\mathbf{d}'$ that is separated from noise. The transform can be inverted by both using the inverse of the transform matrix $\mathbf{L}$ or following the steps of the lifting in an inverse way. The steps of the lifting scheme will be explained in detail in Chapter 5.

1.7 Thesis structure

Voronoi cells have geometrical properties such as the cell area, perimeter, number of edges, interior angles of cells etc., that have been widely studied. Currently the

literature include the analytic derivation of the mean cell properties and numerical approximations using appropriately selected parametric distributions based on computer experiments for particular point pattern types.

We extend the study on the statistical properties of Voronoi cells by considering Voronoi cells in the infinite plane and finite regions using different types of boundaries, and demonstrate the differences in the cell properties in the presence of boundaries in Chapter 2. This separate part of the study contributes significantly to the Voronoi tessellation literature since little attention has been given to how these properties change when a boundary is imposed. A better understanding of the statistical properties of Voronoi cells in bounded regions is especially important due to the usage of such properties on the next topic that we focus on, the lifting scheme.

The consideration of various boundary types and their effects is also important. A convex hull of points is the smallest polygon that includes all points and can be drawn for any type of point patterns. On the other hand, a suitable window i.e., a rectangular window can be used as the boundary. However, different boundary types are likely to have distinct impact on cell properties. For instance, consider a set of uniform random points $X = \{x_1, \ldots, x_n\}$ generated in a unit square. Hence, the finite region is defined as $\Omega_u = [0, 1]^2$ which we can consider as a boundary. Then we perform Voronoi tessellation of points and record the cell area for each cell V_i. If the convex hull of points is used for the same set of sampled points, unless there are points precisely at the corners of the unit square, the convex hull will be a subset of the unit square such that $\Omega_c \subset \Omega_u$. Therefore, the observed cell area will be different for the cells close to the boundary with the usage of these two types of boundaries. Although the choice of the boundary could be expanded, we consider a limited number of boundary types that demonstrates the important consequences caused by the boundaries.

We conduct a simulation study in Chapter 2 to investigate the statistical properties of Poisson Voronoi cells in the infinite plane, and when unit square and convex hull boundaries are imposed. The chapter discusses the exploratory analysis of the cell properties for different boundary cases, the fitting of parametric distributions for cell area, perimeter and number of cell edges, and explores the changes in cell properties when the boundaries are imposed. It also presents the results for different intensities of points. Findings in Chapter 2 open an important discussion about the reduction of issues caused by the boundaries which is discussed in Chapter 3.

Another major contribution of this thesis is to propose a method that treats the data in a finite bounded region as if it is a subset of a larger region or an infinite plane. This part of the thesis aims to reduce the unfavourable consequences of the boundaries and is a transition between the Voronoi tessellation study and the lifting scheme. In Chapter 3, we use the data obtained from the simulation in Chapter 2 that is a large data set containing many variables which are characteristic information such as the cell area (infinite plane, unit square, convex hull), perimeter, number of cell edges, cell type (interior, edge, corner), distance from the boundary etc., for 10^6 cells that are sampled from individual realisations. Using the remaining variables, we fit regression models to predict the *true cell area* that is the cell area in the absence of boundary. This is done by dividing the data obtained from simulation into training and validation sets, fitting regression models to the individual training sets which we call *base models*, and using an ensemble approach in the prediction of cell area in the validation set. We also identify influential points that cause large error in the validation set and add them to the training sets, and fit *augmented models* that are capable of predicting observations that are hard to predict. Evaluation of the performances of the base and augmented models, and implementation of this approach into the lifting framework is an important contribution of the thesis.

Chapter 4 investigates the application of the method in Chapter 3 for general data situations such as the departure from homogeneity. The models are used for the area prediction of Voronoi cells based on regular and clustered point patterns. The preliminary objective is to see the performances of the models on the violation of the homogeneous patterns. Furthermore, we present a way to update the models hence they perform more efficiently for the regular and clustered data cases.

Next, we explain the lifting scheme and how the Voronoi tessellation is used as an ingredient in the method Chapter 5. Non-parametric regression is one of the classical concerns in statistics including the analysis of spatial data. The lifting scheme is a relatively new method introduced by Sweldens (1998), and is an extension of the wavelet methods that are used for function estimation in non-parametric regression using shrinkage schemes (Donoho & Johnstone, 1994; Donoho *et al.*, 1995). While the conventional wavelet methods require equally spaced data with size $n = 2^J$ for some $J \in \mathbb{N}$, the lifting scheme relaxes such restrictions as being applicable to any type of data structure regardless of the size n. We rely on the framework described in Jansen *et al.* (2009) when using Voronoi tessellation-based lifting scheme in two-dimensions.

Chapter 5 is a background chapter for lifting scheme in two dimensions with some discussion on wavelet methods which the lifting scheme is built upon. The technical details about the steps of the lifting scheme and illustrative examples are given. Thresholding methods are also explained.

As mentioned previously, the Voronoi cell area is used as the weights in the lifting scheme. These weights determine the calculation order of the coefficients in $\mathbf{d}$, and the update of function values in the stages of lifting. Hence, the choice of the weights has a direct effect on the estimation of the underlying true function. For spatial data in a finite region, we consider observed weights such as the calculated cell area using boundaries. More importantly, we use the predicted cell area from the models we developed, and use the predicted area as weights. Ultimately, we consider various weight methods, and evaluate their performances especially aiming our proposed method to reduce the boundary effects and improve the function estimation. Since we target our new method to be used in general data situations, we take into account various data location structures, such as randomly distributed, regularly spaced and clustered points and test the method using numerous test functions that have different spatial characteristics.

Chapter 6 explains the types of weight methods, and two-dimensional test functions we consider throughout the thesis. Then it presents the lifting results for simulated homogeneous data locations with different configurations of weight methods. We focus on the local information to check and identify the differences between weight methods in the estimation of test functions. Lifting results for regular and clustered data from simulations, and real data sets are presented in Chapter 7 where we evaluate the weight methods in the case of departure from homogeneity and use the suggested method for the real data examples. The weight methods we consider in this chapter also include the area prediction with local intensity-based scaling since we use regular and clustered points that have local features.

Finally, we give an overall summary of the thesis, discuss the meaning and importance of our findings, and talk about the potential future work in Chapter 8.

Chapter 2

Statistical properties of Voronoi tessellations in bounded regions

2.1 Objective of of the study

Voronoi tessellation is a standard space subdivision method that has wide application areas such as seismology, astronomy, ecology, meteorology, metallurgy, material science, and architecture. Also, the structures obtained from Voronoi tessellation are used as an auxiliary tool in the analysis of spatial data. For instance, the partitions obtained from the Voronoi tessellation are used as a curvature parameter in spline methods as discussed in Ripley (2005) or as the weights in spatio-temporal analysis methods, and for the intensity estimation and efficient computation algorithms Illian *et al.* (2008).

Although Voronoi tessellation is a wide topic on its own, it is also used in conjunction with other methods such as the lifting scheme which we aim to develop some aspects of in this thesis. The lifting scheme discussed in Jansen *et al.* (2009) uses Voronoi tessellation as a key ingredient in the algorithm. The neighbourhood structure determined from the Voronoi tessellation, and the properties of Voronoi cells such as the cell area are used in the steps of the lifting scheme. Therefore, this thesis aims to give a good understanding of the Voronoi tessellation and investigate the aspects which are important but has not been studied thoroughly.

This chapter discusses the statistical properties of Voronoi tessellations based on homogeneous Poisson points in the infinite plane and in the bounded regions. The description and examples of Poisson Voronoi tessellation is given in Section 2.2. Geometrical and statistical characteristics of Voronoi cells have been investigated

theoretically and numerically for decades and applied to a range of data types. Over the years, properties of the Voronoi cells such as the mean cell area, perimeter, numbers of edges or vertices and vertex angles have been explored. These cell properties are used in different context in many disciplines. Hence, the relevant literature was reviewed and findings are discussed in Section 2.3.

The techniques and approaches used in the previous work provide us guidance how the experiments on point patterns were conducted for particular cases. However, in this chapter, new perspectives are considered which will be a significant contribution to improve the current approaches. Point patterns are generally considered to be in an infinite plane so that Voronoi cells are surrounded by the neighbour cells. This chapter mainly focuses on Voronoi tessellation in two-dimensional space, and explores the characteristics of the cells when boundaries are imposed on the point patterns. Section 2.4 explains the design of the simulation study we perform and Section 2.5 summarizes our results. This chapter is published as a journal article in Gezer *et al.* (2021).

2.2 Voronoi and Poisson Voronoi tessellation

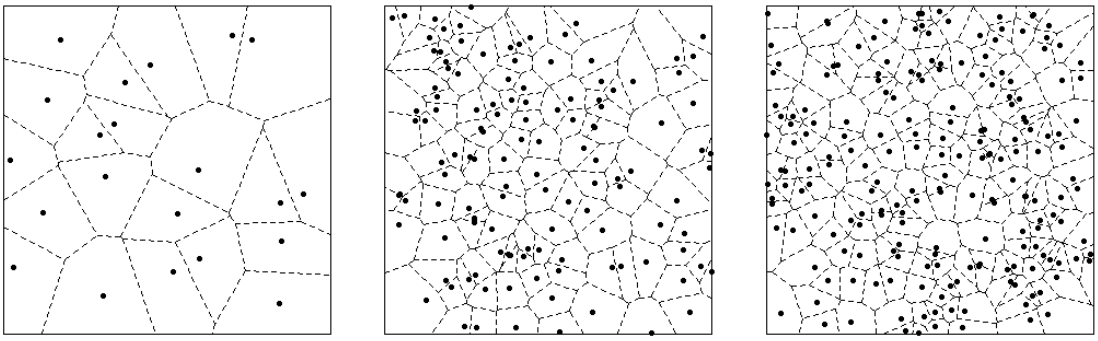
Revisiting the definition (1.1) in Section 1.2, Voronoi tessellation partitions the two-dimensional space into disjoint regions V_i called Voronoi cells, given a set of points $x_i \in \mathbb{R}^2$, $i = 1, 2, ..., n$. Each V_i is associated with a point x_i and cells are determined by the perpendicular bisectors between the point and its neighbours. In the formal way, it can be expressed as

$$V_i = \{x \in \mathbb{R}^2 \big| \|x - x_i\| \leq \|x - x_j\| \quad \text{for} \quad j = 1, 2, ..., i-1, i+1, ..., n\}$$

where $\|.\|$ is the Euclidean distance (Møller, 2012; Okabe *et al.*, 2000).

The formal definition of the Poisson point process is given in Section 1.4. When the number of randomly generated points n in $\mathbb{R}^2$ follow a Poisson distribution with a finite and constant intensity $\rho > 0$, this standard point pattern is called a homogeneous Poisson point process. Therefore, the Voronoi tessellation based on homogeneous Poisson points is called the Poisson Voronoi tessellation.

Consider the set of points $\{x_i\}_{i=1}^n \in \Omega$ and let $\Omega \subset \mathbb{R}^2$ be a convenient region in the space which contains all the points. Therefore, for n homogeneous Poisson points we shall denote $n \sim Po(\rho|\Omega|)$ where $|\Omega|$ is the area of the region Ω. The region may be a suitable rectangle, the convex hull of the points, or some other



(a) (b) (c)

Figure 2.1: Examples of Voronoi tessellation of points with intensity (a) $\rho = 20$, (b) $\rho = 100$ and $\rho = 200$ in a unit square bounded region $\Omega = [0, 1]^2$.

specified region. Realizations of Poisson Voronoi tessellations (PVT) with intensities $\rho = \{20, 100, 200\}$ respectively in a unit square $\Omega = [0, 1]^2$ are given in Figure 2.1 where solid points represent the generated points in each realization. Points are generated uniformly in a unit square and the number of points follows a Poisson distribution with mean ρ. Perpendicular bisectors between points are shown with dashed lines which generate the cells. These lines are called the *cell edges* and two points are considered as neighbours if they have a common edge.

2.3 Background and previous work

The importance of space subdivision methods to investigate spatial splines, and examples of different spatial point patterns for both simulated and real data to relate the subject to the estimation of distributions of the locations within a region using the Voronoi tessellation are discussed in Ripley (2005), Illian *et al.* (2008) and Okabe *et al.* (2000). The Voronoi tessellation has been applied in different sciences such as in seismology (Schoenberg *et al.*, 2009) to find the distribution of the cell areas of Voronoi tessellations based on the locations of earthquakes in Southern California; astronomy (Icke & Weygaert, 1987; Ramella *et al.*, 2001; Yoshioka & Ikeuchi, 1989) to discover how galaxies are distributed in space; to investigate the conditions of the habitat of animals when they are establishing territories (Tanemura & Hasegawa, 1980); in agriculture for maximal weed suppression to plant crops (Fischer & Miles, 1973) and to study atomic crystals (Mackay, 1972), liquids (Finney, 1970), glasses (Luchnikov *et al.*, 2000), and wireless networks (Baccelli & Błaszczyszyn, 2001; Koufos & Dettmann, 2019). An application of constrained Voronoi tessellation is used in micro-structure modeling (Xu & Li, 2009) where a

new space subdivision method is introduced using reverse Monte Carlo based on conditions such as moving the randomly placed points until their geometric features obey a particular distribution.

Preliminary studies (Gilbert, 1962; Meijering, 1953) investigated the mean of interface area, edge length, and number of faces for an aggregate of crystals that are considered as the points. Let N denote the number of cell edges, P the perimeter and A the area of a Voronoi cell. Meijering (1953) presented the initial theoretical results of the Voronoi cells and showed that the mean cell area perimeter and the number of cell edges are

$$E(N) = 6, \quad E(P) = 4\rho^{-1/2}, \quad E(A) = \frac{1}{\rho} \tag{2.1}$$

where ρ is the unit intensity of the points as explained in Section 1.4. Even though the distributions of the cell properties has been investigated empirically, no exact representative distribution has yet been found. However, various authors have recommended that appropriate approximations can be made using the Gamma distribution with appropriately chosen parameters.

Kiang (1966) proposed an appropriate fit for the length of Voronoi line segments in one-dimension, the area of cells in two-dimensions and the volume of polyhedrons in three-dimensions. The length distribution of the Voronoi segments in one dimension is derived analytically and Monte Carlo experiments are performed to estimate the distribution of areas and volumes in two and three-dimensional spaces. A fixed number of points are randomly distributed on a square lattice and the cell areas are recorded for all points. To avoid boundary effects, the coordinates of the points on the opposite end of the region is translated. This process is repeated independently for many realizations to increase the sample size.

Standardized measures are obtained through (2.1). For instance, the standardized cell area and perimeter are derived as $s = A \times \rho$ and $p = \sqrt{\rho}/4 \times P$, therefore, $E(s) = E(p) = 1$ for the standardized measures and number of edges $E(N)$ is taken as the same since it is independent of ρ, (Crain, 1978). This makes the comparison of the parameter estimation results accurate when different studies used different ρ. The two-parameter gamma distribution has density function

$$f(s|b,c) = \frac{b^c}{\Gamma(c)} s^{(c-1)} e^{-bs}, \qquad 0 < s < \infty, \qquad b, c > 0 \tag{2.2}$$

where b is the shape parameter, c is the rate parameter and s is the standardized

cell area. Kiang (1966) found that (2.2) can explain the observed histograms of the standardized cell areas. The cell area A is standardized and denoted as s instead of a due to simplicity of the expression of the Gamma distribution with three parameter which we will use later.

Analytic derivation of the distribution of cell area for Voronoi tessellation cells in two dimensions in Weaire *et al.* (1986) approximated the shape parameter as $b = 3.63$. Both in Kiang (1966) and Weaire *et al.* (1986), it is assumed that $b = c$ based on the similarity of the parameter estimation results.

Another study by Kumar & Kurtz (1993) was based on Poisson-Voronoi tessellation with intensity $\rho = 100$ were one of the points was always placed at $(0.5, 0.5)$ and the remaining points are randomly placed within a unit square. Then, the distributions of the area, perimeter, length of the each side of the cell and the numbers of the sides were investigated for the centered point. Kumar & Kurtz (1993) also found that the two-parameter gamma distribution gave an accurate fit for the observed histograms of the cell properties. The shape and rate parameters are estimated $b = 3.7176$ and $c = 3.7174$, respectively.

Hinde & Miles (1980) carried a simulation based on homogeneous Poisson point process with intensity $\rho = 100$ and recorded the properties of the point that is closest to the centre of the unit square region based on many independent simulations. The observed shape of cell area and perimeter distributions suggested a uni-modal density function dominated by an exponential and controlled by a simple power in $(0, \infty)$. Therefore, the three-parameter generalized gamma distribution in (2.3) is used.

The three-parameter gamma distribution by Stacy (1962) has density function

$$f(s|a, b, c) = \frac{ab^{c/a}}{\Gamma(c/a)} s^{(c-1)} e^{-bs^a}, \qquad 0 < s < \infty \qquad a, b, c > 0. \tag{2.3}$$

Note that (2.3) is a two-parameter gamma distribution as in (2.2) when the shift parameter $a = 1$. Parameters of the generalized gamma distribution should ideally be estimated by maximum likelihood estimations as stated in (Stacy & Mihram, 1965). However, two other computationally simpler methods gave very similar results.

Statistical distributions of the Voronoi cells from homogeneous Poisson points in two and three dimensions are studied by Tanemura (2003) in an extensive manner. A set of homogeneous Poisson points with intensity $\rho = 200$ is generated in a

two-dimensional space where the unit square is used as the sampling region. To construct an independent sample of Voronoi cells, a point shifting method is used. The procedure is based on random selection of a point and to move it to the centre of the sampling region $\Omega = [0,1]^2$ that is $(0.5, 0.5)$ while the relative positions of the other points are kept. This procedure will be explained in Section 2.4.

The intensity of the points and the numbers of the realizations were modified in Tanemura (2003) to check whether any difference exists but the results for the standardized cell properties were very similar. Three-parameter gamma distribution is fitted for the observed histograms using the maximum likelihood estimations. The log-likelihood function $l(a, b, c|s)$ for the standardized areas is derived as

$$\begin{aligned} l(a,b,c|s) &= \log \prod_{i=1}^{r} f(s_i|a,b,c) \\ &= \sum_{i=1}^{r} \log f(s_i|a,b,c) \\ &= \sum_{i=1}^{r} \{\log \frac{ab^{c/a}}{\Gamma(c/a)} + (c-1)\log s_i - bs_i^a\} \\ &= r \log \frac{ab^{c/a}}{\Gamma(c/a)} + (c-1)\sum_{i=1}^{r} \log s_i - b\sum_{i=1}^{r} s_i^a. \end{aligned} \tag{2.4}$$

Tanemura's approach was to approximately maximize the log-likelihood function given in (2.4). The same distribution for the perimeter, as well as for discrete measures such as the numbers of the edges is used even though the generalized gamma distribution is a continuous distribution. Estimated parameters in Table 2.1 shows that $a \neq 1$, and b and c are not as similar in multi-dimensions as is seen in the previous work that relied on two-parameter gamma distribution.

Dimension	a	b	c
1	1.0	2.0	2.0
2	1.07950	3.03226	3.31122
3	1.16788	4.04039	4.79803

Table 2.1: Generalized gamma distribution parameter estimates for dimensions $d = 1, 2, 3$ in Tanemura (2003).

Arvanitakis (2014) overlaid the density lines of the estimated parameters of the generalized gamma distribution by Tanemura (2003) over the two-parameter gamma

density fitted by Tanemura (2005) for the same data. However, two densities did not show a meaningful difference on fitting the histograms.

In a recent paper, Koufos & Dettmann (2019) conducted research on the distribution of bounded Poisson Voronoi cell areas. An integral based method from Brakke (1987) is extended to calculate the mean cell area. The method is to consider PVT over a quadrant in two-dimensional space and calculate the mean cell area for the points located at the edge and corner of the region, and in the bulk which is interior part of the region. The parameters are estimated by two-parameter gamma distribution in (2.2) from the first two moments of the cell area as

$$c = \frac{E(s)^2}{Var(s)} \quad \text{and} \quad b = \frac{E(s)}{c}.$$

The mean and variance for area of the cells located at the corner, edge and bulk are given in Table 2.2. A corner cell is a Voronoi cell that has more than two vertices located on the boundary, the edge cell has exactly two vertices, and a bulk cell has no vertices on the boundary. The standardized mean cell area is found less than 1 at the corner and edges of the region. Also, the parameter estimates are different in these two cases. Under these considerations, computer simulations verify the results from the integral based method.

Type	$E(s)$	$Var(s)$	$\hat{b}$	$1/\hat{c}$
Corner	0.36351	0.10567	1.25052	0.29069
Edge	0.61082	0.17198	2.16935	0.28157
Bulk	1	0.28018	3.56918	0.28018

Table 2.2: Mean cell area, variance and parameter estimations for two-parameter gamma distribution by Koufos & Dettmann (2019) for corner, edge and bulk cells.

In this section, a discussion of previous studies and the directions that they followed are given. The main purpose of the previous work was to find suitable distributions to estimate the properties of Voronoi cells. Analytic derivations of the cell properties are verified as the performance of programs for statistical computing increased. In the remainder of this chapter, investigation of Voronoi cells in two dimensional space will be extended to the cases of regions with imposed boundaries.

2.4 Design of the simulation

In the entire experiment, the intensity of the points is set to $\rho = 200$ for $r = 10^6$ realizations. To generate independent samples of Voronoi cells, a technique briefly mentioned in Section 2.3 is used. We generate uniform random points with the specified intensity and perform the Voronoi tessellation. Next, one cell is selected at random and moved to the centre of the region. The relative positions of the other points are kept the same using periodic boundary conditions. Finally, the properties of the selected cell are calculated. The procedure is repeated for a new set of points, for a total of $r = 10^6$ realizations. An illustration of this the point shifting process for a randomly selected point is given in Figure 2.2 for $\rho = 100$ for visual clarity.

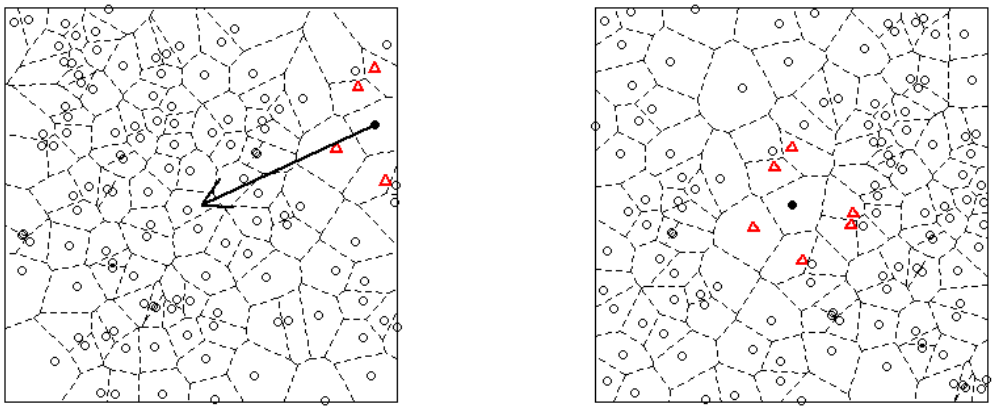


Figure 2.2: A randomly sampled cell with solid black point along with its neighbour points as red triangles is shown (left). The arrow shows the direction of the sampled point to the centre of the region where all other points keep their relative positions. Cell moved to the centre of the region (right), where new points from the opposite end of the region form the new neighbourhood.

It is known that the properties of Poisson Voronoi cells do not change by conditioning on the location of a point in an infinite plane for the homogeneous Poisson points (Koufos & Dettmann, 2019). This first step of the experiment focuses on PVT in an infinite plane and adopts the shifting illustrated in Figure 2.2 to generate independent samples of cells in the infinite plane. Additionally, two different boundary cases (the unit square and the convex hull of points) are used as in Figure 2.3 to investigate how the imposed boundaries affect the cells.

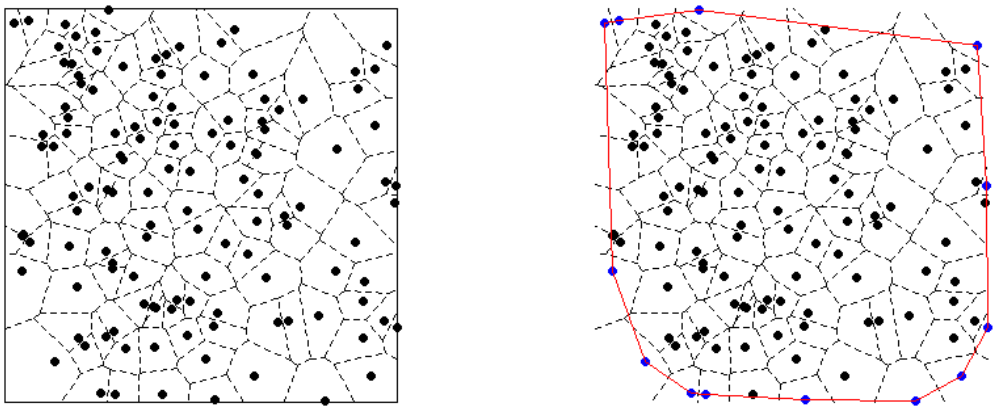


Figure 2.3: Voronoi tessellation of points with $\rho = 100$ bounded with a unit square (left). The convex hull of points shown with red lines and points which are on the convex hull are in blue (right).

2.5 Results

This section presents the simulation results based on the three cases; infinite plane, unit square, and convex hull bounded cells, fitting of parametric distributions, and a discussion of the effects of the boundaries on the cell properties. We present the results for three cases separately.

2.5.1 Voronoi tessellation in the infinite plane

In the first part of the experiment, Voronoi tessellation of points in the absence of the boundary is considered. Initially, homogeneous Poisson points with $\rho = 200$ are simulated within the unit square region $\Omega = [0, 1]^2$ domain, and a point is sampled randomly. Then the sampled point is moved to the centre of the region $(0.5, 0.5)$ by keeping the relative positions of all other points as described in Figure 2.2. Finally, the Voronoi tessellation of the shifted points is performed, and the cell properties of the sampled point is recorded. This temporary process allows us to eliminate a possible boundary effect on the sampled cell. Therefore, the recorded cell property reflects as if there is no boundary.

Histogram of the cell area of one million Poisson Voronoi cells each of which is randomly sampled from one million realisations of point patterns is presented in Figure 2.4 (left). A uni-modal left skewed distribution is observed in the histogram. Mean cell area over pixel bins is sown as a image plot (right) that summarizes the information over the two-dimensional surface. The region is divided into equal size of bins and the mean cell area is visualized based on the cell area observed at the

sampled points in each bin. The initial positions of the sampled points before the shifting process are used when creating the image plots.

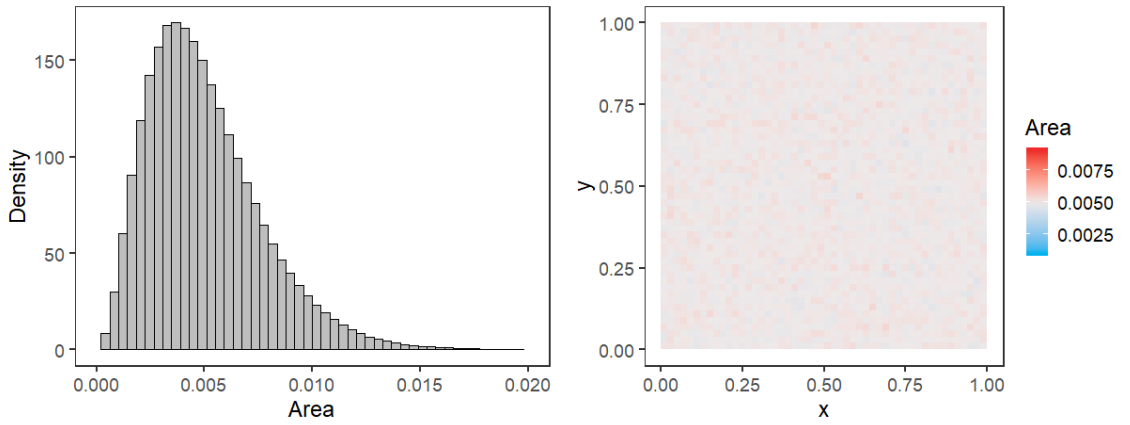


Figure 2.4: Area of Voronoi cells in the infinite plane (left). Surface plot of cell area in the infinite plane (right). The unit square is divided into a 50×50 grid of bins and the mean of the observations falling in each bin is plotted.

The histogram and the surface plot summarizes the results from all realizations in the experiment. Hence, each pixel bin of the image contains a number of data points from different realizations. The surface plot therefore shows the spatial patterns of cell area at different locations. We observe an unstructured pattern in Figure 2.4 since the data is based on the cells in the infinite plane where the unstructured pattern is expected. However, there are interesting spatial features when the boundaries are imposed which will be discussed later. Although the right plot in Figure 2.4 appears to be a trivial example, it is included since the equivalent plots will be presented in the following sections. The surface plots in the results section and throughout the other chapters of the thesis are created using the functions in the `ggplot2` package by Wickham (2016). The observed unstructured pattern in Figure 2.4 implies that the cell area takes value around the expectation $1/\rho$, and does not change over the infinite plane, that supports the statement that the characteristics of Poisson Voronoi cells are independent of the location.

Similarly, the perimeter of each cell in one million realizations is calculated and the histogram and the surface plot is shown in Figure 2.5. The histogram of the perimeter (left) is less skewed and even has a symmetric-like shape compared to the area. However, the observed cell perimeter over the infinite plane (right) shows unstructured characteristics as in the cell area.

The bar chart of the number of Voronoi cell edges along with the surface plot is shown in Figure 2.6. It is likely for the cells to have 6 edges as expected and no anomalies over different parts observed. Numbers of the cell edges and their occurrences in the infinite plane is shown in Table 2.3 which ranges from 3 to 15 and has very small number of observations for $N > 9$.

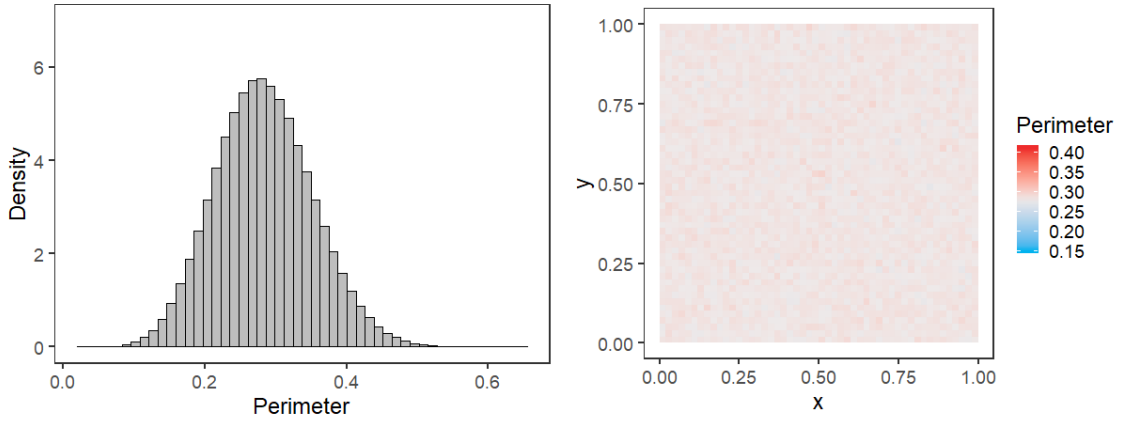


Figure 2.5: Perimeter of Voronoi cells in the infinite plane (left). Surface plot of cell perimeter in the infinite plane (right).

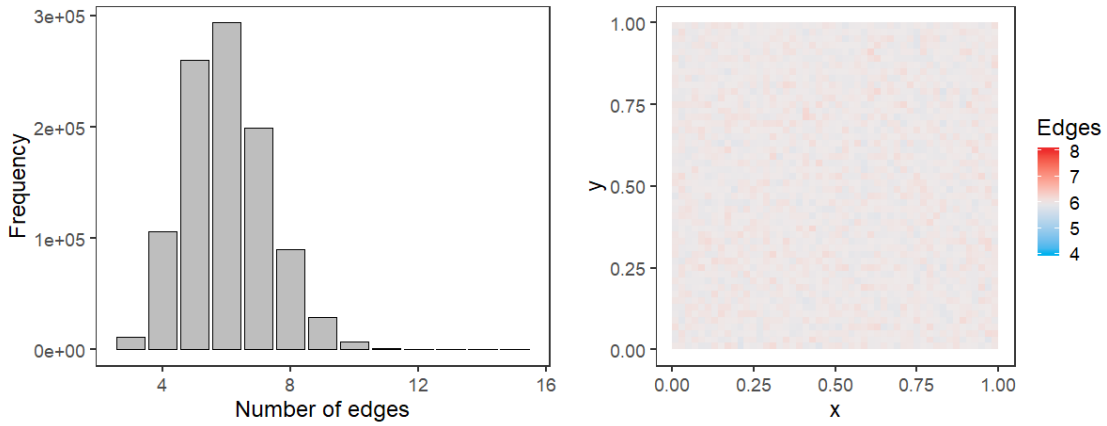


Figure 2.6: Number of cell edges in the infinite plane (left). Surface plot of number of cell edges in the infinite plane (right).

# of edges	3	4	5	6	7	8	9
Counts	11360	106358	260419	293821	199110	90317	29523
# of edges	10	11	12	13	14	15	
Counts	7312	1495	252	27	5	1	

Table 2.3: Number of cell edges in the infinite plane and the occurrences observed.

2.5.2 Voronoi tessellation using unit square boundary

In this section, we consider imposing boundaries for homogeneous Poisson points in the infinite plane, particularly, the unit square boundary. Similar to the Figure 2.4, cell areas are calculated for one million Poisson Voronoi cells bounded with the unit square and the surface (left) and line (right) plots are produced in Figure 2.8. The lines are created based on the points that are sampled from different transects of the region that are explained in Figure 2.7. This allows us to investigate the local details of the surface plots.

We select three transects and create line plots based on the cell area of the points located on these transects. In Figure 2.8 (right) the line plots are created for the transects based on the image (left). The transects are shown in Figure 2.7 using the arrows with the same colours of lines in Figure 2.8. First, we chose a *diagonal* transect from bottom-left corner to the top-right corner of the region and showed in red colour. Note that this can be achieved in four different ways. Second, another transect called *middle* that is horizontal and located at the $y = 0.5$, and shown in a green colour. The vertical middle transect can also be used which will have symmetrical properties with the horizontal transect. Lastly, an *edge* transect is selected and shown in blue. This can also be done for any four edges of the region. Using these transects, we average the data over the pixel bins in that transect to create the line plots in Figure 2.8 (right).

In the surface plot in Figure 2.8, it is seen that the Voronoi cells are likely to behave differently depending on their location. Points that are very close to any edge have smaller cell areas (region coloured in blue) than the ones close to the centre of the region which are not affected by the boundary. The red parts that are relatively close to the boundaries show that the cell area is higher than the mean cell area at these regions. This is the case when the point associated with a Voronoi cell is far from the boundary but the cell has a vertex on the boundary. Hence the size of the cell become large. Cells which are not affected by the boundary, which can be

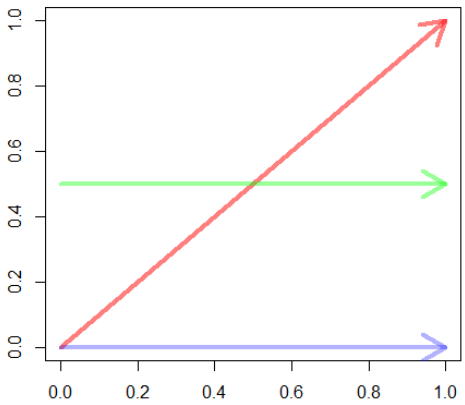


Figure 2.7: Transects used in the line plots in Figure 2.8.

thought as the ones located interior to the region shows the same characteristics as in the infinite plane case.

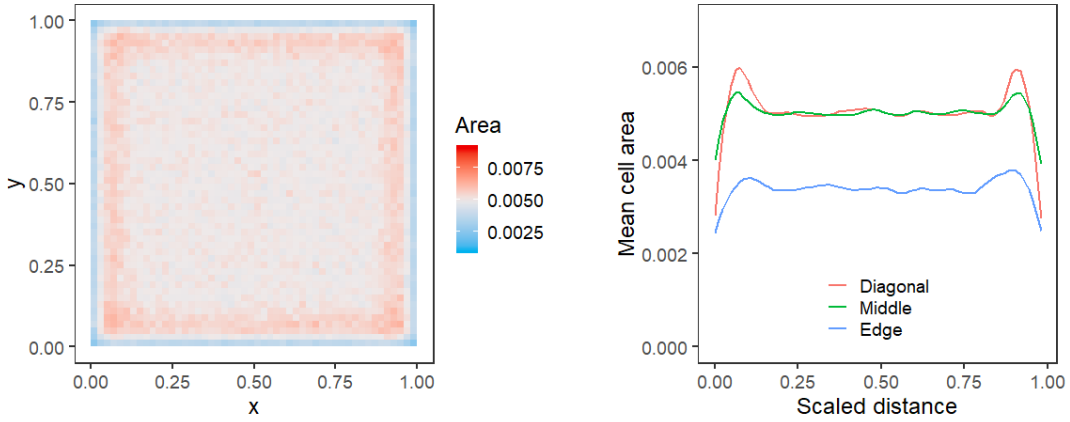


Figure 2.8: Surface plot of cell area in a unit square (left). Averaged cell area over the grids against the scaled distance of the direction being followed on the region (right). Different directions are shown in Figure 2.7.

In conclusion, restriction of the infinite plane with a regular rectangle boundary, namely the unit square, causes cells to have different sizes conditioning on the location. For instance, Voronoi cells located very close to the corner of the boundary are found to be very small in size as it deviates from location to location as specified from blue to red colour in the surface plot in Figure 2.8. Line plots also demonstrate the changes in the cell area proximity to the boundary.

Another measure, cell perimeter, is similarly visualized in Figure 2.9. Cells intersect the unit square boundary, have perimeter partly constitute a small part of the

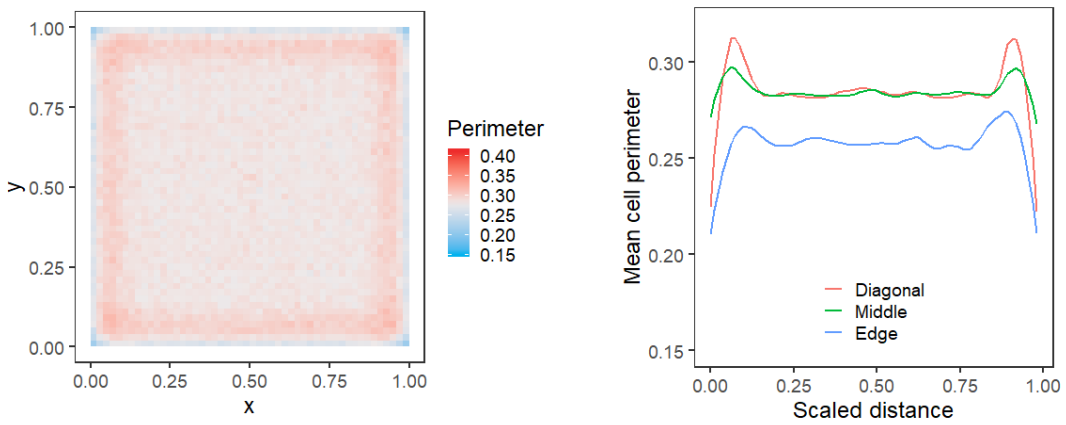


Figure 2.9: Surface plot of cell perimeter in a unit square (left). Averaged cell perimeter over the grids against the scaled distance of the direction being followed on the region (right).

boundary. We observe similar patterns in both surface and line plots as the cell area but the gap between the cell perimeter for cells laying on the boundary is not too wide than the mean cell perimeter that means the cells close to the boundary do not have very small perimeter. Only the corner cells have noticeably small perimeter.

Number of cell edges as a surface plot for unit square boundary is given in Figure 2.10. Considering the expected number of cell edges $E(N) = 6$ over the infinite plane, cells affected with a regular rectangle boundary are likely to have smaller number of edges than the mean cell edges and the number of cell edges gets smaller as it gets closer to the boundary. Diagonal and middle transects from the surface plot shows a similar pattern in terms of the mean cell edges. However, if we walk from one corner to the next corner, namely on the edges of the region, a significant reduction in the mean cell edge is observed. More importantly there is no location where the number of cell edges are observed to be greater than 6.

2.5.3 Voronoi tessellation using convex hull boundary

In this section, the properties of Voronoi cells within the convex hull of points will be investigated. In this case, homogeneous Poisson points generated in $\Omega = [0, 1]^2$ are restricted using the convex hull of the points. Hence, the convex hull stands for the boundary that restricts the Voronoi cells. Convex hull is the smallest convex polygon which contains all the generated points. For the sets of points we generated in the realizations of of the simulation, we use the convex hull as the boundary and

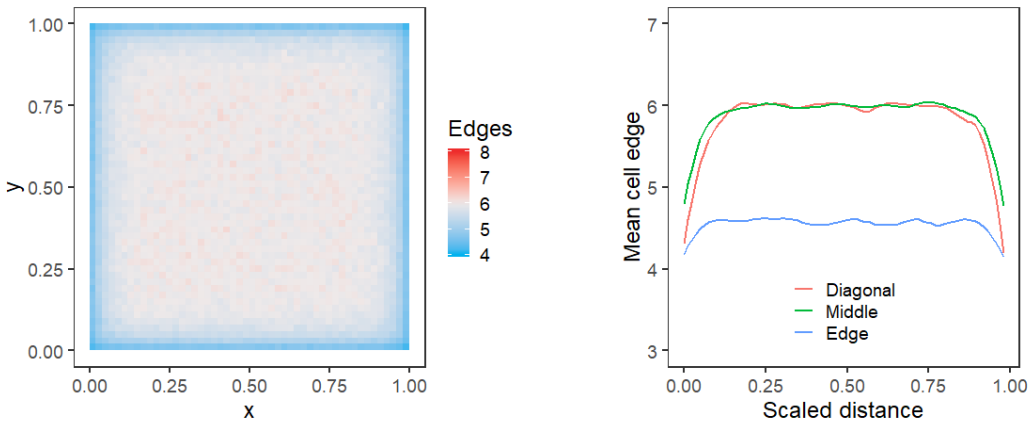


Figure 2.10: Surface plot of number of cell edges in a unit square (left). Averaged number of cell edges over the grids against the scaled distance of the direction being followed on the region (right).

calculate the cell properties for the cells that are restricted by the convex hull. The results for each property are presented and discussed respectively.

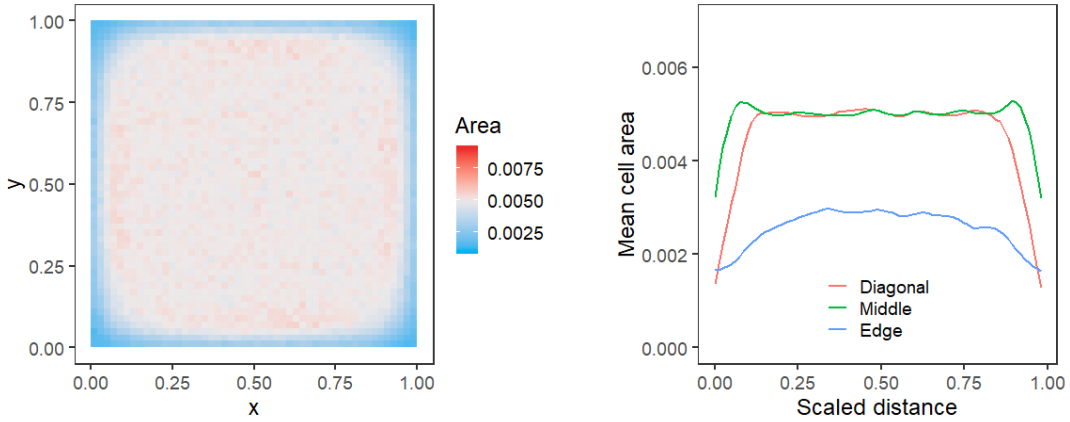


Figure 2.11: Surface plot of cell area in convex hull (left). Averaged cell area over the grids against the scaled distance of the direction being followed on the region (right).

Surface plots are given in Figure 2.11, 2.12, and 2.13 for the area, perimeter and number of cell edges respectively with line plots as described in Figure 2.7. Recall the cells in a unit square which intersect the boundary with point falls apart from the boundary were likely to be larger than the mean cell area. Area of such cells are observed in the levels of the red colour. On contrary to the unit square boundary, it is seen from Figure 2.11, 2.12, and 2.13 that the area of cells close to the convex hull are usually smaller than the expected cell area. Also, cells affected

by the convex hull are likely to have smaller area and perimeter compared to the ones affected by the unit square and such measures take the smallest values at the corners of the region. Number of cell edges will be larger in the convex hull case than the unit square for the cells close to the boundary which is possibly because of having irregular shapes in the convex hull compared to the unit square where the boundary has four straight lines which reduces the number of cell edges intersecting the boundary.

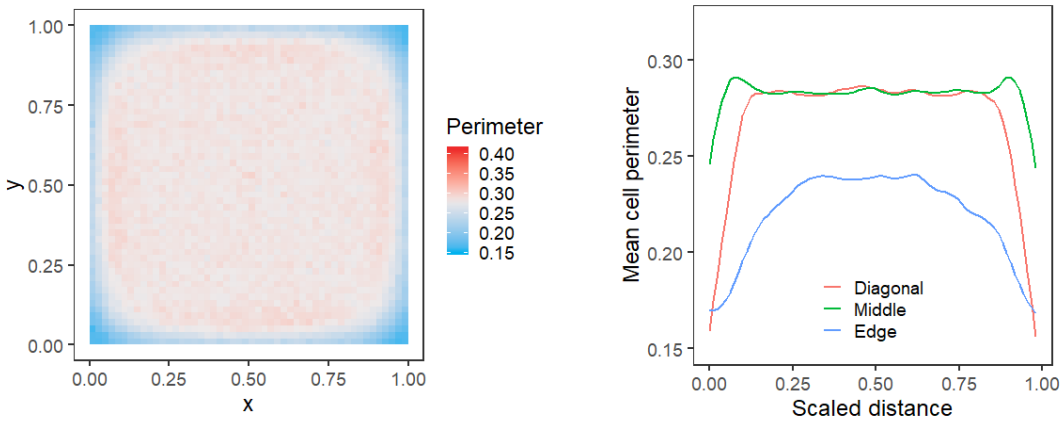


Figure 2.12: Surface plot of cell perimeter in convex hull (left). Averaged cell perimeter over the grids against the scaled distance of the direction being followed on the region (right).

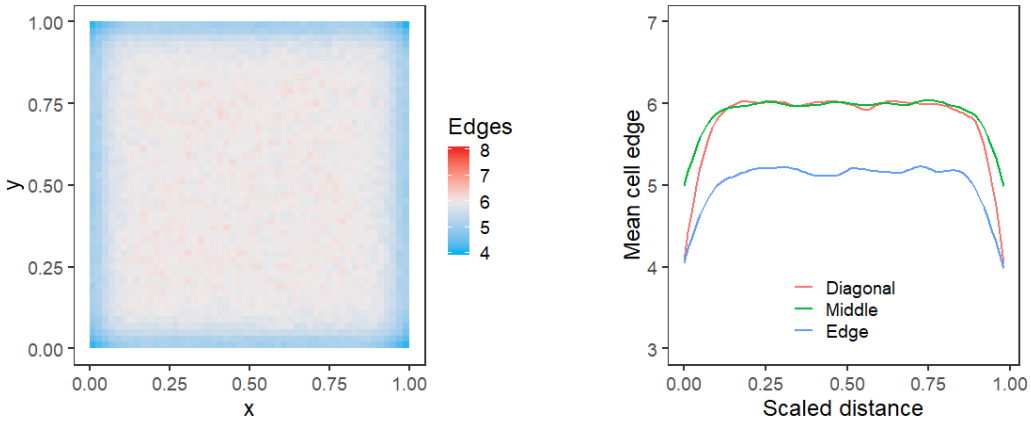


Figure 2.13: Surface plot of number of cell edges in convex hull (left). Averaged number of cell edges over the grids against the scaled distance of the direction being followed on the region (right).

2.6 Comparisons of different boundary cases and the previous work

In this section, the aim is to discuss the results from Section 2.5.1, 2.5.2, and 2.5.3 in a comparative manner. The same standardization method with the previous work is used and the results are presented for standardized area s, standardized perimeter p, and the number of cell edges N. Summary statistics are given in Table 2.4 for the measures of Voronoi cells for three different cases. Mean cell area $\bar{s}_I$ and perimeter $\bar{p}_I$ are calculated for cells in the infinite plane are found very close to the analytically calculated expectations $E(s) = E(p) = 1$, however, the mean cell area and perimeter for the unit square boundary case are calculated as $\bar{s}_U = 1.137$ and $\bar{p}_U = 1.097$. For the convex hull case, it is calculated as $\bar{s}_C = 0.886$ and $\bar{p}_C = 0.951$. Therefore, cells bounded with unit square will have larger area and perimeter than the cells in the infinite plane, and convex hull bounded cells will have the smallest area and perimeter. Here, when calculating the measures for cells in the unit square and convex hull, only the cells affected by the boundary are taken and the ones interior to the region are avoided which carries the same information from the infinite plane case. Hence, the summary statistics are calculated for the cells given they are affected by the boundary.

	Case	Mean	SD	Skewness	Kurtosis
Area	Infinite plane	1.004	0.531	1.022	1.525
	Unit square	1.137	0.654	1.164	2.247
	Convex hull	0.886	0.599	1.186	1.957
Perim.	Infinite plane	1.002	0.244	0.190	-0.025
	Unit square	1.097	0.294	0.240	0.106
	Convex hull	0.951	0.307	0.146	-0.141
Edges	Infinite plane	6	1.334	0.432	0.204
	Unit square	5.364	1.203	0.497	0.278
	Convex hull	5.432	1.185	0.509	0.321

Table 2.4: Mean, standard deviation, skewness and kurtosis of standardized area, perimeter and number of edges of Poisson Voronoi cells in infinite plane, and for unit square and convex hull boundaries.

Two and three-parameter gamma distributions are fitted for the standardized cell area, perimeter and number of edges. It is discussed in the previous work that the gamma distribution gives the best approximation for these measures. Although the number of cell edges takes integer values in the rage of $[3, 15]$ and hence has

a discrete distribution, the gamma distribution is still used and the parameters of the gamma distribution are estimated only using the observed integer values as suggested in Hinde & Miles (1980). We used the gamma distribution for the number of cell edges to have a comparison of the estimated parameters with the previous work Hinde & Miles (1980) and Tanemura (2003). An appropriate alternative could be the Poisson distribution.

Figure 2.14 shows the mid points of the histogram bins for the observed measures with solid points (•) and fitted two and three-parameter gamma densities (·····) and (——) with estimated parameters from Table 2.5 and 2.6 respectively. Plots in the first column are the results for the infinite plane, second row for the unit square, and the bottom row are for the convex hull cases. Three-parameter gamma distribution with blue lines shows a great performance to fit the measures, however, even though two-parameter gamma performs well in many cases, it cannot fit the cell perimeter as good as the three-parameter. In addition to the gamma distribution, several others, Weibull and log-normal distributions are checked but their performances were not satisfactory as can be seen in Figure A.1 in Appendix A for the standardized areas in the infinite plane.

Disparities on the statistical properties are discovered through the surface plots, and summary statistics for three cases of Voronoi cells are verified by the estimated parameters of two and three-parameter gamma distributions in Table 2.5 and 2.6 respectively. We can conclude that the measures of Voronoi cells can be estimated via three-parameter gamma distribution with appropriately chosen parameters. Parameters estimated for the cells in the infinite plane shows a great agreement with (Hinde & Miles, 1980) and (Tanemura, 2003) for three-parameter gamma distribution, and similar parameters are estimated for two-parameter gamma case.

In addition to the parameter estimates in Table 2.5 and 2.6, we also calculated the 95% confidence intervals for the parameter estimates. The parameters are estimated by maximizing the log-likelihood function in (2.4). The second-order partial derivative of the log-likelihood function evaluated at the maximum creates the Hessian matrix

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 l}{\partial a^2} & \frac{\partial^2 l}{\partial a \partial b} & \frac{\partial^2 l}{\partial a \partial c} \\ \frac{\partial^2 l}{\partial b \partial a} & \frac{\partial^2 l}{\partial b^2} & \frac{\partial^2 l}{\partial b \partial c} \\ \frac{\partial^2 l}{\partial c \partial a} & \frac{\partial^2 l}{\partial c \partial b} & \frac{\partial^2 l}{\partial c^2} \end{bmatrix}. \tag{2.5}$$

The diagonal elements of the inverse of negative Hessian matrix $(-\mathbf{H}^{-1})$ are the estimated variances $(\hat{\sigma}_a^2 \;\; \hat{\sigma}_b^2 \;\; \hat{\sigma}_c^2)$ for the parameters. Then a confidence interval for

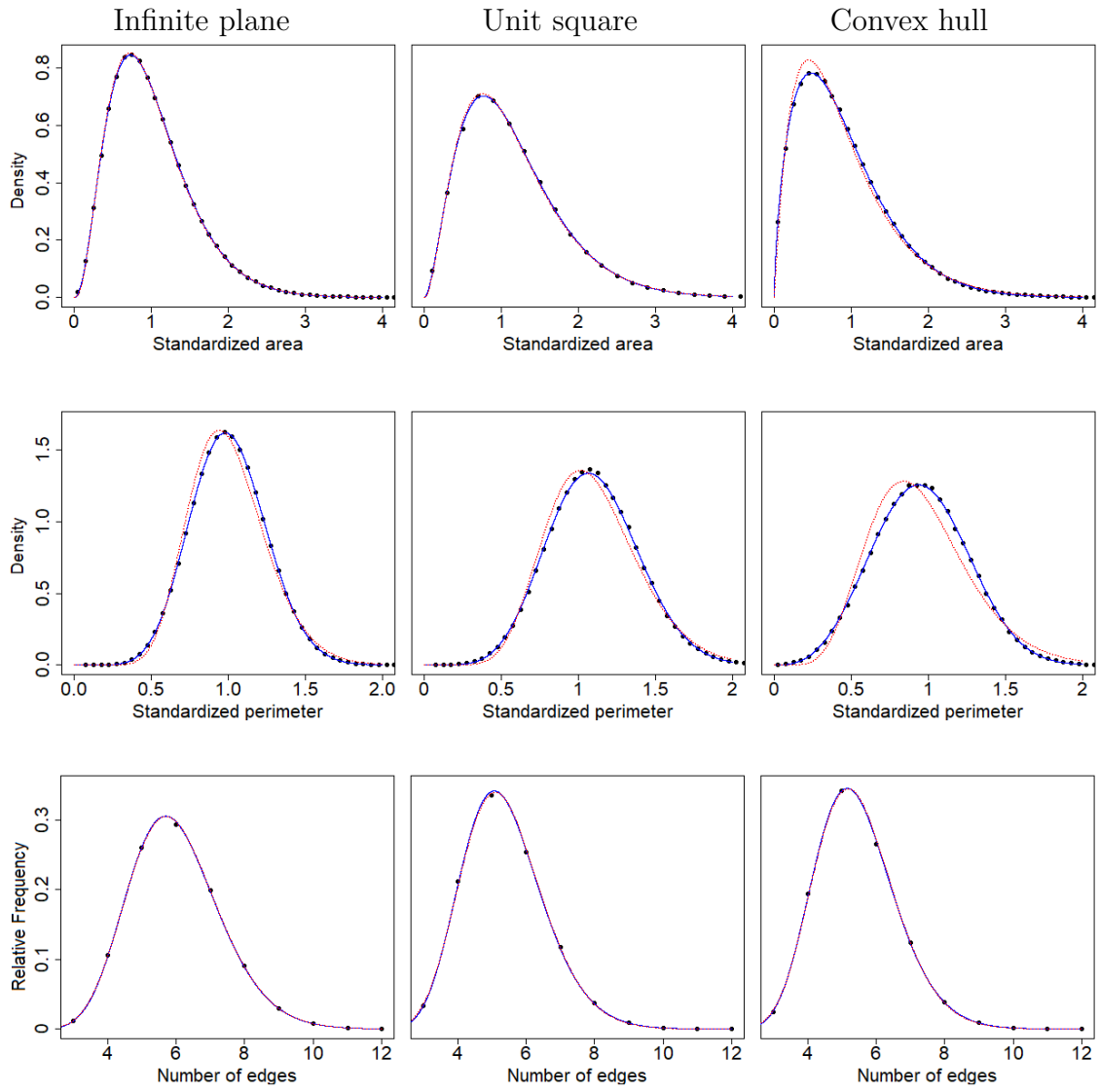


Figure 2.14: Density lines of two and three-parameter gamma distributions following the estimated parameters given in Table 2.5 and 2.6. Mid points of the histogram bins of observed values for cell measures are shown in (•). Two and three-parameter gamma fit are shown in (·····) and (——) respectively. First column of plots are for the cells in the infinite plane, second column for the unit square and the third column is for the convex hull case. Each row of plots are for the measures; standardized area, perimeter, and number of edges respectively.

an estimated parameter $\hat{a}$ is calculated as $[L, U]_{\hat{a}} = \hat{a} \pm 1.96 \times \hat{\sigma}_a$ where L and U are the lower and upper bounds of the interval respectively.

Lastly, fitted density lines for cell area and perimeter based on the number of cell edges are overlaid on lattice plots in Figure 2.15 and 2.16 for all boundary cases. Since the fitted gamma distribution does not differ much for cells having more

	Case	$\hat{b}$	$\hat{c}$
area	Infinite plane	3.510 (3.500-3.520)	3.526 (3.516-3.535)
	Unit square	2.626 (2.612-2.640)	2.986 (2.971-3.000)
	Convex hull	2.271 (2.259-2.284)	2.012 (2.003-2.022)
	Kiang (1966)	4.0	4.0
	Weaire *et al.* (1986)	3.63	3.63
	Kumar & Kurtz (1993)	3.7176	3.7174
	Koufos & Dettmann (2019)	3.5691	3.5691
perim	Infinite plane	16.038 (15.993-16.082)	16.074 (16.030-16.118)
	Unit square	11.903 (11.840-11.965)	13.062 (12.995-13.129)
	Convex hull	8.892 (8.846-8.938)	8.464 (8.422-8.507)
edges	Infinite plane	3.370 (3.361-3.380)	20.220 (20.164-21.276)
	Unit square	3.740 (3.721-3.760)	20.064 (19.960-20.167)
	Convex hull	3.911 (3.891-3.931)	21.245 (21.137-21.353)

Table 2.5: Parameter estimations and confidence intervals in the parentheses of the two-parameter gamma distribution fitted to the area, perimeter and number of edges of Poisson Voronoi cells in the infinite plane, and with unit square and convex hull boundaries.

than 7 edges, observations for $N = 8, ..., 15$ are aggregated and denoted as $N > 7$. Infinite plane, unit square, and convex hull cases are shown in (——), (·····), and (- - - -) respectively. It is obvious that different number of cell edges leads to different estimates for parameters of three-parameter gamma distribution. Also, fitted density lines never coincide as the number of cell edges change. Consequently, the mean of the fitted gamma distributions are different for different values of N. The mean is smallest when $N = 3$ and it increases for larger N. We note that the cell perimeter distributions are more similar for the two bounded cases than for the infinite plane case, while the cell area distributions do not show this pattern.

The changes in the cell propeties is also considered. We denote the change in the area of a cell as

$$\text{area reduction} = (\text{area with no boundary - area with boundary}).$$

To visualize the area reduction when the boundaries are imposed, we presented the histogram and surface plot concerning this reduction in Figure 2.17 for unit square (first row) and convex hull (second row) boundaries. Surface plot in the first row of Figure 2.17 indicates that cells very close to the unit square boundary had shrinkage in their sizes where the ones close to the corner shrank the most. Also, cells whose

	Case	$\hat{a}$	$\hat{b}$	$\hat{c}$
Area	Infinite plane	1.080 (1.073-1.106)	3.015 (2.961-3.069)	3.311 (3.285-3.336)
	Unit square	1.089 (0.969-0.987)	2.195 (2.123-2.267)	2.792 (2.755-2.828)
	Convex hull	1.336 (1.320-1.352)	1.336 (1.304-1.367)	1.667 (1.652-1.681)
	Tanemura (2003)	1.0795	3.0322	3.3112
	Hinde & Miles (1980)	1.0787	3.0328	3.3095
Perim.	Infinite plane	2.334 (2.313-2.356)	2.963 (2.906-3.019)	7.593 (7.533-7.654)
	Unit square	2.278 (2.242-2.314)	2.062 (1.987-2.137)	6.412 (6.326-6.498)
	Convex hull	2.917 (2.882-2.953)	1.156 (1.126-1.186)	3.779 (3.746-3.812)
	Tanemura (2003)	2.33609	2.97006	7.58060
	Hinde & Miles (1980)	2.3389	2.9563	7.5579
Edges	Infinite plane	0.931 (0.920-0.941)	4.406(4.222-4.590)	21.706(21.469-21.943)
	Unit square	0.775 (0.768-0.783)	9.088 (8.807-9.368)	25.804 (25.543-26.064)
	Convex hull	0.813 (0.805-0.820)	8.089 (7.843-8.335)	25.928 (25.675-26.181)
	Tanemura (2003)	0.96853	3.80078	20.86016
	Hinde & Miles (1980)	1.0186	3.130	19.784

Table 2.6: Parameter estimations and confidence intervals in the parentheses of the three-parameter gamma distribution fitted to the area, perimeter and number of edges of Poisson Voronoi cells in the infinite plane, and with unit square and convex hull boundaries.

points are far from the boundary likely to expand if they intersect the boundary. Cells located at the white area has no change in their area.

On the other hand, convex hull bounded cells in the second row are more likely to shrunk when the boundary is imposed. There are some cells located far from the boundary but affected boundary showed a slight expansion that is visible at the parts with the light-blue colour. Histograms of area reduction in both cases show a high peak around zero where the change is very small. The skew on the histogram of area reduction for convex hull bounded cells indicates that having convex hull boundary causes higher reduction on their sizes.

Perimeter reduction for the imposed boundaries are given in Figure 2.18 which shows similarities with cell area results. The perimeter reduction is formulated as

$$\text{perimeter reduction} = (\text{perimeter with no boundary - perimeter with boundary}).$$

Shrinkage on the perimeter in the unit square case is not very dramatic and similar patterns are observed in both boundary cases. Figure 2.19 shows the reduction in the number of cell edges. Imposing both boundary cases causes the number of cell edges be less than the infinite plane case. Especially, cells very close to the corner of the boundary are likely to have smallest number of edges. Reduction in the edge

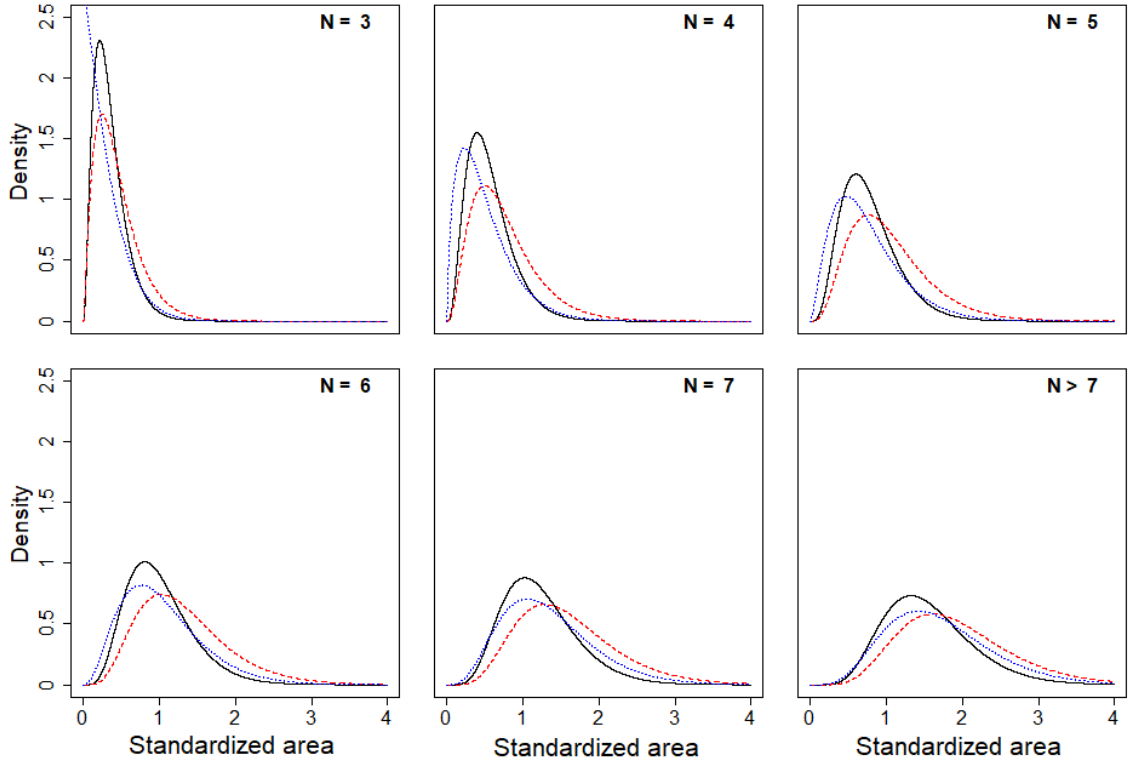


Figure 2.15: Density lines of three-parameter gamma distribution for the estimated parameters of standardized area for infinite plane (——), unit square (·····), and convex hull (- - - -) cases with respect to number of cell edges $N = 3, 4, ..., 7$ and $N > 7$.

is also formulated as

edge reduction = (# of edges with no boundary - # of edges with boundary).

Ratio of the area, perimeter and and number of cell edges are shown in Figure 2.20, 2.21 and 2.22. Ratios are calculated replacing the formulas above by the proportion of the cell properties rather than the differences. Gray area corresponds to ratio of 1. In the unit square boundary case, similar patterns for the ratio of all measures are observed with the reduction. However, very extreme values are detected for the convex hull boundary. These extreme values of area and perimeter ratios are generally located very close to the corners of the unit square boundary.

Reduction and ratio of cell properties have histograms with a high peak is attempted to be estimated through possible distributions. Two candidate distributions are considered and fitted to the histograms of reductions and ratios of the measures.

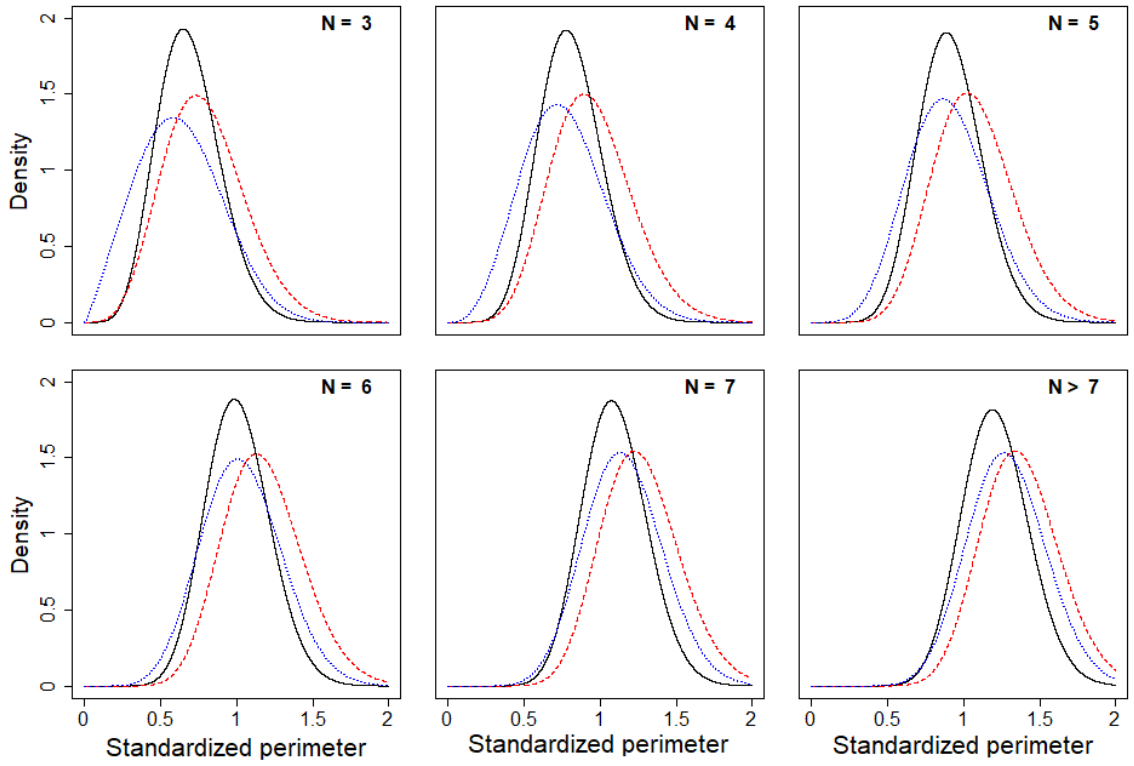


Figure 2.16: Density lines of three-parameter gamma distribution for the estimated parameters of standardized perimeter for infinite plane (——), unit square (·····), and convex hull (- - - -) cases with respect to number of cell edges $N = 3, 4, ..., 7$ and $N > 7$.

First, asymmetric Laplace distribution with density function

$$f(y|\mu, \sigma, p) = \frac{p(1-p)}{\sigma} \exp\left\{-\rho_p(\frac{y-\mu}{\sigma})\right\} \tag{2.6}$$

with location parameter $-\infty < \mu < \infty$, scale parameter $\sigma > 0$ and skewness parameter $0 < p < 1$ is used (Koenker & Machado, 1999; Yu & Moyeed, 2001). Here, $\rho_p(.)$ is the loss function defined as $\rho_p(u) = u(p - I_{u<0})$. Second, a weighted double exponential distribution with a variant mode which has a density function

$$f(x; \omega, a_1, a_2, \mu) = \begin{cases} \omega a_1 \exp\left(-a_1(\mu - z)\right) & \text{for} \quad z < \mu \\ (1-\omega) a_2 \exp\left(-a_2(z - \mu)\right) & \text{for} \quad \mu \geq z \end{cases} \tag{2.7}$$

is defined. Here, $0 < \omega < 1$ is the weight parameter corresponding to some mean parameter μ which is taken as fixed but can take values in $-\infty < \mu < \infty$, and $a_1, a_2 > 0$ are the rate parameters.

Parameters of both distributions are estimated using maximum likelihood method,

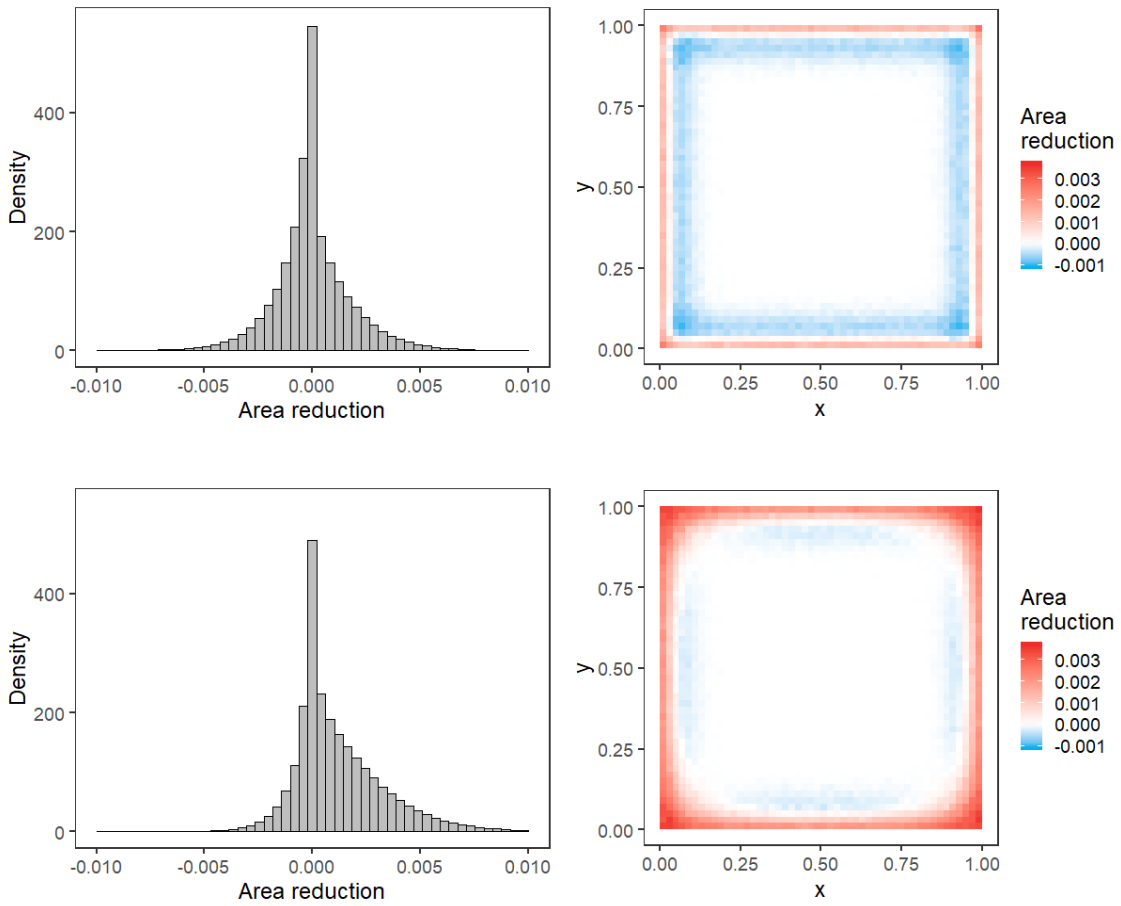


Figure 2.17: Area reduction of cells when the unit square (top) and convex hull (bottom) are imposed as a boundary. Histogram of density (left) and surface plot of area reduction (right).

and fitted density lines are given for the standardized area reduction for the unit square boundary case in Figure 2.23. Only one example for the performance of asymmetric Laplace and adjusted density is shown. Estimated parameters for the asymmetric Laplace are $\mu = -0.021$, $\sigma = 0.122$ and $p = 0.480$ and for the adjusted density, $\omega = 0.601$, $a_1 = 4.810$, $a_2 = 3.304$ and $\mu = 0.004$. Although the best approximations are observed using these densities, their performance is not satisfactory. Therefore, the rest of the properties are not considered.

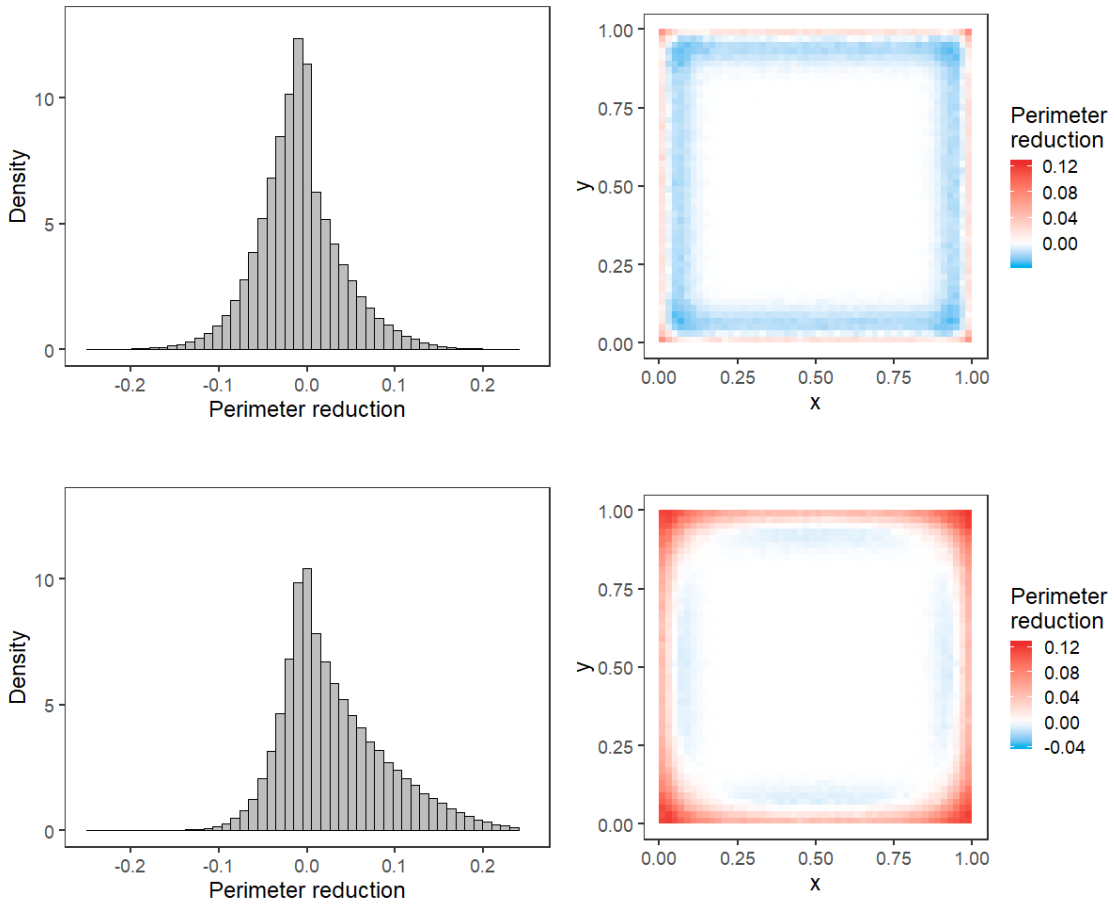


Figure 2.18: Perimeter reduction of cells when the unit square (top) and convex hull (bottom) are imposed as a boundary. Histogram of density (left) and surface plot of perimeter reduction (right).

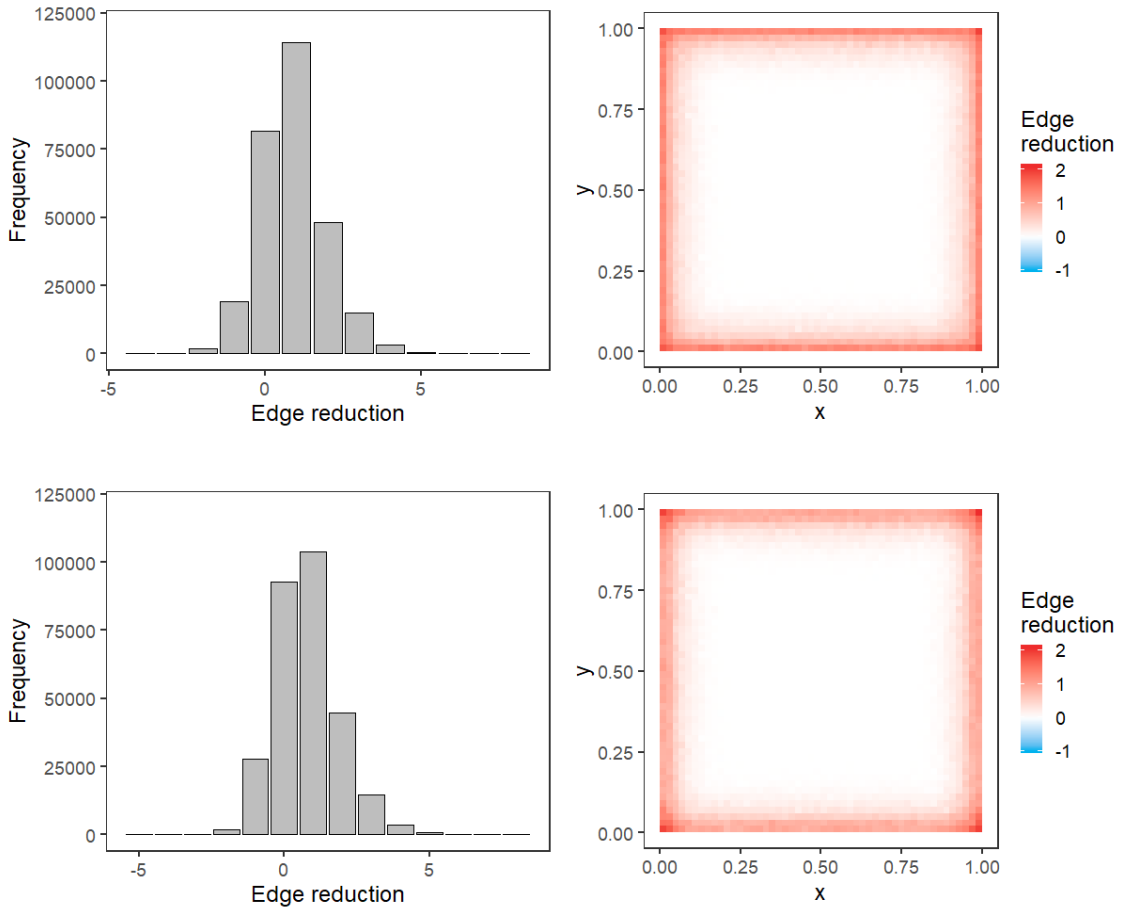


Figure 2.19: Reduction on the number of cell edge when the unit square (top) and convex hull (bottom) are imposed as a boundary. Histogram (left) and surface plot of reduction on the number of cell edge (right).

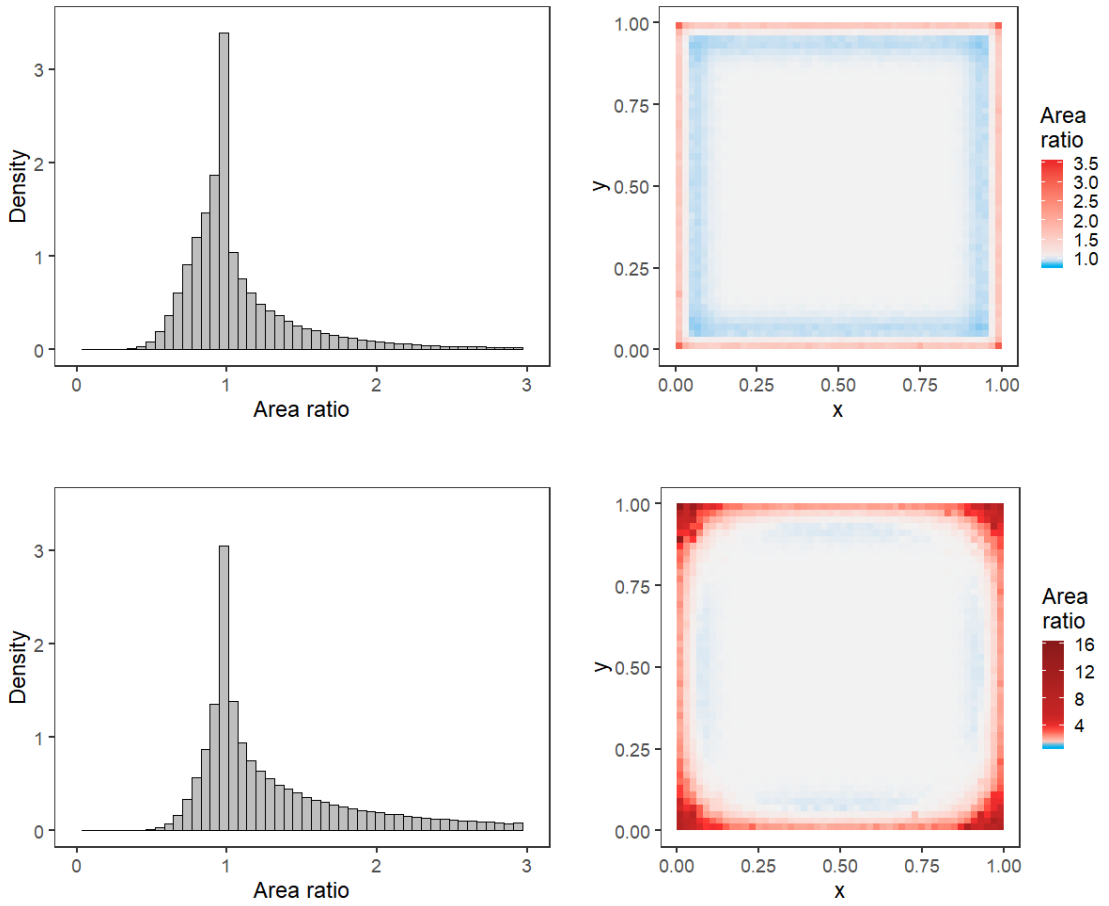


Figure 2.20: Area ratio of cells when the unit square (top) and convex hull (bottom) are imposed as a boundary. Histogram of density (left) and surface plot of area ratio (right).

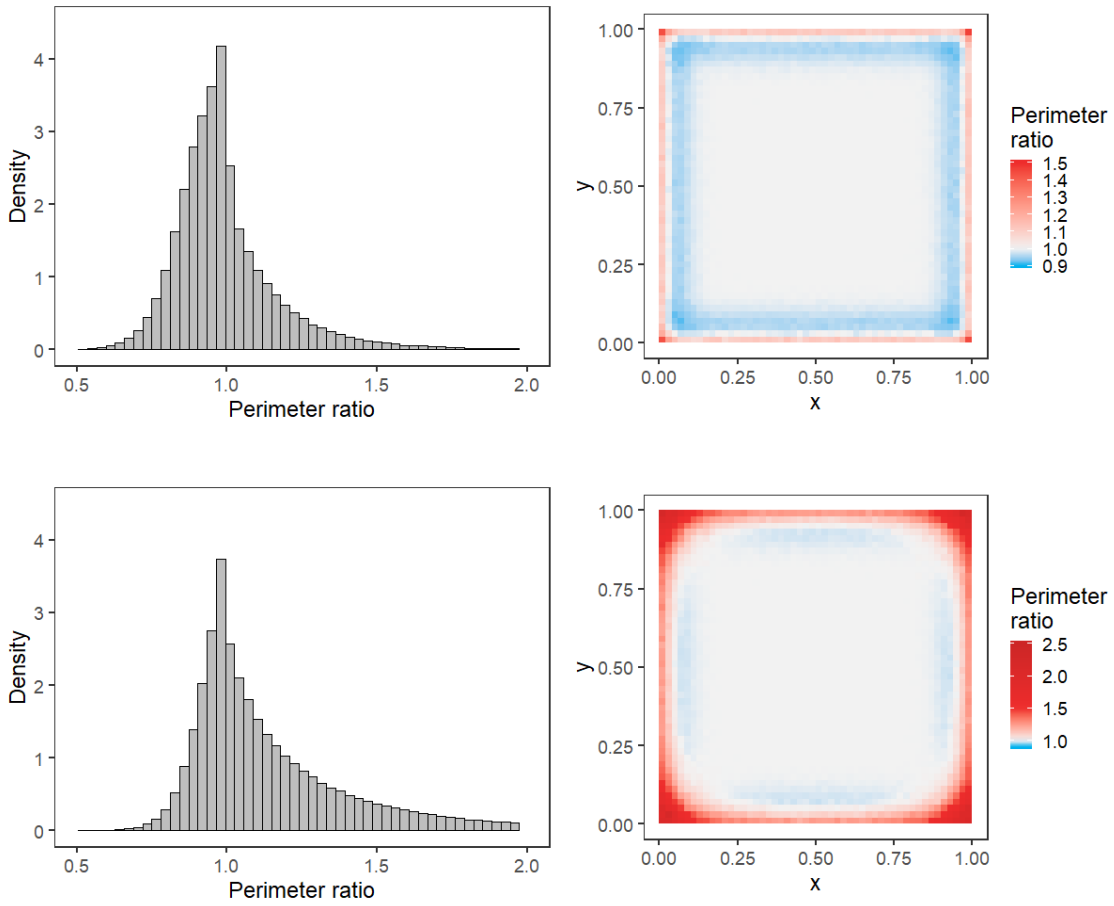


Figure 2.21: Perimeter ratio of cells when the unit square (top) and convex hull (bottom) are imposed as a boundary. Histogram of density (left) and surface plot of perimeter ratio (right).

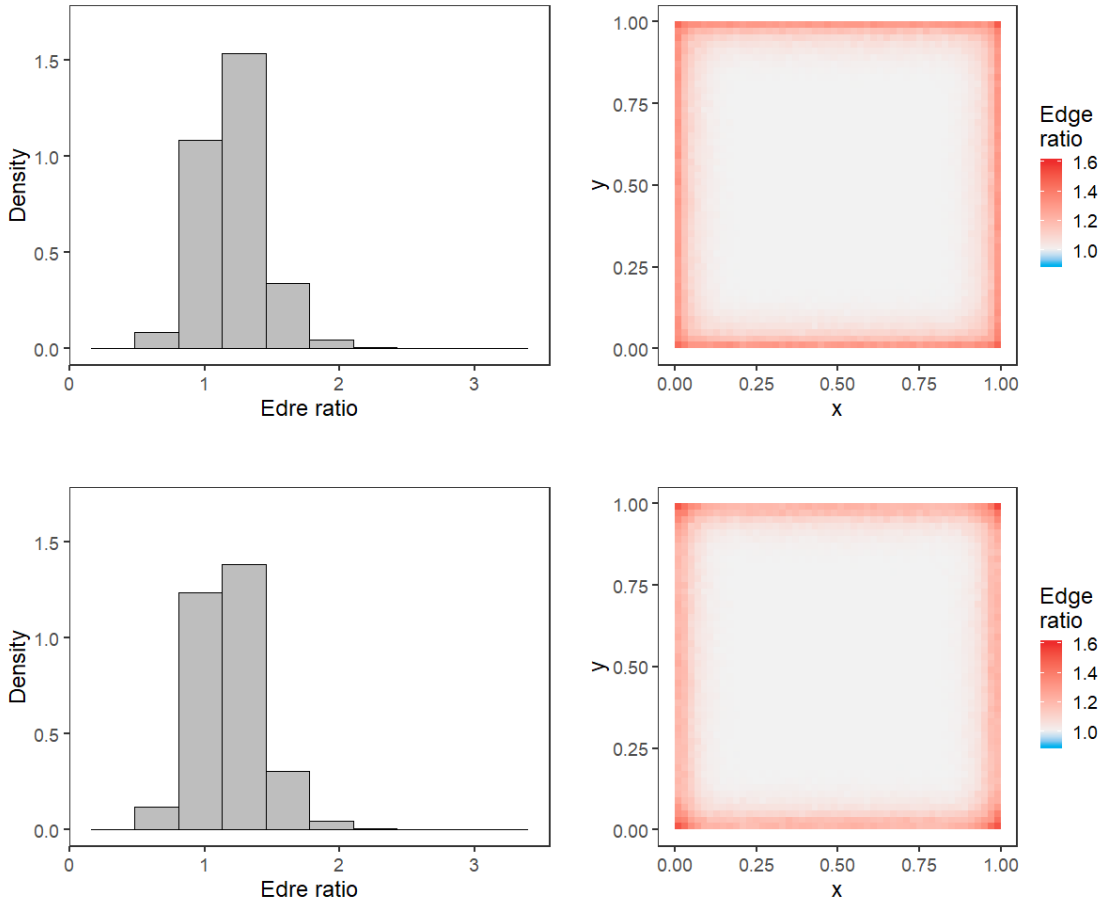


Figure 2.22: Ratio of number of cell edges when the unit square (top) and convex hull (bottom) are imposed as a boundary. Histogram of density (left) and surface plot of ratio of number of cell edges (right).

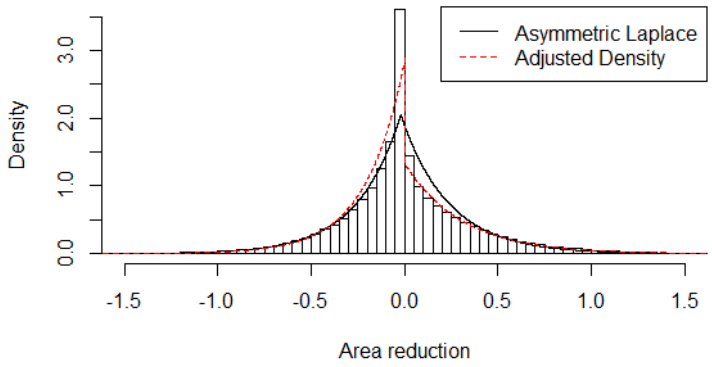


Figure 2.23: Histogram of area reduction for the unit square bounded cells and density lines of asymmetric Laplace and the adjusted distributions with the estimated parameters.

2.7 PVT for different intensities

In this section, Poisson Voronoi tessellation for different intensities is explored. The entire experiment discussed in Section 2.5 was based on PVT with intensity $\rho = 200$, however, it is important to discover whether the behaviour of cells vary as the ρ changes. Hence, another experiment is performed considering $\rho = \{30, 50, 100, 300\}$ for $r = 10^6$ realizations and results are compared with the ones obtained before.

Again, surface plots are aimed to be created to see the patterns of cell measures over the surface as in Section 2.5.2 and 2.5.3. Since the mean cell area depends on the intensity of points as $E(s) = 1/\rho$, the same standardization method is applied for all experiments based on different ρ values to make results comparable. Line plots are produced in Figure 2.24 from the surface plots following a horizontal direction at the middle of the region as described in Figure 2.7. Each ρ value is assigned to a colour and put together for i.e standardized area for unit square bounded region (top left), number of edges for convex hull bounded region (bottom right) and so on.

Standardization of measures is useful for the comparison of different cases. A general comment from the plots is the occurrence of different patterns over a surface depending on the location. However, the variability is mostly observed on the cells closer to the boundary of the region. There is not a big difference on the cell measures when moved closer to the centre of the region which are possibly the cells that are not affected by the boundary for all cases except when $\rho = 30$. The initial intuition for $\rho = 30$ is that the number of the points generated for such intensity could sometimes be very smaller than 30. Hence, a randomly sampled cell could still be affected by the boundary even though it is located very far from the boundary.

Probability of a cell being affected by the boundary for lower intensities lead us to understand the circumstances of having smaller number of points and the pattern that they generate on the surface. Particularly, unit square or convex hull boundary may still affect the characteristics of a Voronoi cell whose associated point is very far from the boundary. Another simulation is performed to find the probability of a randomly sampled cell being affected by the boundary when the number of points are $m = \{10, 15, 20, 30, 50\}$. Number of points are fixed in each simulation instead of fixing the intensity to avoid the variability on the number of points generated for each specified intensity especially when the ρ is very small, and the simulation is run for $r = 10^5$ realizations.

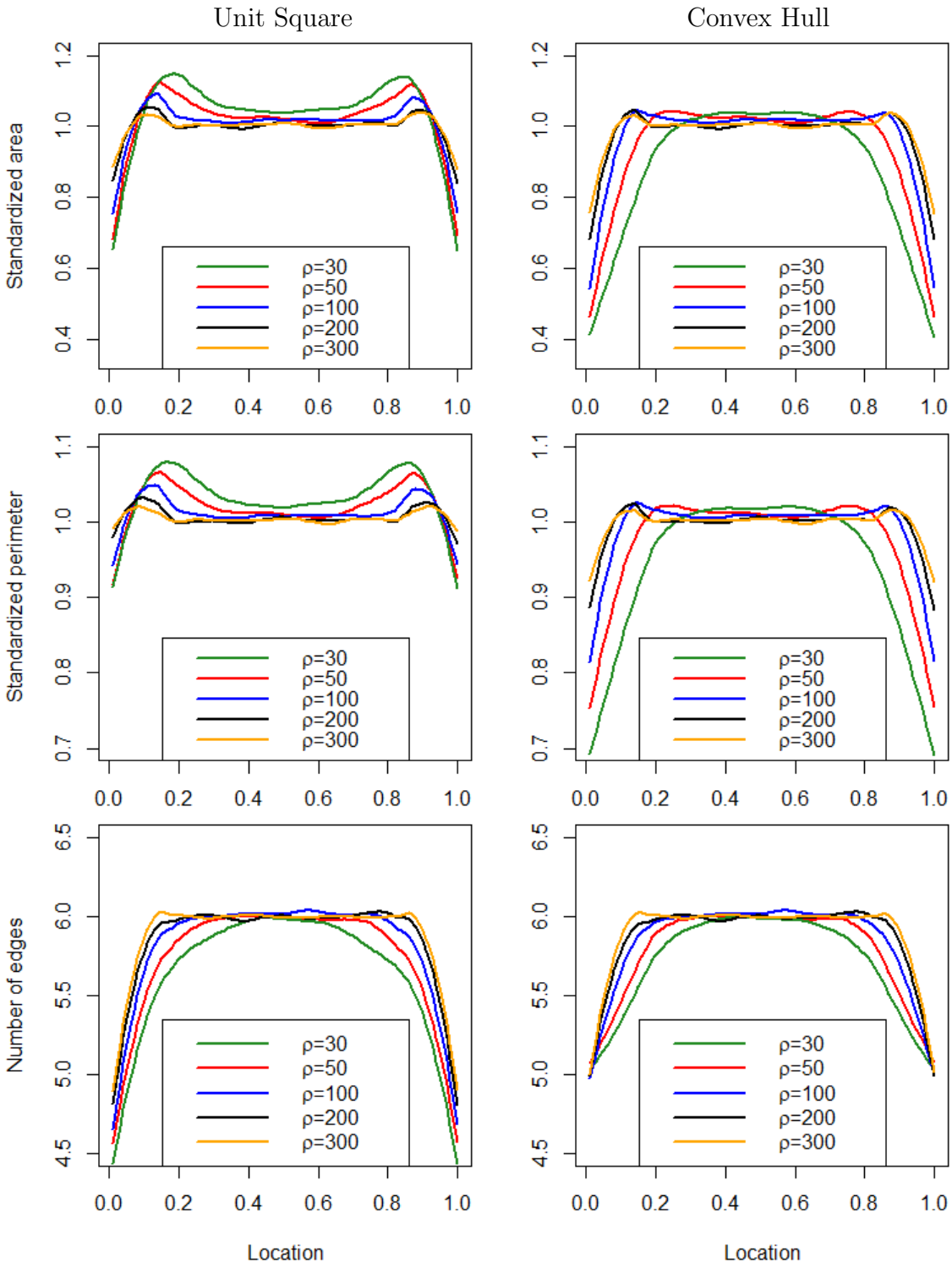


Figure 2.24: Standardized PVT properties for intensities $\rho = 30, 50, 100, 200, 300$ for points across the centre of the region. Each ρ is assigned to a colour that show the pattern standardized cell properties. First and second column of the plots show the results for unit square and convex hull bounded cells, and rows panels are for cell area, perimeter and number of edges respectively.

Proportions of number of cells affected by the unit square and convex hull boundaries based on the number of points is presented in Figure 2.25. Almost every randomly sampled cell among 10 points is affected by the boundary and the proportion decreases as the number of points increases. The proportion of cells affected by the convex hull is slightly higher than the unit square case but the proportions converge at higher number of points. The different pattern observed in Figure 2.24 for $\rho = 30$ is also highlighted here for lower intensities in general.

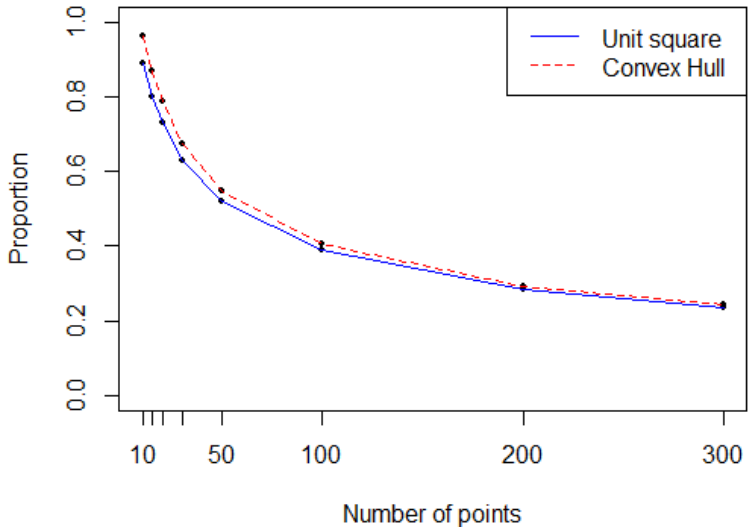


Figure 2.25: Proportion of boundary-affected cells in 10^6 realizations for varying numbers of points $n \in \{10, 15, 20, 30, 50, 100, 200, 300\}$

2.8 Conclusion

To sum up, this chapter investigated Poisson Voronoi cells in two-dimensional space and their statistical properties extensively. Simulations are performed for intensity $\rho = 200$ for $r = 10^6$ realizations. Poisson Voronoi cells are initially considered in the infinite plane where the results related to distribution fitting showed a great association with the relevant literature. More importantly, boundaries are imposed to the homogeneous Poisson points. Using the unit square and the convex hull of points as boundaries, cell area, perimeter, and the number of cell edges can be estimated using the three-parameter gamma distribution. The difference in the fitted distributions raises the importance of taking into account these boundary effects in the analysis of spatial data which usually comes with its own boundary case.

Although the cell properties do not change over the infinite plane, it is not the case when the boundaries are imposed. The distance and location of the points from

from the boundary play a key role on the statistical properties of the cells. Also, we observe similar properties of cells at the edges and the corners. For instance the cell properties show symmetric properties when the unit square is folded with respect to the axes $x = 0.5$ or $y = 0.5$. In Gezer *et al.* (2021), the image plots are constructed using the folded regions to increase the sample sizes in the pixel bins. The cells affected by both boundaries do not have identical properties. Hence, in practice, further study considering other specifically determined boundaries or real boundaries such as state borders or coastlines may need to be performed for any given data set.

We have only considered the homogeneous Poisson points in a unit square region, hence further work may investigate how he results change for other point patterns and boundary types. In the previous work, Schoenberg *et al.* (2009) studied the distributions of the Voronoi tessellation cell area and perimeter of the locations of earthquakes in Southern California. The data is based on the epicentres of 7567 earthquakes that had magnitude over 3.0 between 1984 and 2007 in Southern California. The cells intersecting the various boundary options are excluded from the study due to their biased values. The study found that the tapered Pareto distribution is a suitable distribution to model the area and perimeters of the Voronoi cells. The same distribution is also used to approximate the seismic moments. This study is an example of the application of Voronoi tessellation to a real data set which has an irregular boundary type and clustered earthquake epicentres. We learn from this study that the Voronoi cells that are obtained from data types with different spatial characteristics can be modeled through different distributions than the gamma that is mostly suitable for data locations with spatial randomness.

Chapter 3

Prediction of Voronoi tessellation cell area

3.1 Overview

The statistical properties of Voronoi tessellation cells of homogeneous Poisson points in two dimensions is studied in Chapter 2. Careful attention is necessary when boundaries are imposed since they change the geometric structure of the Voronoi cells and hence the statistical properties, such as the cell area, perimeter, number of cell edges. Figure 1.1 was an excellent illustration of the changes on the Voronoi cells when a boundary line is drawn.

Previous chapters clarified that Voronoi tessellations subject to boundaries may not reflect the true cell properties since the boundary has a constraining effect. In this chapter, we aim to propose and develop ways that treats the spatial data available in a finite bounded region as if there is a larger region or an infinite plane which the data in the finite region is a subset of. Hence, the boundary effects in the data are aimed to be reduced.

One approach to accomplish this would be to predict the true cell area using regression-based models. We will give a detailed explanation and examples of conducting this approach throughout this chapter. This process can be thought as creating models that adjusts the cell area especially for the cells near the boundary. The predictors of the model is the observed cell properties within a boundary and the response is the true cell area (as if there is no boundary). To fit the regression models, we need a data set. The simulation study in Chapter 2 generated large data set of many cell properties. Splitting this data into training and validation sets, we

aim to fit regression models using the training set and evaluate and improve the performances of the fitted models in the validation sets.

The issues that the boundaries cause in the analysis of spatial data is briefly discussed in Chapter 3.2. Then, we described the data set and variables that we are going to use in the modeling in Section 3.3. The methodology we use, the steps of the model fitting process such as the division of the data into the training and validation sets, and model selection criterion etc. are explained in Section 3.4. The results of model fitting are presented in Section 3.5 and discussed extensively. Section 3.6 mentions an approach to classify the boundary-effected cells respectively. Finally, the suggestions for alternative scenarios are given in Section 3.7.

3.2 Boundary issues

Spatial data usually come with its own boundary structure. This can be a regular or irregular boundary of any sort. In some situations, a suitable rectangle can be defined as the boundary for a set of data points in two-dimensional space. This is suitable if the points are generated within the constraint of a rectangular domain. However, it is always possible to take the convex hull of points whether or not a rectangular boundary is useful. Consider a data set that contain data locations in a two-dimensional space but we do not know the exact boundary of the data. In such cases, we use the sampling region, or we can always draw the convex hull of data points and consider the convex hull as the boundary.

In the simulation study in Chapter 2, the sampling region was defined as the unit square hence we used both the unit square and convex hull as two boundary types and investigated the cell properties based on them. We may generalize the definition as *'known'* boundary and *'unknown'* boundary for these two cases respectively. Known boundary stands for the case where we are given the boundary information such as the sampling region. Otherwise, we can use the convex hull in the unknown boundary case.

There are properties that can be calculated for Voronoi cells such as the cell area, perimeter, number of cell edges, cell type, or the shortest distance from a boundary. The relationship between the cell properties may help us to understand the changes in the cell area which we are interested in. For each boundary type, unit square *known* and convex hull *unknown*, these properties are calculated differently. For instance, the closest distance from a data point to the unit square, and to the convex

hull boundary is likely to be different. Hence, we define these two distances as separate variables. In Section 3.3, we give a detailed explanation of these variables.

3.3 Description of variables

The variables are the properties of the Voronoi cells which we are able to measure. In Chapter 2 we presented how the distributions of cell area, cell perimeter and the number of cell edges change in the presence and absence of the boundaries and only focused on these three cell properties. However, in this chapter, we are interested in the prediction of true cell area (in the absence of the boundary) using the information obtained from the cells in the presence of a boundary. This approach has an implicit act such that the Voronoi cells are treated in a continuum rather than a restricted region.

To create models that predicts the true cell area, that is the outcome, we rely on regression methods that require independent variables which are the other measured properties of the Voronoi cells. We consider measuring all possible meaningful cell properties that is likely to have an effect on the cell area and use them as predictor variables. Hence the collection of variables are the ones which are likely to have a casual relationship with the outcome. Since a variable selection procedure for the choice of best model is also taken into account when fitting the models, the number of variables we initially consider is not restricted and we included as many variables as possible.

The list of candidate variables are presented in Table 3.1. These variables are obtained from the simulation data in Chapter 2 and the code to perform a single realization of the simulation is given in Appendix B. We repeatedly perform the simulation to obtain independent realizations of data sets. First, a set of homogeneous Poisson points with a specified intensity within a unit square region is generated and a point is selected at random. Voronoi tessellation of the points is done when the boundary is the original unit square boundary and the convex hull of the generated points. Then, the cell properties listed in Table 3.1 are recorded for the sampled cell. Some cell properties are calculated differently based on the boundary type that is why the first and the last columns are created separately. Finally, the sampled point is moved to the centre of the unit square by translating the relative positions of the other points and the cell area is recorded which indicates the outcome variable, true cell area.

The separate sets of variables are considered for the unit square and convex hull boundary cases. There are *(i)* variables in common, such as the point coordinates x_1, x_2, *(ii)* similar variables such as the cell area that is calculated differently based on the boundary type such as the unit square area x_3, and the convex hull area z_3 respectively, and *(iii)* some variables which are only possible to calculate for one boundary type such as the cell type x_{11}. As it is not obvious how to define the corner cells for irregular boundaries.

Unit Square Boundary	Type	Definition	Unknown Boundary
y: inf.area	CN	Cell area without boundary.	y: inf.area
x_1, x_2: x.coord & y.coord	CN	Coordinates of data points.	-
x_3, x_4, x_5: unit.area unit.per, unit.edge	CN, CN, DN	Area, perimeter and number of edges.	z_3, z_4, z_5: chull.area chull.per, chull.edge
x_6: on.chull	C	Whether a point is located on the convex hull.	z_6: on.chull
x_7: m	DN	Number of points generated.	-
x_8: dist.edge	CN	Distance of the point from the nearest boundary.	z_8: dist.edge2
x_9: dist.vert	CN	Minimum distance between the boundary and the cell vertices.	z_9: dist.vert2
x_{10}: dist.cent	CN	Distance of the point from the centre of the finite region.	-
x_{11}: type	DN	Cell type based on how many points of the cell segments intersect the boundary (0, 2, 3, 4).	-

Table 3.1: A list of the variables for both the unit square and the unknown boundaries. The variable labels x and z refer to the variables obtained using the unit square and convex hull boundaries respectively. Variable names and notations with corresponding definitions are given. Variable types are labeled as CN: continuous numerical, DN: discrete numerical, and C: categorical. Dashes represent the unavailability of the usage of a particular variable.

There are several variables shown in Table 3.1 that have direct or indirect relationships in some fashion. This situation is called collinearity or multicollinearity which happens when an independent variable can be linearly predicted by other variable or multiple variables. A more general term for this case is concurvity that happens when a smooth term is predicted by other smooth terms is the model, i.e. the generalized additive model (Morlini, 2006). Collinearity and concurvity cause interpretation issues and unstable predictions that may lead large errors.

The variables x_1 and x_2 are the coordinates of the points, so the distance from the boundary x_8 and the distance from the centre x_{10} use x_1 and x_2 to calculate the Euclidean distances. Hence, there is a situation where one of the variable is being a function of the other variables. However, we do not have any collinearity issues for the unknown boundary case since we do not calculate the distance from the boundary or the centre and use them as variables in the model. The available variables when the boundary is unknown are given at the right column of Table 3.1. Their existence in the models will be discussed in Section 3.4.

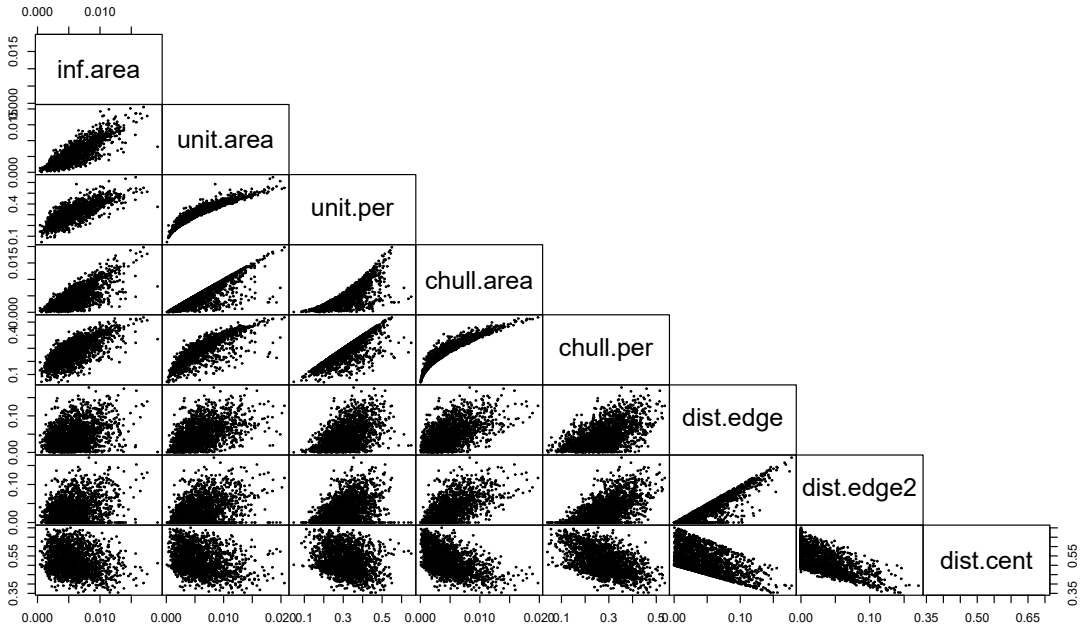


Figure 3.1: Scatterplots of selected variables at the lower triangular part of the scatterplot matrix. Variable labels are given in the boxes on the diagonal panel.

The scatterplot matrix in Figure 3.1 shows the correlations among some variables we are interested in. We aimed to inspect the data and see what kind of patterns are observed among the variables. Therefore, this would help us to decide on the model we would like to use. The plots are created using a randomly chosen sub-sample from the entire data since the visualisation would not be clear for 10^6 observations.

The levels of categorical variables, cell type, being located on the convex hull and number of cell edges are separately checked. In Figure 3.2, the cell area based on the data points that are affected by the unit square and convex hull boundaries are shown based on cell type, being located on the convex hull, and number of cell edges. The cell area gets larger as the number of cell edges increases for both unit square and convex hull boundaries. The cell type has a slight effect on the unit square boundary area where the smallest median cell area is observed for the edge cells. Being on the convex hull causes the cells to have a smaller area than the otherwise case.

3.4 Area prediction for Voronoi tessellation cells

In this section, the study design and methodology of the area prediction is explained. Area prediction procedure involves model fitting using regression models based on

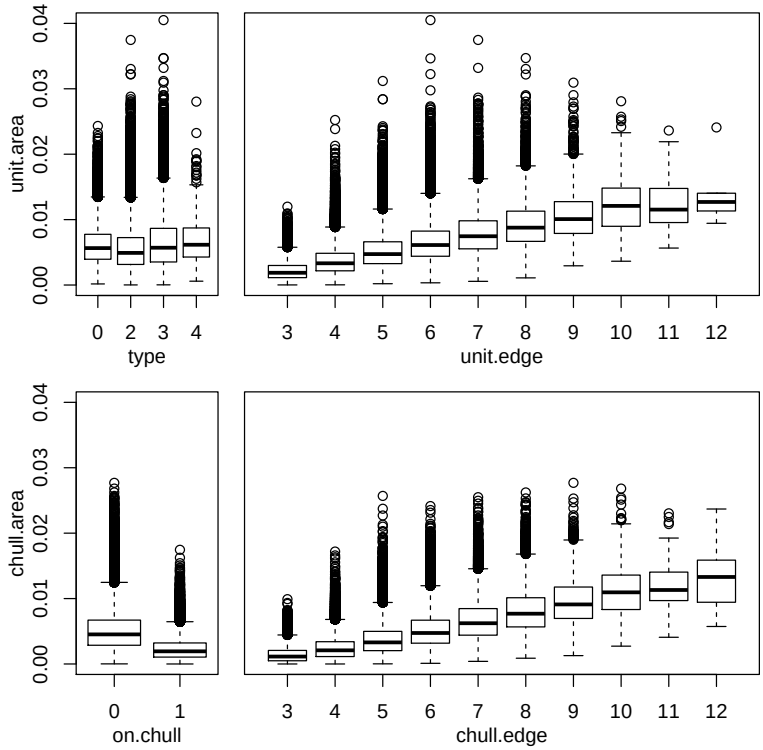


Figure 3.2: Unit square area and convex hull area based on cell type, convex hull points and the number of cell edges.

the variables given in Table 3.1. The infinite plane area labeled as *inf.area* is the response variable we aim to predict, and the remaining variables are used as covariates.

Considering the presence of non-linear relationships between the covariates and the response, there is a need for appropriate regression methods. The linear regression would not capture the non-linear pattern in the data. The techniques such as the polynomial regression that models the response variable by adding $k-$th degree polynomials of the covariates might be useful, however, it may only capture a certain amount of non-linearity. Hence, the usage of a more flexible method such as spline regression is essential.

3.4.1 The generalized additive model

Generalized Additive Models (GAMs), a generalized form of the linear and generalized linear models, is a flexible non-parametric regression method that models non-linear relationship using the sum of smooth functions $\sum\ f_j(\theta_j)$ of the predictors $\{\theta_j\}_{j=1}^{p}$ (Hastie & Tibshirani, 1990; Wood, 2017) by a replacement of the linear components $\sum \beta_j \theta_j$ in the multiple linear regression. The advantage of GAMs is

the flexibility of the non-linear smooth functions f_j that are calculated for each θ_j and added together. The general form of the additive model is expressed as

$$Y_i = \beta_0 + \sum_{j=1}^{p} f_j(\theta_{ij}) + \varepsilon_i \tag{3.1}$$

where f_j denote smooth, non-parametric functions that can take various shapes, p is the total number of covariates, and $\varepsilon_i \sim N(0, \sigma^2)$ is the error term. The smooth functions are generated by many smaller functions called basis functions that can be expanded as

$$f_j(\theta_j) = \sum_{k=1}^{K_j} \beta_{jk} b_{jk}(\theta_j) \tag{3.2}$$

where $b_{jk}(\theta_j)$ denote the basis functions that construct the smooth components, and β_{jk} are the coefficients to be estimated during the model fitting. The K_j is the dimension of the basis function for the component f_j where an optimal choice is necessary since it decides on the wiggliness of the smooth component f_j. A small number of basis functions is likely to miss the wiggly patterns of the data as known as under-fitting, whereas very large number of basis functions results over-fitting that captures very fine details. The sum of the basis functions constructs the smooth functions as shown in (3.2).

When a spline basis is used, we obtain a form of linear model that has a penalty term which can be written as

$$Y_i = \mathbf{B}\boldsymbol{\beta} + \varepsilon_i \tag{3.3}$$

where the matrix $\mathbf{B}$ is created stacking the columns of a basis matrix for each covariate together. Hence the matrix $\mathbf{B}$ evaluates the basis functions for each observation. Then the model fit is achieved by choosing the vector of $\boldsymbol{\beta}$ that minimizes

$$(\mathbf{y}\text{-}\mathbf{B}\boldsymbol{\beta})^\top(\mathbf{y}\text{-}\mathbf{B}\boldsymbol{\beta}) + \boldsymbol{\beta}^\top\mathbf{P}\boldsymbol{\beta} \tag{3.4}$$

where $\mathbf{P}$ is the penalty matrix in a block-diagonal form. The penalty matrix is obtained from individual components of the model such that the $j-$th component is being $\lambda_j \mathbf{D}_j^\top \mathbf{D}_j$, where $\mathbf{D}_j$ is a differencing matrix. Therefore, the following solution for the estimate of the vector of weights $\hat{\boldsymbol{\beta}}$ is obtained as

$$\hat{\boldsymbol{\beta}} = (\mathbf{B}^\top\mathbf{B} + \mathbf{P})^{-1}\mathbf{B}^\top\mathbf{y}. \tag{3.5}$$

The components f_j are identifiable only when a constraint is imposed. We require

the sum-to-zero constraints $\sum_i f_j(\theta_{ij}) = 0$ so that the addition of a constant to f_1 whereas it is subtracted from f_2 without changing the prediction Claire & Neocleous (2019).

The estimation of the smoothing parameter can be done using maximum likelihood (ML), and restricted maximum likelihood (REML) that uses the random effects to estimate smoothing parameters. Wood (2011) showed that their performance is better than other methods such as generalized cross validation (GCV) and AIC.

The generalized additive can also be expressed in the form of

$$g(\mu_i) = \beta_0 + \sum_{j=1}^{p} f_j(\theta_{ij}) \tag{3.6}$$

where $\mu = \mathbb{E}(y|\theta_1, \ldots, \theta_p)$ is the mean and $g(.)$ is a link function function such that $\eta_i = g(\mu_i)$. The possible choices for the distribution of the response include the normal, gamma, Poisson, binomial, inverse Gaussian, negative binomial and quasi distributions and the fitting of the model for different link functions can be done using a local scoring procedure or penalized iteratively re-weighted least squares Hastie & Tibshirani (1990); Wood (2017).

Based on the variables listed in Table 3.1, we denote the response variable *true area* that we aim to predict as $\hat{A}_i$ for $i = 1, \ldots, n$, and the full set of covariates is denoted as $\boldsymbol{\theta} = \{x_1, x_2, \ldots, z_3, z_4, \ldots\}$, hence we have $\{\theta_j\}_{j=1}^{p}$ are the vectors of covariates such that $\theta_1 = x_1, \theta_2 = x_2, \ldots, \theta_p = z_9$ for all $j = 1, 2, \ldots, p$. We will use the θ_j notation in the expression of the model for simplicity, and x_j, z_k will be used as labels of the variables when necessary. Then the full model that we consider takes the form

$$\begin{aligned} \hat{A}_i &= \beta_0 + \sum_{j=1}^{p} f_j(\theta_{ij}) + \varepsilon_i \\ &= \beta_0 + f_1(\theta_{i1}) + f_2(\theta_{i2}) + \cdots + f_p(\theta_{ip}) + \varepsilon_i, \end{aligned} \tag{3.7}$$

Some low-dimensional interaction terms can be added into the model. For instance the interaction function is denoted as $f_{p+1}(\theta_k, \theta_m)$ where $k, m \in [1, \cdots, p]$ and $k \neq m$. The $f_{p+1}(\theta_k, \theta_m)$ term indicates a two dimensional spline for the interacting covariates (James *et al.*, 2013). However, this is only valid when both of the covariates are numerical variables. If one of the variables is categorical, then smooth functions of the numerical variables are separately determined based on each level of the categorical variable.

3.4.2 Study design

The data set obtained from Chapter 2 is for the statistical properties of $n = 10^6$ cells that are sampled from independent realizations. This large data set can be split into training, and validation sets with specifically determined sizes to fit the models and evaluate the performance of the models. The sample sizes for the partitioned data sets are given in Table 3.2. The training and validation sets are independent.

	Training	Validation–1	Validation–2
Size	5×10^5	10^5	10^5

Table 3.2: Sample sizes of training, and validation sets. The numbers refer to the number of randomly sampled cells from independent realisations.

The training set is used for fitting additive models that we call '*base models*'. The Validation–1 is used to test these models and identify influential data points that lead to large prediction error. Then, some influential points are added to the training data and the models are fitted again which we call these models '*augmented models*'. Therefore, the augmented models are aimed to be capable of predicting the '*hard to predict*' cells better than the base models. This is checked in the separate '*left alone*' data set, Validation–2. We evaluate the performances of the *base*, and *augmented* models in Validation–2 and highlight in which situations these models perform better. These steps will be explained in detail in the following sections.

3.4.2.1 Description of training data

It is aimed to fit initial *base* models to the training data based on the model assumption in (3.7). Due to the high number of predictors, we consider the trade-off between the goodness of fit and the model complexity to decide the model that has the most useful variables and leaves the insignificant ones out. Two approaches are tried for variable selection. The first one, is achieved by starting with an intercept only model $y_i = \beta_0$ and iteratively adding and removing variables, namely the stepwise model selection that optimizes the goodness of fit. That is performing a model selection based on the Akaike information criterion (Akaike, 1987), $AIC = 2p - 2\ln(\hat{L})$, where $\hat{L}$ is the maximum value of the likelihood function of the model may be an appropriate approach. There is an option to achieve this in the `gam` package (Hastie, 2020) in `R` using the `step.Gam()` function.

The second approach is called the double penalty approach, introduced by Marra & Wood (2011) and found to perform significantly better along with another proposed

method that is shrinkage-based than the competing methods in a comparative study. The space of a spline basis is expressed as a sum of two components where the first term is based on the functions in the penalty null space and the second is based on the penalty range space. Marra & Wood (2011) explains the approach as follows; functions in the range space can be shrunk to zero via a high penalization by the smoothing penalty but the function component in the null space is unchanged. Therefore, penalization of the null space is required in order to shrink the entire spline component to zero. Their double penalty approach applies penalty for the null space hence the smooth component can be eliminated. Double penalty is applied for each smooth function and the functions with smoothing parameters approach to infinity (such as the straight lines), will be removed the model. The `R` package `mgcv` introduced in Wood (2015) has functions to fit additive models with an option of variable selection by double penalization as described in Marra & Wood (2011). This approach has significantly lower computational cost compared to the stepwise model selection hence it is adopted for variable selection.

The model fitting and variable selection in the training data is performed using available functions in the `mgcv` package. The model is defined as `gam(y` $\sim$ `s(theta_1) + s(theta_2) + ⋯ )` in the function where `s(`$\cdot$`)` denotes the smooth terms. Also, some interaction terms can be added as $f(\theta_3, \theta_8)$ for the interaction of (unit square area) and (the distance from the nearest edge) that fits a two-dimensional surface. The interaction of the (convex hull area) and a categorical variable (being located on the convex hull) takes binary values $f(\theta_{12}, \theta_{15})$ can also be added. These interactions are defined in the function by writing `s(theta_3, theta_8)` and `s(theta_12, by = theta_15)` respectively.

Instead of using the entire training set at once, we divide it into training subsets since the size of the training set permits this flexibility. Therefore, we may fit model using each training subset and these models can be used in an ensemble learning approach. We first randomly sample smaller training subsets with equal size $n_{train} = 5000$ without replacement that gives 100 training subsets. Then the GAMs are fitted to each training subset and these models are used for area prediction in the Validation–1. We check the individual and combined performances of the models in the Validation–1. Having numerous models from independent training sets for the same prediction purpose is also useful to reduce the sampling bias and the variability in the predictions.

3.4.2.2 Description of validation–1 data

The initially fitted base models in the training subsets are used for the prediction of the cell area in the Validation–1, and an unbiased evaluation of the model fit is made. Next, we investigate the influential points that we define as the observations that cause large prediction error. These observations are the ones that the model is not able predict well. One could suspect that this is due to the lack of the data with similar characteristics in the training set. Hence we identify the influential points and investigate whether they have characteristics in common (such as all being corner cell, or extremely large cell, etc.). Moreover, we consider enriching the training sets by adding the influential points and fitting previously mentioned *augmented models* using the augmented training sets. The *augmented models* are aimed to improve the accuracy of the *base models*.

The predicted cell area can be denoted as $\hat{A}_i$ for $i = 1, 2, \ldots, n_{V_1}$ where n_{V_1} is the size of the Validation–1. Since the area prediction is made using 100 individual models, the predicted area can be denoted as $\hat{A}_{it}$ where $t = 1, 2, \ldots, 100$ indicate each training subset. We may represent the predicted area using all models in a matrix since there will be a practical use of it later. The matrix of predicted cell areas $\hat{\mathbf{A}}$ can be expressed as

$$\hat{\mathbf{A}} = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} & \hat{A}_{13} & \cdots & \hat{A}_{1,100} \\ \hat{A}_{21} & \hat{A}_{22} & \hat{A}_{23} & \cdots & \hat{A}_{2,100} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{A}_{n_{V_1}1} & \hat{A}_{n_{V_1}2} & \hat{A}_{n_{V_1}3} & \cdots & \hat{A}_{n_{V_1}100} \end{bmatrix}, \tag{3.8}$$

where the columns indicate the vector of predicted area using the t-th model for the Validation–1. Each row is the predicted area for the observations of Validation–1. For instance, the first row $\hat{A}_{11}, \hat{A}_{12}, \ldots, \hat{A}_{1,100}$ indicates the prediction of area for the 1st observation of Validation–1 using individual *base models* which are 100 in total.

3.4.2.3 Influential points

A classical modelling approach that fits models in the training data and evaluate the performances of the models in a test or validation data is not our sole target. We would also like to create improved versions of the initial models that have a better predictive performance. To achieve this, we propose a way that takes *hard to predict* observation in the Validation–1 data into account. Hard to predict observations refer to the area of Voronoi cells which the model is the least capable of

predicting. We identify these data points by checking the observations that causes the largest absolute prediction error. Since the individual models are applied to the Validation–1 data separately, each model has its own set of *hardly predicted* data points. Therefore, we call this sets of points as the *influential points*.

The separate sets of influential points, chosen by individual models, are going to be combined with the training data sets which constructs the augmented training data sets. Then the new models are fitted using the augmented data which we call *augmented models*, and keep the previously fitted *base models*. The augmented models are aimed to be the updated versions of the base models. These type of modeling approaches are discussed in Hofner *et al.* (2014) where a model-based boosting method is introduced and used to fit boosted additive models by optimizing the general risk functions using penalized least squares estimates as base-learners. Also the ensemble learning approach we use is similar to the random forest technique for classification and regression introduced by Ho (1995) and Liaw *et al.* (2002).

The columns of $\hat{\mathbf{A}}$ in (3.8) indicate the predicted cell area using the initially fitted models on Validation–1. Each column is obtained using a separate model. Therefore, for each column a set of influential points can be identified. Here, we define the set of influential points as being the data points which has the largest 500 absolute prediction error. By choosing this threshold, we aim to select the most influential points.

The next stage will be using the base and augmented models in our second validation data *Validation–2* that is going to be discussed in Section 3.4.2.4 and their performances are going to be compared in the results section. We also investigate what is special with these influential points and check whether they have common features. Hence, we can have a better understanding of what type of cells cause largest absolute prediction error.

3.4.2.4 Description of validation–2 data

A hold out data set with the same size as Validation–1 is created for further inference about the base and augmented models. This is the stage that the two modeling approaches are compared and the most appropriate model is suggested. Area prediction in Validation–2 is made using individual base and augmented models, and the individual performances, and their overall performances are aimed to be evaluated. Besides the design of the training, Validation–1 and Validation–2, the specification of the distribution family of the residuals and link functions has an importance. As the default choice, the residual distribution has a Gaussian family and

identity link function. Hence, it is worth considering the alternative assumptions as well.

3.5 Results

This section presents and discusses the results of base model fitting, identification of influential points using Validation–1, data augmentation, fitting the augmented models, and eventually the comparison and evaluation of the base and augmented models in Validation–2. We consider fitting models and presenting the results for two types of boundary scenarios separately: the unit square boundary, and the unknown boundary where the convex hull of points is used as the boundary in Sections 3.5.1 and 3.5.2 respectively.

3.5.1 Unit square boundary case

This section presents the results for the unit square boundary case which is used as the sampling region for the data we use from Chapter 2.

3.5.1.1 Training base models

The additive model in (3.7) is fitted to 100 randomly sampled training data sets, and hence we obtained individual models for each training subsets. Model fitting is performed by taking the variable selection into account. We are interested in seeing how frequently the variables are chosen based on separate models. Then we may have an idea about the importance of the frequently selected variables.

In Figure 3.3, the occurrences of the variables based on 100 base models are listed. Note that (:) stands for the interaction of variables. We use the x and y labels of the variables referring to Table 3.1. The ranking of the selected variables highlights some important variables in the base models. For instance, (unit square boundary area) and its interaction with the (cell type), that is $x_3 : x_{11}$, are selected in all models. Similarly the interaction of (cell type) and the (distance from the boundary) that is $x_8 : x_{11}$. Also, z_8 (distance from the convex hull boundary), x_4 (unit square boundary perimeter), the interaction of (convex hull boundary area) and (being on the convex hull) that is $z_3 : z_6$, and z_4 (convex hull boundary perimeter) are selected in most of the models. On the other hand, the least important variables are x_{10} (distance from the centre), x_8 (distance from the unit square boundary), x_1 and x_2 that are (the point coordinates), and x_7 (the number of points).

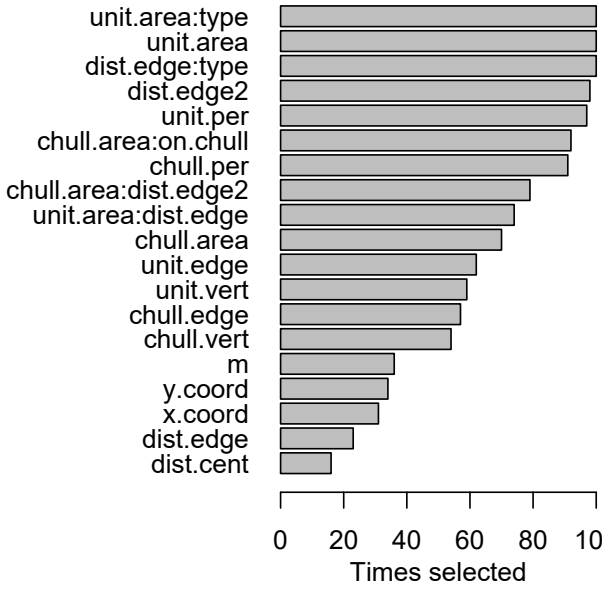


Figure 3.3: Variables and interaction terms, and how many times selected in the base models.

Each base model can be expressed as a function as shown in the model in (3.7) that contains all variables and some interactions. The useless variables in the models will still appear in the equations, but since their coefficients are penalized to zero, they have no effect in the model.

To see the trajectories of the individual models in the prediction of cell area, we may visualize the estimated smooth components. The smooth components for the covariates are given in Figure 3.4. Each plot is associated with a covariate that is labeled in the title and many curves appear in each plot. The curves or straight lines are the estimated smooth components for individual models. The smooth components with gray colour are obtained from the base models, and the black lines are for the augmented models which will be discussed in Section 3.5.1.2. The y-axis always shows the values of the response variable and the x-axis is for the value range of the covariate. Hence, we will be able to compare the smooth components obtained from the base and augmented models.

Flat lines in particular variables, such as the point coordinates and the number of points, indicates the penalized covariate that has no effect on the response variable. The smooth components that show different characteristics have the major effects in the base models. The smooth components that have the curvy shapes are the ones that are frequently selected in the models whereas the least selected variables are the flat lines. It is also observed that some individual models behave very differently

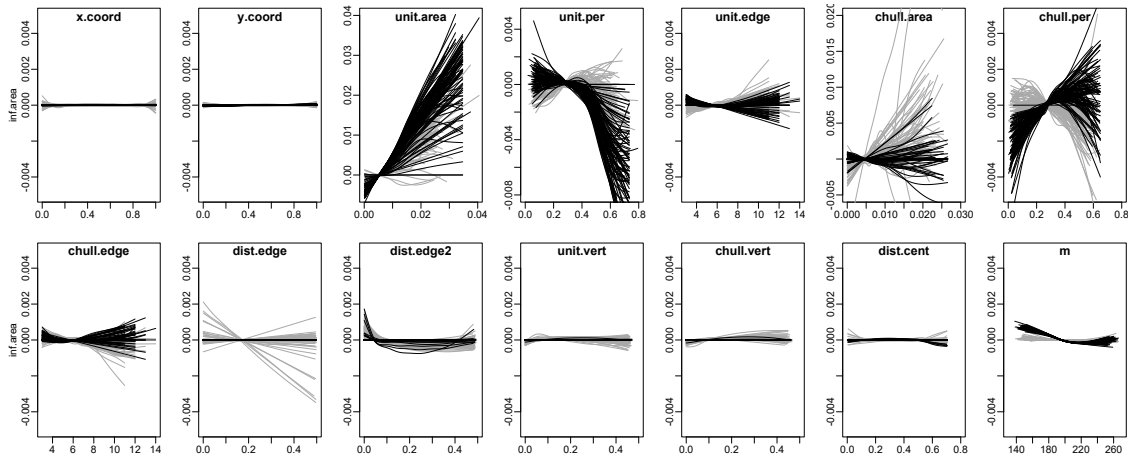


Figure 3.4: Estimated smooth components of the GAMs in the individual base and augmented models. Gray lines are the smooth components for the base models that are overlaid for 100 models, and the smooth components for the augmented models are shown in black.

to the majority of the other models. It is possible to make this conclusion based on the smooth lines that do not follow the pattern of the majority.

The GAM is a generalization of the linear model, hence, it is natural to have a mixture of smooth and linear components in a model even though our models do not include any linear terms explicitly. The x_8 (distance from the unit square boundary) shows an example of this case where the lines are nearly straight in all models. Note that there are only the numeric variables and no interaction terms shown in this figure since the other types of interactions should be presented in different ways. For instance, the interaction of two numeric variables in the model generates a two-dimensional spline, and the interaction of a numeric and a categorical variables estimates separate smooths based on each category level.

Lastly, let us check the residual patterns of the base models. Figure 3.5 shows the normal quantile plot of base models in grey lines. The variation is high at the lower and higher values of the residuals within models, but the lines get closer around zero. Also, there are many values close to zero in the sample quantiles, that is because many points that are close to the centre of the spatial region are not affected by the boundaries and the prediction is very accurate.

3.5.1.2 Training augmented models

In this section, we explain the procedure of training the augmented models. The augmented models are a continuation of the base models which are fitted using

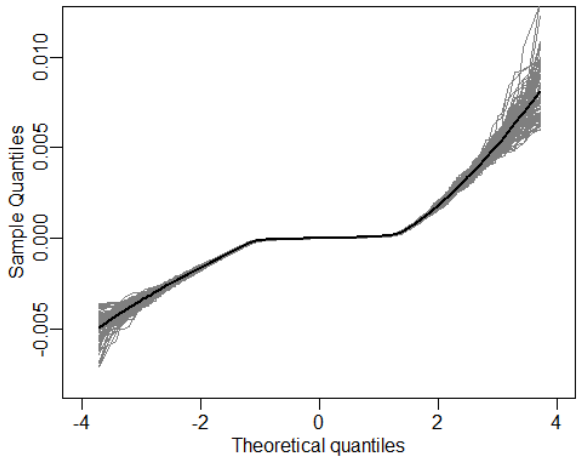


Figure 3.5: The normal quantile-quantile plot of residuals versus fitted values in base models in gray lines, and the averaged values as the black line.

the training subsets and used in Validation–1 to identify the influential points that is explained in Section 3.4.2.3. Augmented models are fitted using the augmented training data which is obtained adding the influential points identified in Validation–1. We first explain how to identify the influential points, and summarize their characteristics, and explain how the augmented models are fitted.

Recall the matrix $\hat{\mathbf{A}}$ of predicted cell area in equation (3.8) where each column is the vector of predictions using individual base models. To identify the influential points, we base on the criterion of absolute error that can be denoted as $|A_{it} - \hat{A}_{it}|$ for observations $i = 1, 2, \ldots, n_{V_1}$ in Validation–1, and for models $t = 1, 2, \ldots, 100$. For each column t, we select from the points that cause the largest absolute errors. Using each base model in Validation–1, we identify 500 points that cause the largest absolute error. Therefore, each base model identifies a set of points that are the most influential. Each set of influential points can be denoted as as $\mathbb{I}_1, \mathbb{I}_2, \ldots, \mathbb{I}_{100}$. In this case, $\mathbb{I}_1$ is the set that contains influential points identified using the first base model and would contain some points such as $\mathbb{I}_1 = \{x_{125}, x_{431}, x_{1480}, \ldots\}$.

It is important to note that sets $\mathbb{I}_1, \mathbb{I}_2, \ldots, \mathbb{I}_{100}$ are independently generated, therefore, these sets are likely to contain influential points in common. Hence, we can check the influential points in terms of the number of sets in which they are observed. The union of separate sets can be denoted as $\mathbb{I} = \mathbb{I}_1 \cup \mathbb{I}_2 \cup \cdots \cup \mathbb{I}_{100}$ where $\mathbb{I}$ is the set that has all influential points. Figure 3.6 shows the number of times each influential point is identified in the y-axis, and the x-axis shows the index numbers of influential points. Note that it is not the index numbers i in $\{x_i\}_{i=1}^{n_{V_1}}$, we sorted the influential points based on the number of base models they are identified by, and

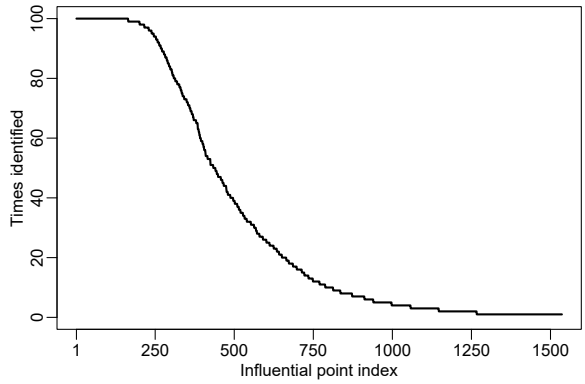


Figure 3.6: Index of the influential points and how many times they are identified as influential.

gave new index numbers $k \in 1, 2, \ldots, K$. Therefore, x-axis in Figure 3.6 denotes the k. The unique number of influential points in $\mathbb{I}$ is approximately $K \approx 1500$, and around 200 points are identified as influential point by all base models.

It is important to check what is special with these influential points. The summary statistics of the influential points and all other points are presented for comparison in Table 3.3. The first row for each variable panel shows the summary statistics of all points and the second row shows the influential points coloured in blue. Many comparisons can be made based on this table. The mean properties are significantly different for the influential and all points. The distance between the influential points to the boundaries are much smaller compared to the all points, which demonstrates the influential points are located closer to the boundaries. The area and perimeter of influential points are larger than the and points and influential points have fewer cell edges. Also the standard deviations of these properties are substantially different.

Cell types of the influential points show differences compared to all points. The proportions of cell types are calculated for influential, and all points are shown in Table 3.4. The dominance of type–2 (edge cells) is obvious in the influential points. Also, the proportions of type–3 (corner) and type–4 (corner+) cells which are close to the corner but has two cell vertices lying on each perpendicular boundary is higher. Only a small number ($\approx 1\%$) of interior cells are identified as influential.

Similar differences on the proportions of convex hull points are seen in Table 3.5. Most of the influential points are located on the convex hull boundary whereas it is not the case when all points are considered.

	Min	Max	Mean	SD	Range
inf.area	0.0001	0.0197	0.0050	0.0026	0.0196
	0.0006	0.0250	0.0115	0.0039	0.0244
unit.area	0.0000	0.0197	0.0050	0.0027	0.0196
	0.0006	0.0284	0.0062	0.0035	0.0278
unit.per	0.0222	0.6085	0.2845	0.0723	0.5863
	0.1203	0.6696	0.3404	0.0828	0.5493
unit.edge	3	13	5.759	1.306	10
	3	8	4.923	1.044	5
dist.cent	0.0020	0.7056	0.3793	0.1412	0.7036
	0.4258	0.6978	0.5566	0.0696	0.2720
dist.edge	0	0.4981	0.1690	0.1175	0.4981
	0.0002	0.0986	0.0190	0.0161	0.0985
m	139	264	200.05	14.036	125
	157	239	193.50	14.254	82
chull.area	0.0000	0.0197	0.0047	0.0026	0.0197
	0.0001	0.0147	0.0041	0.0027	0.0146
chull.per	0.0100	0.5965	0.2744	0.0730	0.5865
	0.0583	0.4970	0.2791	0.0876	0.4387
chull.edge	3	13	5.7826	1.294	10
	3	8	5	1.144	5
dist.edge2	0	0.4932	0.1564	0.1191	0.4932
	0	0.0757	0.0068	0.0117	0.0757

Table 3.3: Summary statistics of the variables in the validation data. The first row for each variable panel is the results for all points and the second row for the influential points coloured in blue.

Sets of influential points $\mathbb{I}_1, \mathbb{I}_2, \ldots, \mathbb{I}_{100}$ identified by the base models are moved from Validation–1 to the corresponding training data sets $D_{tr_1}, D_{tr_2}, \ldots, D_{tr_{100}}$. Then GAMs are refit using these augmented training data that contain the initial training data sets and the influential points. The new models, namely the augmented models are expected to improve some features of the base models.

The variable selection results are given for the augmented models in Figure 3.7. These results are an extension to the results in Figure 3.3 with the variable selection results for the augmented models given on the right. Some variables keep their position in the ranking, and some changed position. For instance, x_7 (number of points) was one of the least selected variables in the base models, however, it was

type	0	2	3	4
All points	0.761	0.219	0.020	0.0006
Influential points	0.008	0.603	0.350	0.0400

Table 3.4: Proportion of the cell types (0: interior, 2: edge, 3: corner, 4: corner+) for all points, and the influential points.

on.chull	0	1
All points	0.933	0.067
Influential points	0.351	0.649

Table 3.5: Proportion of the cells located on the convex hull (0: No, 1: Yes) for all points, and the influential points.

selected by all augmented models. Also, one of the variables selected in all base models x_3 (unit square area), and z_8 (distance from the convex hull boundary) are less important variables in the augmented models. The interaction terms x_8, x_{11} (interaction of the distance from the unit square boundary and type), x_3, x_{11} (unit square area and cell type) are included in all base and augmented models so they have importance in both methods.

The overlaid smooth components for single variables from individual augmented models are also shown in Figure 3.4 where the estimated components for the base models are given in the background in gray. Most black curves lie between the minimum and maximum value of the x-axis. This means the larger observations of the variables are available in the augmented data. The gray lines do not always lie within the same range of x-axis because the initial training data did not have that observations, but addition of the influential points to the training data provided these observations. This means while the base models do extrapolation, augmented models are doing interpolation for the influential points. The gray and black curves for a specific variable, the unit square, shows that the lengths of many of the curves in the base models are different. The issue is reduced in the augmented training data by the placement of the influential points because the influential points enriched the training data with infrequently seen observations.

There are slight changes on the patterns of the estimated smooth curves for the base and augmented models. Also, penalized variables are closer to a straight line at zero in the augmented models where they were slightly off from being flat in the base models. The residual pattern of the augmented models using the augmented

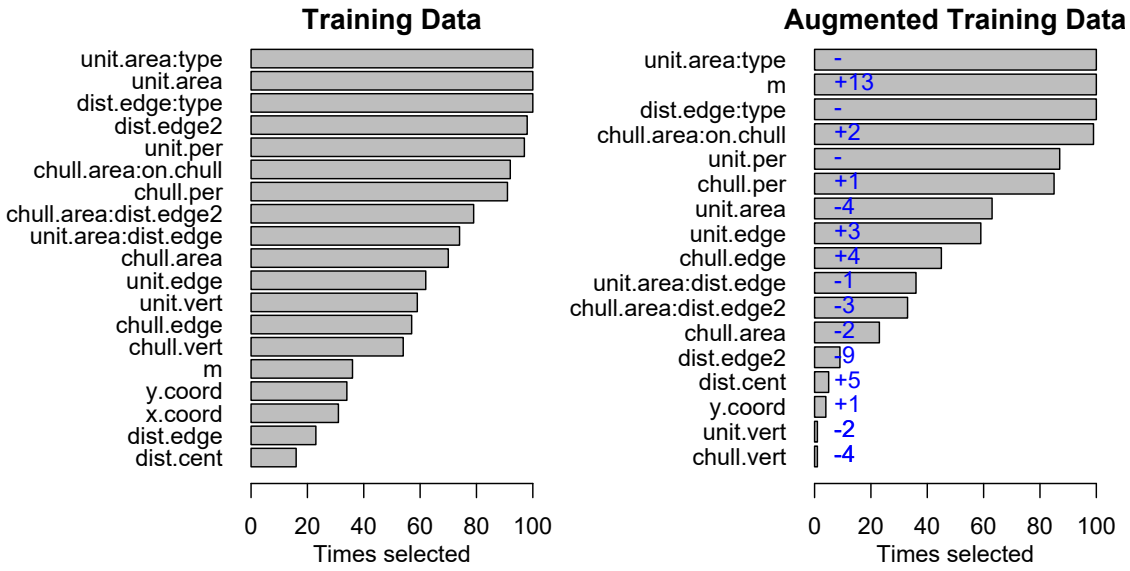


Figure 3.7: Selected variables in the unit square boundary models and the number of times each term is selected. Results are given for the base models (left) and augmented models (right). The change in the ranking for each term is highlighted in blue and the total number of times selected is given in parentheses.

training data is given in Figure 3.8 which shows some difference from the base models.

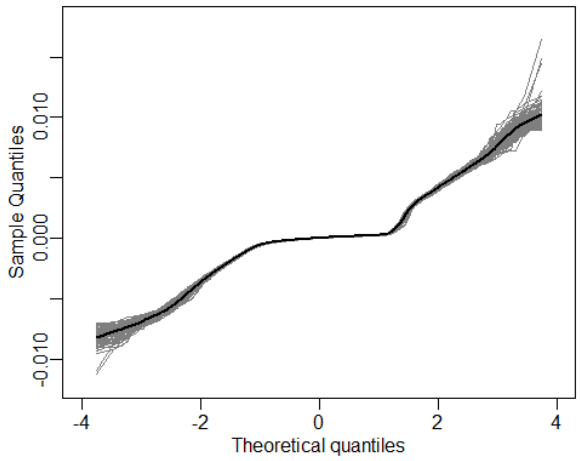


Figure 3.8: The normal quantile-quantile plot of residuals versus fitted values in augmented models in gray lines, and the averaged values as the black line.

We now move on to the final evaluation of base and augmented models in an independent data set. The Validation–1 data is only used for the base models to identify the influential points which are moved to the training data for augmentation. Hence, Validation–1 is lacking some important data points (influential) and can no longer be used. As an independent data set, Validation–2, will be used

for the further evaluation and inference on the base and augmented models. The Validation–2 has a size 10^5.

An ensemble approach is used for the area prediction in Validation–2 to obtain a better predictive performance of the individual models. The 100 individual base and augmented models are used to predict the area, and the ensemble prediction is calculated by averaging the estimates over data points. Using general notations, the matrices for the prediction in the Validation–2 data be denoted as $\hat{Y}'$ and $\hat{Y}^*$ for base and augmented models respectively

$$\hat{Y}' = \begin{bmatrix} \hat{y}'_{11} & \hat{y}'_{12} & \hat{y}'_{13} & \cdots & \hat{y}'_{1t} \\ \hat{y}'_{21} & \hat{y}'_{22} & \hat{y}'_{23} & \cdots & \hat{y}'_{2t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{y}'_{n_{v_2}1} & \hat{y}'_{n_{v_2}2} & \hat{y}'_{n_{v_2}3} & \cdots & \hat{y}'_{n_{v_2}t} \end{bmatrix}, \quad \hat{Y}^* = \begin{bmatrix} \hat{y}^*_{11} & \hat{y}^*_{12} & \hat{y}^*_{13} & \cdots & \hat{y}^*_{1t} \\ \hat{y}^*_{21} & \hat{y}^*_{22} & \hat{y}^*_{23} & \cdots & \hat{y}^*_{2t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{y}^*_{n_{v_2}1} & \hat{y}^*_{n_{v_2}2} & \hat{y}^*_{n_{v_2}3} & \cdots & \hat{y}^*_{n_{v_2}t} \end{bmatrix} \tag{3.9}$$

for points $i = 1, \ldots, n_{v_2}$ and models $j = 1, \ldots, t$, where $n_{v_2} = 10^5$ and $t = 100$. The ensemble prediction is calculated as

$$\tilde{Y}' = \Big(\frac{1}{100} \sum_{j=1}^{t} \hat{y}'_{1t} \quad \frac{1}{100} \sum_{j=1}^{t} \hat{y}'_{2t} \quad \cdots \quad \frac{1}{100} \sum_{j=1}^{t} \hat{y}'_{n_{v_2}t} \Big)^\top \tag{3.10}$$

$$\tilde{Y}^* = \Big(\frac{1}{100} \sum_{j=1}^{100} \hat{y}^*_{1t} \quad \frac{1}{100} \sum_{j=1}^{t} \hat{y}^*_{2t} \quad \cdots \quad \frac{1}{100} \sum_{j=1}^{t} \hat{y}^*_{n_{v_2}t} \Big)^\top \tag{3.11}$$

where $\tilde{Y}'$ and $\tilde{Y}^*$ are the vectors of predictions for base and augmented models respectively.

To compare the performance of prediction, the prediction error over the spatial region is a good way to see where these methods perform well and badly. Area prediction is made for a total of $n_{v_2} = 10^5$ data points. The squared error at each point is calculated as $SE' = (y_i - \tilde{Y}')^2$ and $SE^* = (y_i - \tilde{Y}^*)^2$ for base and augmented ensemble predictions. The mean squared error (MSE) is calculated by averaging the squared error over pixel bins and it is visualized in Figure 3.9. The unit square region has symmetric properties on each quadrant, so the spatial region is folded to increase the data points in each bin using the data points at the relevant bin.

The high MSE occurs near the boundaries and highest at the corners for both base and augmented models in (a) and (b) in Figure 3.9. The pixel bins in (a) have lighter colour compared to (b) near the edges that means the MSE is smaller for base models. The ensemble predictions perform very well if the points are located far from the boundary. In (c) and (d), the squared error is calculated as

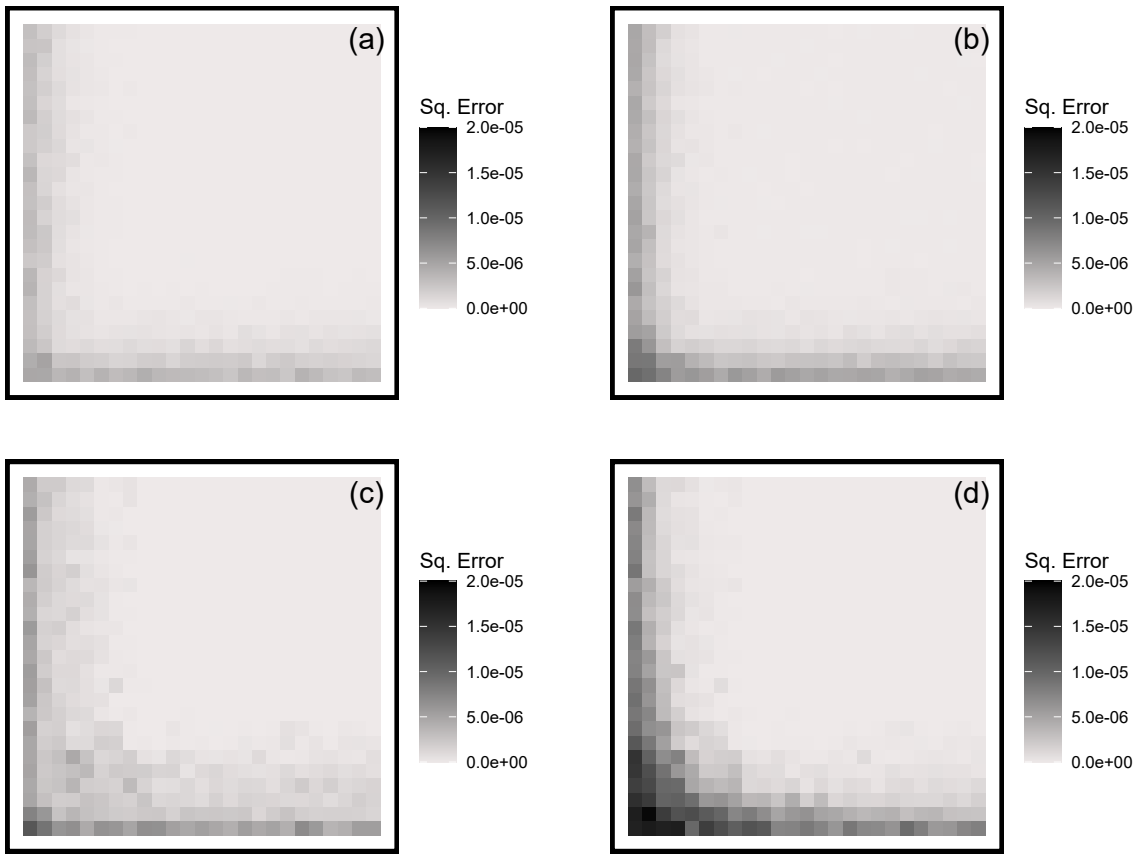


Figure 3.9: The squared error of infinite plane area and predicted area averaged over pixel bins for (a): base models, (b): augmented models, (c) unit square area, and (d) convex hull area.

the difference between the infinite plane area, and the observed area due to both boundary scenarios. The MSE in (c) and (d) seems higher compared to (a) and (b) that shows the predicted area is closer to the truth. The darkest colour is at the edges and the corner of the observed area due to the convex hull boundary in (d).

We may also check the numeric values of the MSE. Table 3.6 shows the MSE calculated globally, and for interior and edge parts of the region. The first two columns are results for the base models and the last two columns are for the augmented models. For these two modelling types, we obtained the results for the full models and reduced models separately. The full model considers all variables listed in Table 3.1. On the other hand, the reduced model excludes the variables coordinates x_1 and x_2, and the distance from the centre x_{10} that have dependence with distance from the edge x_8 that is kept in the model.

We observe from Table 3.6 that the results for the full models and reduced models

are extremely similar and even almost identical for the base models. For the augmented models, the reduced models gave just slightly higher MSE. We also checked the ranking of variables based on how many times they appear in the reduced models and showed in Figure A.2 and compared to the full models from Figure 3.7. The rankings are very similar and only a few minor differences are seen. Since the variable selection method also penalized the correlated variables in the full model, similar results are expected. However, the collinearity is an essential issue that should be considered in modelling.

	Base		**Augmented**	
	Full Models	Reduced Models	Full Models	Reduced Models
Global	0.5441	0.5440	0.8525	0.8648
Interior	0.0033	0.0033	0.0418	0.0449
Edge	1.9032	1.9031	2.9001	2.9357

Table 3.6: MSE values calculated in the Validation-2 data using the models that include all variables, and the models that do not contain the correlated variables. The MSE values are multiplied as MSE$\times 10^6$.

The surface plots are not completely informative since it is hard to evaluate different methods by eye. Therefore, we check the MSE along different surface transects in Figure 3.10. Cross checking of the surface plots and the transect plots verifies the previous conclusions and adds more details about the results. We see that the ensemble predictions for both base and augmented models are in the first two quartiles of boxplots. More importantly, the base ensemble predictions always give smaller MSE for all transects and it is the best among all individual models. When the outlier points of the individual model prediction boxplots are checked, some of the individual base models give extremely larger MSE than the augmented models. Therefore, the augmented model approach reduces the maximum errors in the data, but the overall performance of the base models is better. In the edge transect, some of the boxplots have unusual shapes since there are individual base models insufficient to predict the cell area well. But the ensemble prediction performs satisfactorily accurate.

The error plots are shown in Figure 3.11 and 3.12 where the transect boxplots show that base ensemble predictions are spread around zero but the augmented ensemble models mostly predict values to be larger than the truth.

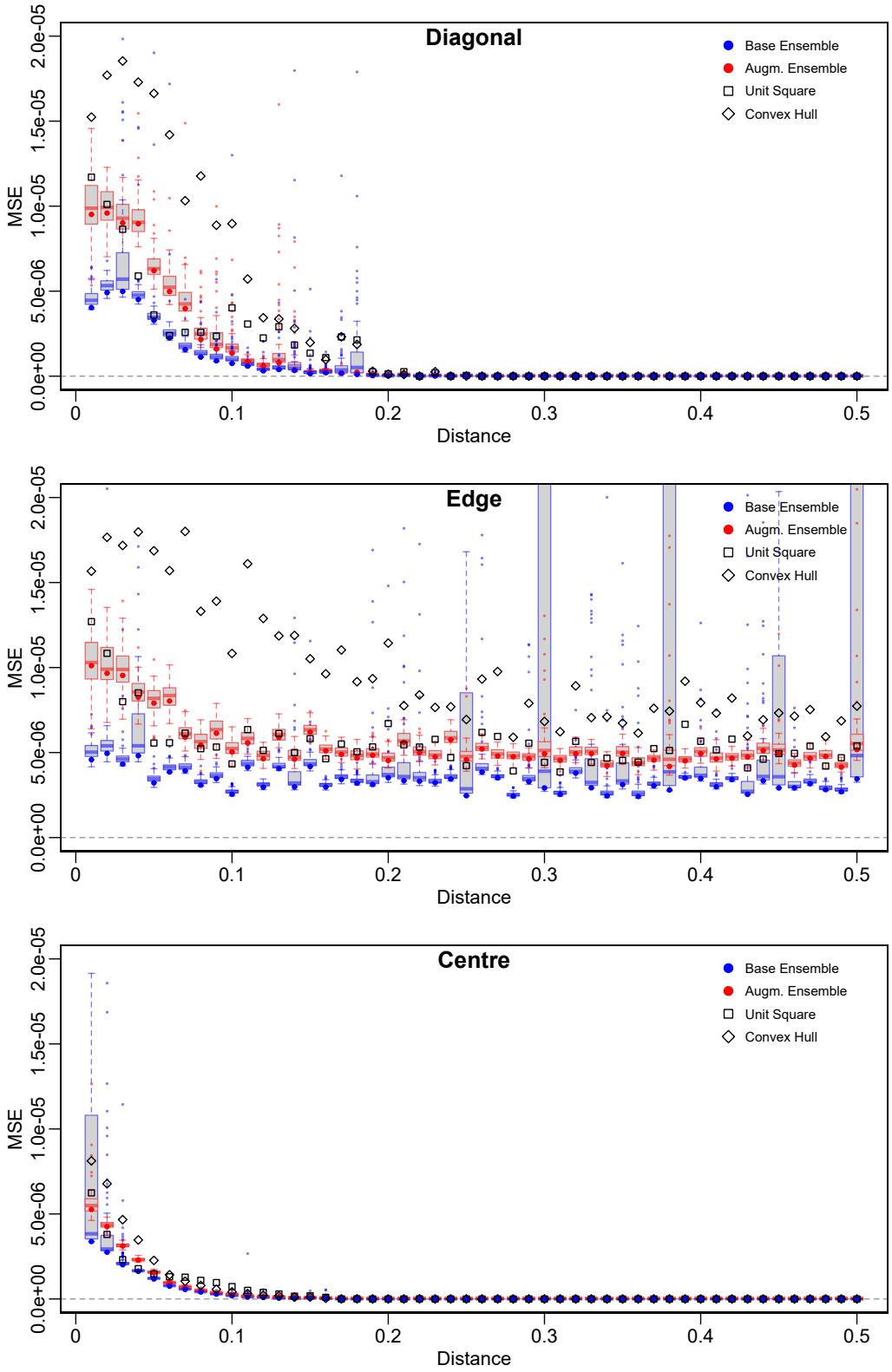


Figure 3.10: The boxplots of MSE for predictions from individual base models (blue) and augmented models (red) over pixel bins along different transects. The ensemble predictions are shown with the solid points in the same colour at each transect bin. The MSE for unit square and convex hull area are shown with square and diamond shaped points.

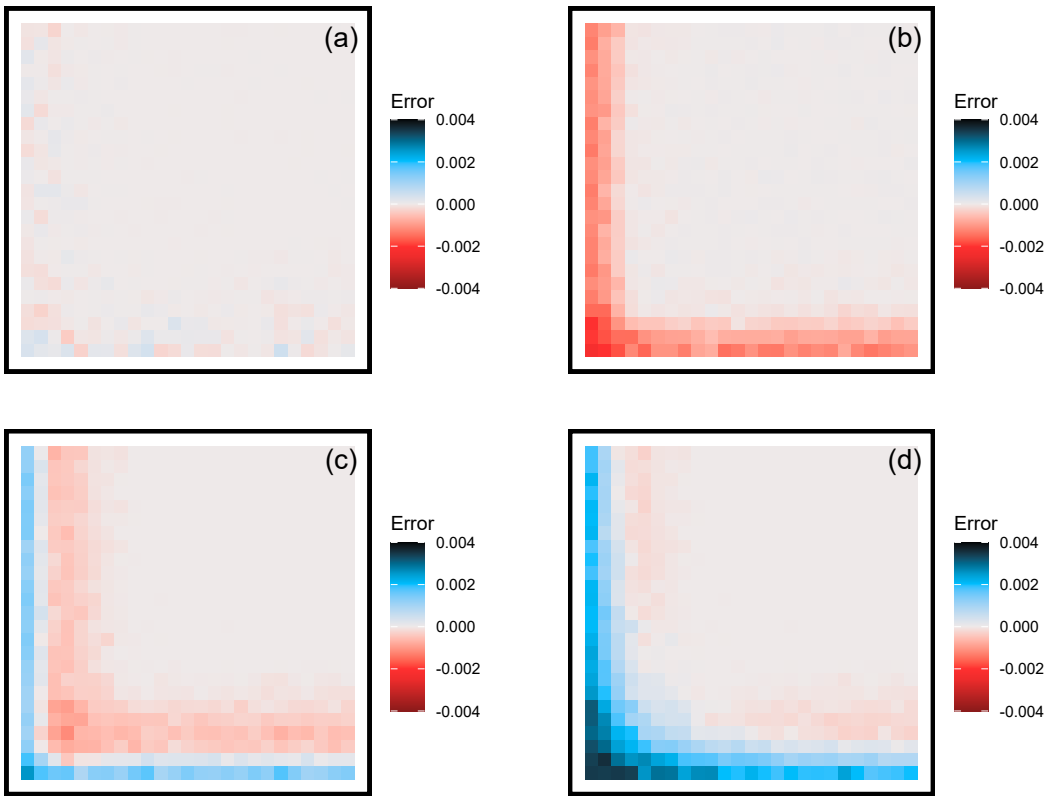


Figure 3.11: The error of infinite plane area and predicted area averaged over pixel bins for (a): base models, (b): augmented models, (c) unit square area, and (d) convex hull area.

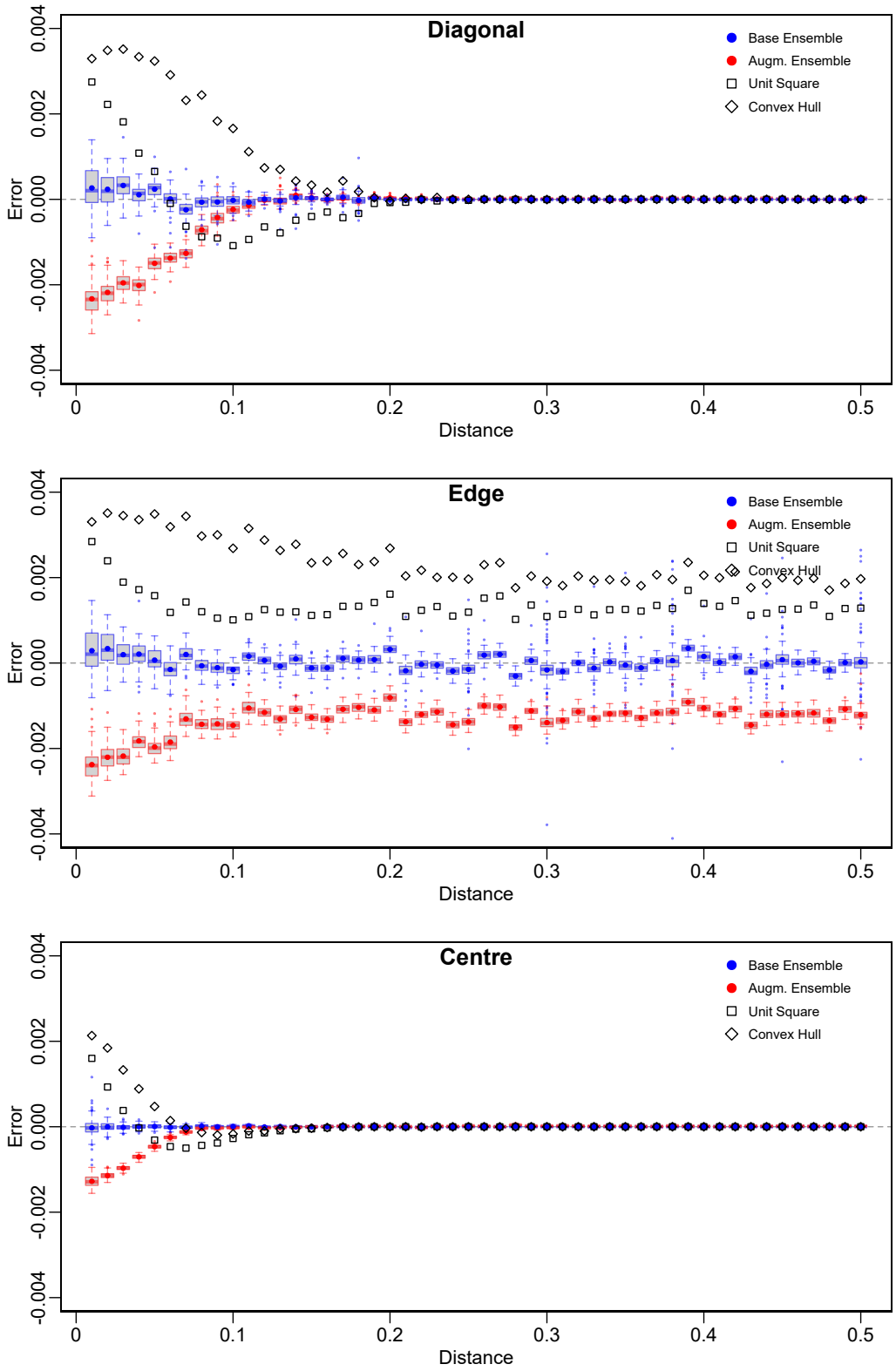


Figure 3.12: The boxplots of mean error for predictions from individual base models (blue) and augmented models (red) over pixel bins along different transects. The ensemble predictions are shown with the solid points in the same colour at each transect bin. The MSE for unit square and convex hull area are shown with square and diamond shaped points.

3.5.2 Unknown Boundary case

This section considers the case where no boundary information is available. In this case, we take the convex hull of the points and use it as the boundary. The model fitting procedure is performed for the unknown boundary case using the available variables from Table 3.1 (right column). A similar strategy is followed to fit the base models as in Section 3.5.1. Then these models are used to predict the cell area in Validation–1 and influential points are identified. Finally, the influential points are added to the training data sets to fit augmented models.

The variable selection results for the base and augmented models are given in Figure 3.13. z_4 (convex hull perimeter), z_5 (convex hull number of edges), $z_3 : z_6$ (the interaction of convex hull area and being on the convex hull), $z_3 : z_8$ (the interaction of convex hull area and distance from the convex hull boundary) are the most selected variables in the base models. z_4 and z_5 kept its position in the ranking for the augmented models and the interaction terms slightly went down although the number of times selected increased. There is a significant jump for x_7 (the number of points) which was also the same for unit square boundary results, and z_8 (distance from the boundary) are the least important terms in the augmented models.

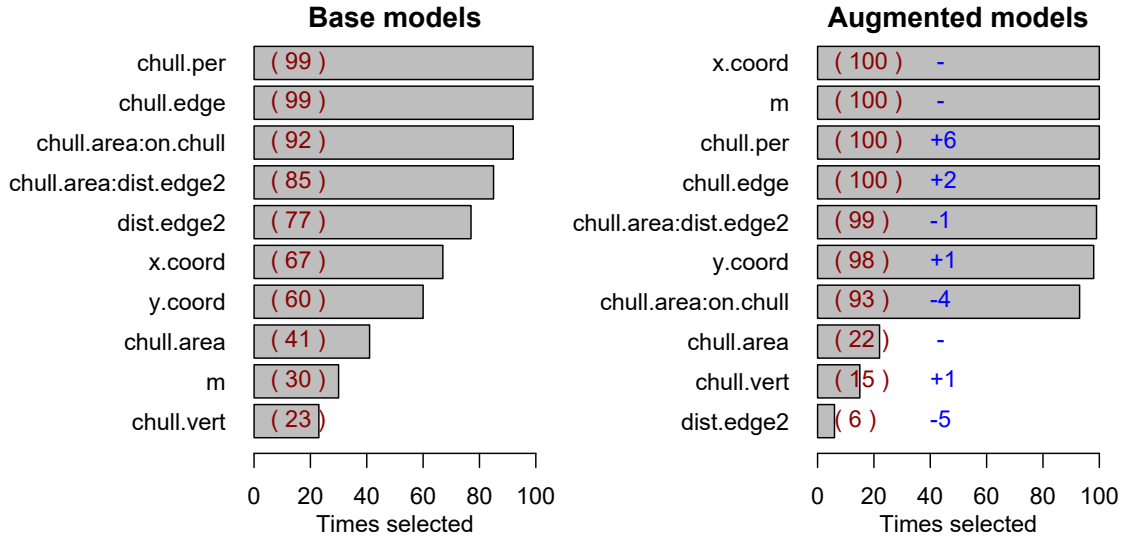


Figure 3.13: Selected variables in the unknown boundary models and the number of times each term is selected. Results are given for the base models (left) and augmented models (right). The change in the ranking for each term is highlighted in blue and the total number of times selected is given in parentheses.

The equivalent process of the identification of the influential points from Section 3.4.2.3 is done for the unknown boundary case models as well. Figure 3.14

shows the number of points which were identified as influential by all models was around 250 which is very close to the previous results in the unit square boundary in Figure 3.6. Figure 3.6 and 3.14 summarizes that there are many influential points which are commonly identified by all 100 models. This shows that influential points may have features in common and the data augmentation process we perform has an importance to deal with the predictive performances of the base models. There are approximately 1000 influential points, and 816 of them are identified as influential in the unit square base models. This demonstrates that mostly the same points are identified as influential. Therefore, another analysis on the influential points is not necessary.

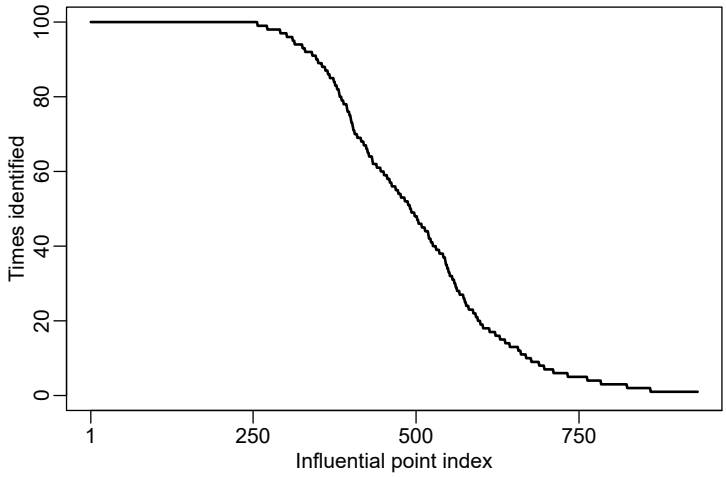


Figure 3.14: Index of the influential points and how many times they are identified as influential.

The estimated smooth components in Figure 3.15 shows that curves for base models in gray and augmented models in black which shows some differences. The data augmentation here also provides the training data to have some infrequently observed data points. Hence the augmented models are trained to estimate such points more accurately. The residuals patterns are very similar to the unit square boundary results.

The spatial patterns of the MSE in Figure 3.17 and at the transects in Figure 3.18 have some overall similarities but also differences in specific parts. For instance, the base and augmented models almost perform equally at the corner. This is noticeable in Figure 3.17 (a) and (b) near the corner, and top panel in Figure 3.18 where the first few boxplots are for the pixels near the corner and they overlap. There are no unusual boxplots for either of the base and augmented ensemble predictions so there are no extreme errors. Apart from the corners, the base models perform better than

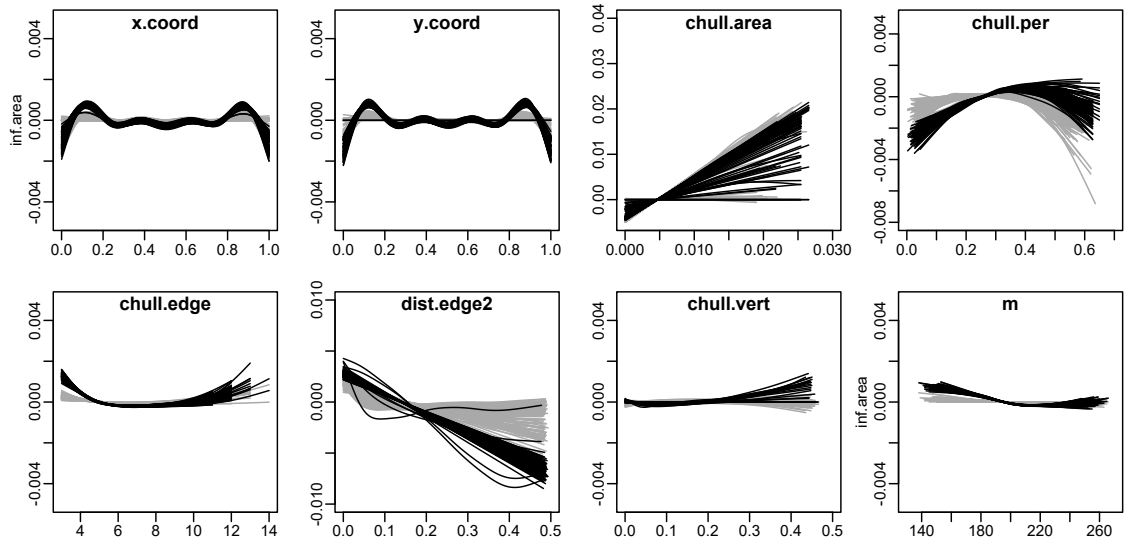


Figure 3.15: Estimated smooth components of the GAMs in the individual base and augmented models. Black lines are the estimated smooth components for the augmented models that are overlaid for 100 models, and the estimated components for the base models are shown in the background in gray lines.

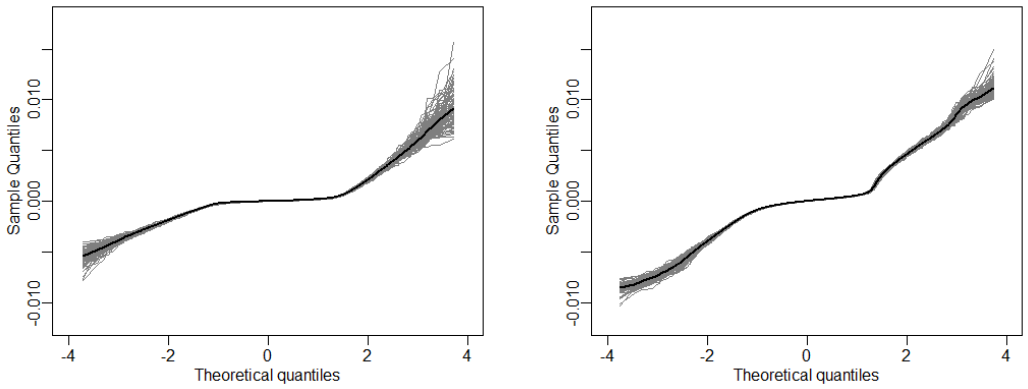


Figure 3.16: The normal quantile-quantile plot of residuals versus fitted values for individual base (left) and augmented models (right).

the augmented models. The spatial patterns of the errors are shown in Figure 3.19 and 3.20. The unusual appearance of the augmented models also exists here.

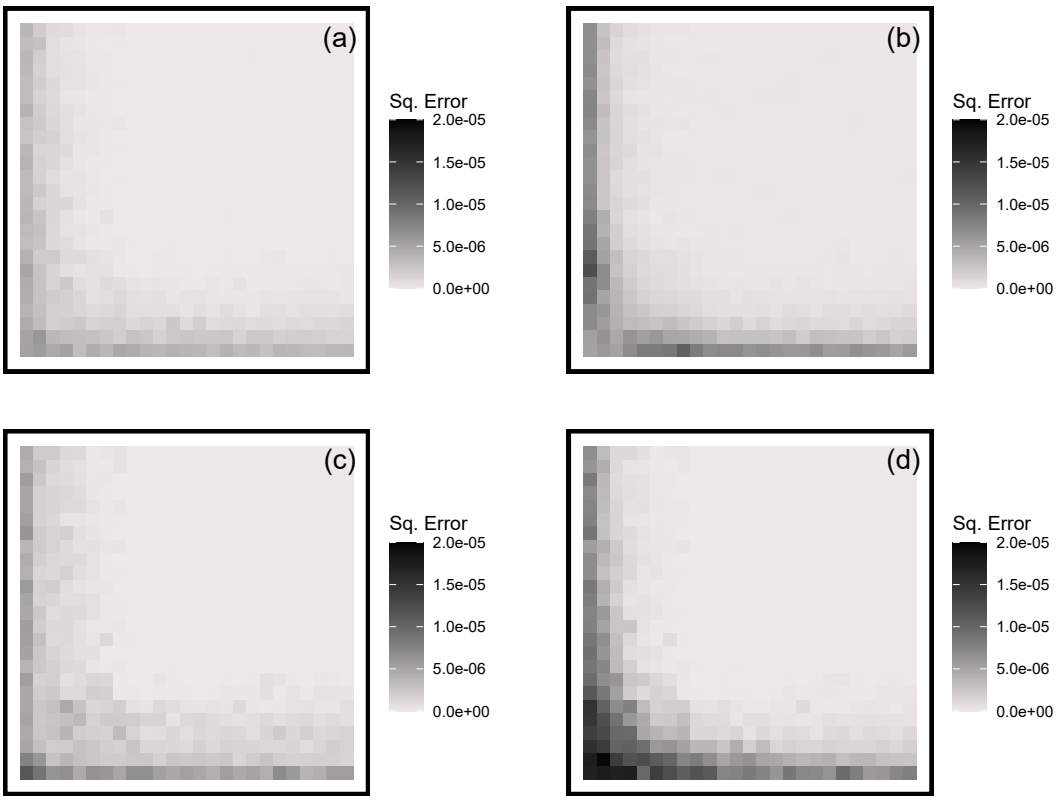


Figure 3.17: The squared error of infinite plane area and predicted area averaged over pixel bins for (a): base models, (b): augmented models, (c) unit square area, and (d) convex hull area.

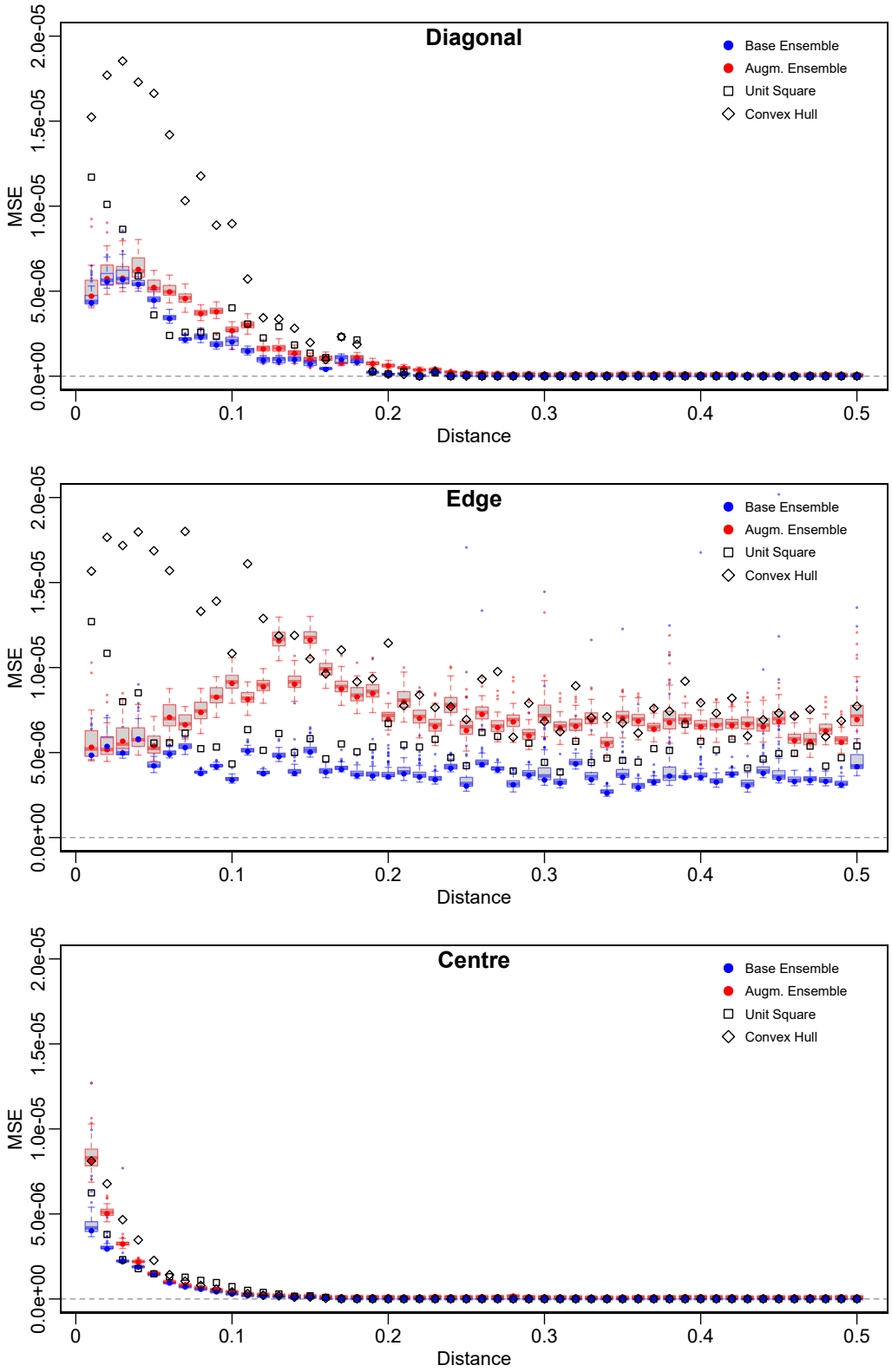


Figure 3.18: The boxplots of MSE for predictions from individual base models (blue) and augmented models (red) over pixel bins along different transects. The ensemble predictions are shown with the solid points in the same colour at each transect bin. The MSE for unit square and convex hull area are shown with square and diamond shaped points.

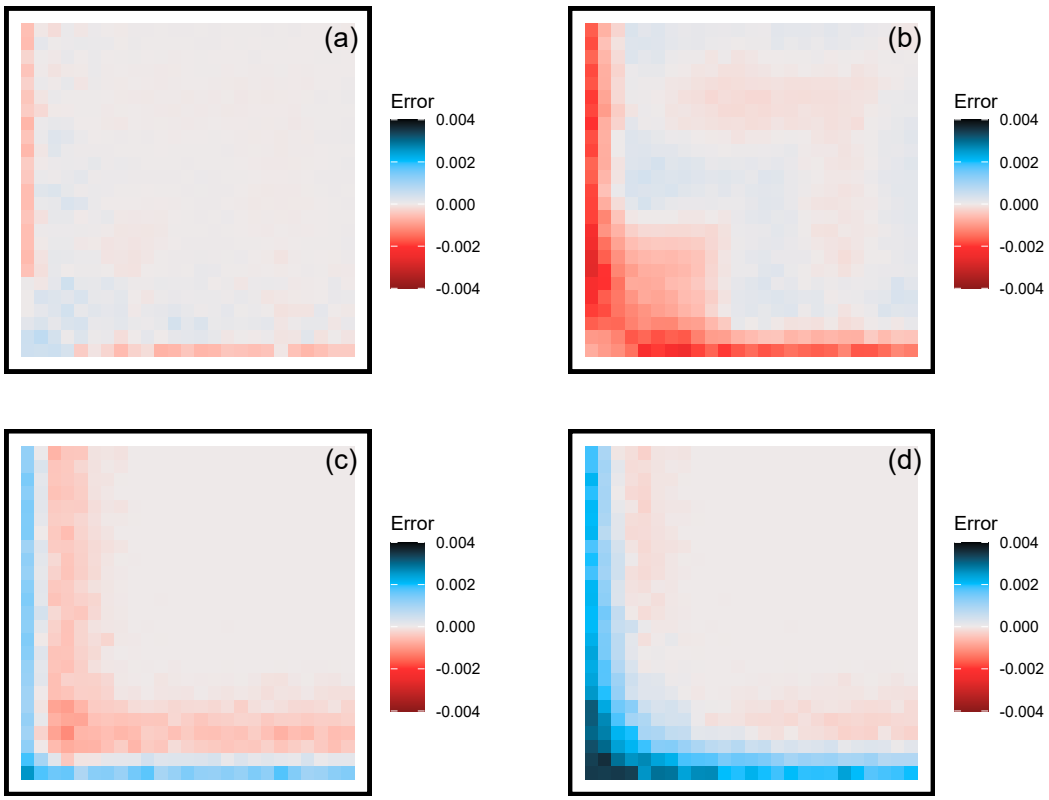


Figure 3.19: The error of infinite plane area and predicted area averaged over pixel bins for (a): base models, (b): augmented models, (c) unit square area, and (d) convex hull area.

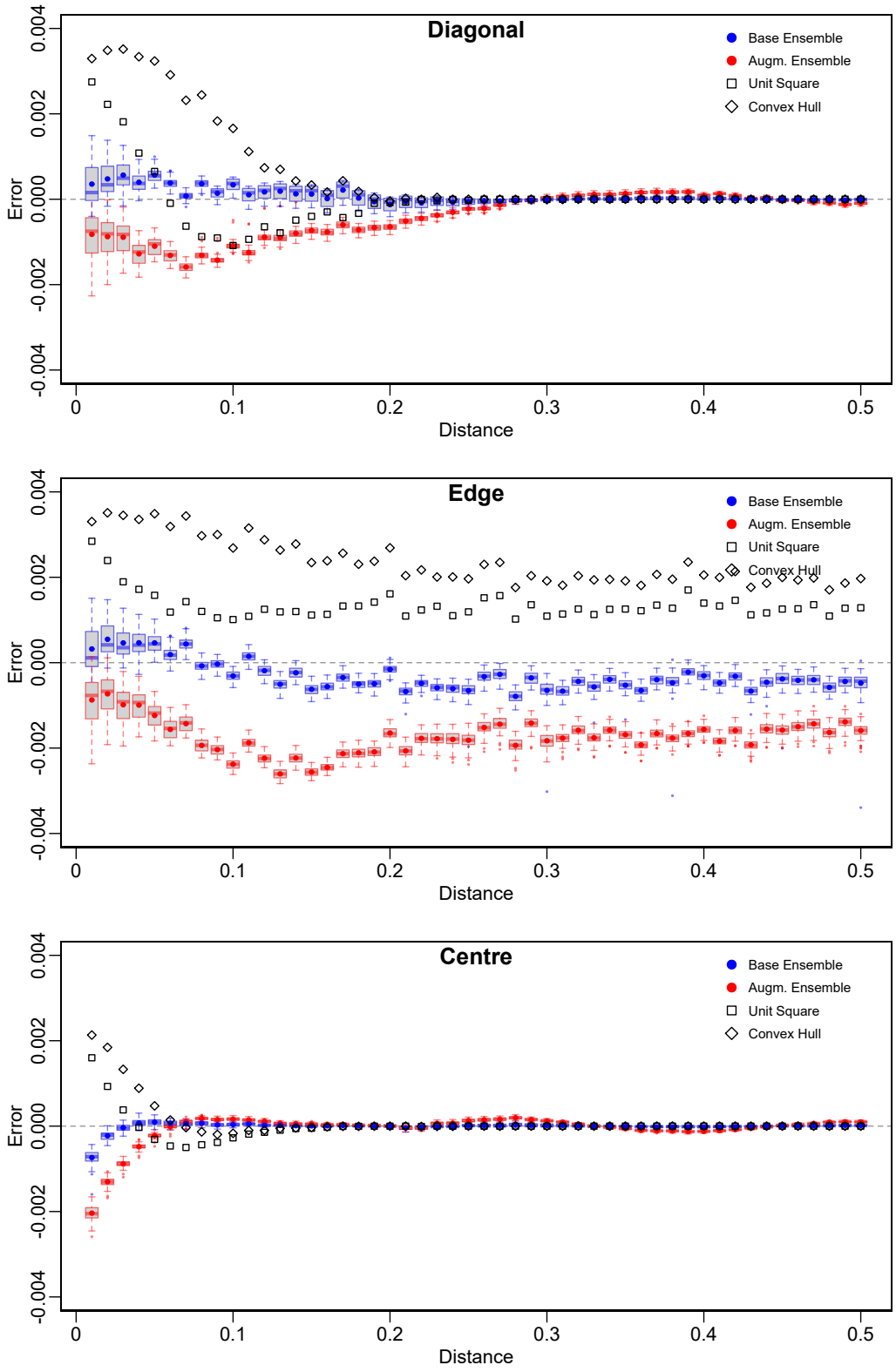


Figure 3.20: The boxplots of mean error for predictions from individual base models (blue) and augmented models (red) over pixel bins along different transects. The ensemble predictions are shown with the solid points in the same colour at each transect bin. The MSE for unit square and convex hull area are shown with square and diamond shaped points.

3.6 Classification of boundary-affected points

Area prediction for Voronoi cells discussed in Section 3.4 is useful for the area prediction of both the interior and edge cells. The area prediction results show that the predicted area differs from what is observed especially near boundaries. This gives an idea that the clipped cell area does not reflect the true cell area since the cells are affected by the boundaries. On the other hand, the interior points that are closer to the centre of the region are less likely to be affected by the boundaries. The boundary-affected points constitute approximately 28% of a test data with size $n_{test} = 3 \times 10^5$ which we will use for the classification of the *boundary-affected points*. This section aims to give an idea on the classification of cells that are likely to be effected by the boundaries and performs a simple logistic regression.

The same training sets from Section 3.4 are used to create individual models for the prediction of the probabilities of being boundary-affected. The model in equation (3.7) is modified to a generalized linear model with Binomial family to fit logistic regression models. The binary response variable Y is denoted as $p = P(Y = 1|\theta)$ that indicates the probability of a point to be boundary-affected given $\theta = \{x_1, x_2, \ldots, z_3, z_4, \ldots, z_9\}$ and the relationship between the predictors and log-odds is written as

$$\log \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \tag{3.12}$$

and the probabilities are recovered as

$$p = \frac{e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots}}{1 + e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots}}. \tag{3.13}$$

Let $\hat{\mathbf{p}}$ be the prediction matrix of probabilities calculated using individual logistic regression models that are fitted to the previously generated training subsets. Hence the $\hat{\mathbf{p}}$ is created using the individual logistic regression models in the test set as

$$\hat{\mathbf{p}} = \begin{bmatrix} \hat{p}_{11} & \hat{p}_{12} & \hat{p}_{13} & \cdots & \hat{p}_{1t} \\ \hat{p}_{21} & \hat{p}_{22} & \hat{p}_{23} & \cdots & \hat{p}_{2t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{p}_{n_{ts}1} & \hat{p}_{n_{ts}2} & \hat{p}_{n_{ts}3} & \cdots & \hat{p}_{n_{ts}t} \end{bmatrix} \quad n_{test} = 1, \ldots, \ 3 \times 10^5 \text{ and } t = 1, \ldots, 100.$$

Then the ensemble predictions of the probabilities in the test set are calculated as

$$\tilde{p} = \left(\frac{1}{100} \sum_{j=1}^{t} \hat{p}_{1t} \quad \frac{1}{100} \sum_{j=1}^{t} \hat{p}_{2t} \quad \cdots \quad \frac{1}{100} \sum_{j=1}^{t} \hat{p}_{n_{ts}t} \right)^{\top}.$$

The prediction results in the test set are illustrated using the receiver operating characteristic (ROC) curves that present a useful evaluation of the performance of a binary classifier depending on the measures of sensitivity and specificity at different thresholds. The ROC curve is calculated based on the values on the confusion

		Actual	
		Boundary-affected	Unaffected
Pred.	Boundary-affected	True Positives	False Positives
	Unaffected	False Negatives	True Negatives

Table 3.7: The confusion matrix table.

matrices as in Table 3.7. We calculate the sensitivity and specificity to construct the ROC curve for all possible thresholds.

$$\text{True Positive Rate} = \text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{False Positive Rate} = (1 - \text{Specificity}) = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

The ROC curve in Figure 3.21 summarizes all confusion matrices from all possible threshold values. The varying threshold values are used to calculate the true positive and false positive rates for each threshold, and the calculated rates construct the curve in Figure 3.21. The curve close to the top-left corner indicates the good performance of the model to classify the boundary effected cells. The threshold that gives the desired rates can be selected as the optimal threshold. A random classifier that has equal true positive and false positive rates gives the diagonal line.

3.7 Alternative data scenarios

We considered two types of modeling strategies; in the case of known and unknown boundaries. These two types of conditions can cope with many important data structures but cannot cover all possible data scenarios. It is not practical to go over all boundary scenarios, however, the methodology we use can be adopted for alternative scenarios possibly with appropriate modifications.

The are two main stages in our study; first, a large data set is created through the simulation where the settings of the simulation such as which cell properties to calculate are carefully chosen for a specific point pattern type. Second, the

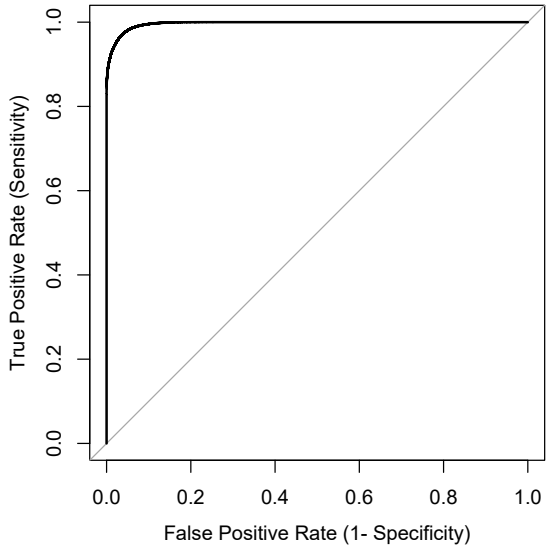


Figure 3.21: ROC curve created from the confusion matrices for the test data.

methodology to predict the cell area or classify the boundary-affected cells should be considered where we gave the major emphasis on the prediction of cell area. The models we created can be used for any type of point pattern and boundary cases, but we have not checked the accuracy of the predictions in this chapter. Therefore, we will investigate and explain the usage of the models for general data scenarios in Chapter 4.

We considered the set of n recorded points $X \in \Omega$ where the sampling region is a unit square $\Omega = [0, 1]^2$ and X follow a homogeneous Poisson process so $n \sim Po(\rho|\Omega|)$ where $|\Omega|$ is the area of the region. The simulation study is designed to generate a data set to learn more about the properties of homogeneous Poisson points. Then the parts of this data set is used as training, and validation sets to fit and evaluate models.

The `R` code in Appendix B calculates the statistical properties of homogeneous Poisson points with intensity $\rho = 200$ for a single realization. Given a set of homogeneous Poisson points within a finite region Ω, it performs the Voronoi tessellation of points, randomly select a point and calculate the cell properties which are listed in Table 3.1. We repeated the process given in the code for one million independent realizations, and by randomly sampling a point at each realization and recording the cell properties, the entire data set is created. If the similar path is followed to repeat the simulation for a different data or boundary scenario, the data obtained from the simulation can be used for further purposes such as fitting models

to predict a cell property or a classification case.

3.8 Conclusions

This chapter discussed a thorough analysis of constrained Voronoi tessellation cell area due to imposed boundaries and provides ways to deal with the issues caused by boundaries. Since the data points in a finite region lack neighbour points beyond the boundaries, the boundaries determine the characteristics of cells that lie on the boundary or close to the boundary. However, the base and augmented models we created treats the Voronoi cells as they are in a larger region or in an infinite plane.

The Voronoi tessellation has a wide use in spatial data analysis and we demonstrated how its statistical properties change near the boundaries. There are ways to reduce the problems that boundaries cause by fitting regression models that predict the cell area. The base and augmented ensemble models are the two approaches we proposed. For the general use, base models perform satisfactorily well. Augmented models on the other hand are able to improve some of the weaknesses of the base models such as reducing the extreme errors, but not have a sufficient global performance. Therefore, one could decide whether to achieve a good global performance, or to minimize the largest errors. Based on accurate area prediction, we suggest the use of base models due to its global performance.

One of the circumstances of this chapter is the consideration of particular boundary types and point patterns. Since it is not possible to cover all possible scenarios, we considered the most important cases that may have a general use. However, as in the previous section, we explained how one can perform the simulation by adopting the code we provided which can be modified easily for different boundary types and point patterns.

Chapter 4

Robustness of area prediction

This chapter extends the study on the area prediction of Voronoi cells described in Chapter 3 by considering homogeneous Poisson points with varying intensity cases, and the situations where the spatial data shows regular and clustered patterns. The main objective of this chapter is to develop generalized versions of the previous models so they can be applied to a wide range of data cases that have different spatial characteristics. More importantly, our proposed approach aims to allow the models to be applicable to real data sets as well.

First, we start by testing the models from Chapter 3 which were created using training sets of Voronoi cells from homogeneous Poisson points with a specific intensity. Section 4.1 explores the beahviour of these models on the test sets that contain Voronoi cells of homogeneous Poisson points from different intensities. In Section 4.2, we describe regular and clustered points based on an existing method, and focus on the local intensities (i.e. the highly clustered parts) that may be also different than the global intensity of the points. Then we propose a way that the models created in Chapter 3 can be updated based on the local intensities to improve the prediction performance. Finally, we use the updated versions of the models for area prediction for regular and clustered points in a simulation study and real data sets in Section 4.3, and an overall summary is given in Section 4.4.

4.1 Misspecification of intensity

Chapter 3 considered models that predict the true area of Voronoi cells given the other properties of the cells. These models are fitted using a training data that contains properties of Voronoi cells from homogeneous Poisson points with point

intensity $\rho_0 = 200$. However, it is uncertain how the models perform for data sets with varying point intensities. more specifically, we have not tested the model for data sets that has $\rho = 50$ or $\rho = 500$. In this chapter, we consider the cases of varying intensities which we call as *misspecified intensity* which is an often case in the real data. Therefore, the aim is to check the robustness of the models that are fitted to the training data with intensity ρ_0 on test data sets that have number of points $n \sim Po(\rho)$ where $\rho \neq \rho_0$. The choices of different ρ values will be discussed later.

To investigate the robustness of the models for misspecified intensities, the simulation study is extended to generate new training and test sets. A simulation is performed which we generate data sets of homogeneous Poisson points with intensities $\rho \in \{50, 100, 200, \ldots, 600\}$. First, we consider each data separately and fit additive models for each case. Additionally, these models are used for area prediction for unrelated intensities. For instance, we fitted separate models for each of $\rho \in \{50, 100, 200, \ldots, 600\}$ but we used the model fitted for $\rho = 50$ data on $\rho = 600$ data to see how the model with misspecified intensity performs. The purpose of this approach is to detect if any issues occur with misspecified intensity, and propose ways to improve the predictions in such situations. Otherwise, relying on a single model for any data set may not be accurate.

To compare all cases, let ρ and $\rho^\star$ indicate the intensity of the training and test data respectively. Results are summarized in Table 4.1 in terms of the mean squared errors. The predicted and true area are standardized as $\hat{\mathcal{A}}_i = \rho_t \hat{A}_{i,\rho_t^\star}$ and $\mathcal{A}_i^\star = \rho_t A_{i,\rho_t^\star}$ hence $E(\hat{\mathcal{A}}_i) = E(\mathcal{A}_i^\star) = 1$ for cell areas associated with points x_i, $i = 1, \ldots, n$ for all intensity cases $t = 1, 2, \ldots, 7$, and the values in the table are MSE $\times$ 10^2. The mean squared error is calculated as $\text{MSE}_i = \sum_i^n (\hat{\mathcal{A}}_i - \mathcal{A}_i^\star)^2/n$. The rows in the table indicate the MSE for a specific data intensity ρ_t when different models that has $\rho_t^\star$ are used. The top panel shows the results for base models and the bottom panel is for augmented models. The blue colour shows the smallest MSE achieved for each data. The most important conclusion from the table is that the smallest MSE is always achieved using the models which are fitted to data which have the same intensity as the data being analysed. Usage of models from different intensities, that is when $\rho_t \neq \rho_t^\star$, gives a larger MSE. Hence, it is not appropriate to use a model when there is a mismatch between the training and test set intensities (ρ_t and $\rho_t^\star$). There is a better performance of the base models as concluded in Chapter 3.

Data intensity ρ	**Model intensity**, $\rho^\star$						
	50	100	200	300	400	500	600
50	4.48	7.81	8.45	9.23	11.95	12.46	17.70
100	4.66	3.09	4.79	5.09	5.73	6.56	7.29
200	3.22	3.01	2.22	3.01	3.11	3.37	3.41
300	2.65	2.40	2.25	1.81	2.24	2.31	2.44
400	2.27	2.06	1.89	1.85	1.51	1.86	1.87
500	1.99	1.80	1.65	1.60	1.56	1.31	1.57
600	1.83	1.65	1.49	1.43	1.40	1.39	1.19
50	6.21	10.94	10.43	10.93	11.09	14.45	12.07
100	5.98	5.05	5.90	6.39	6.36	6.61	6.69
200	4.87	5.14	3.70	4.34	3.96	3.92	3.84
300	4.17	4.69	3.54	3.02	3.01	2.95	2.81
400	3.75	4.14	3.16	2.91	2.25	2.50	2.31
500	3.42	3.93	2.99	2.77	2.36	2.09	2.10
600	3.07	3.62	2.76	2.57	2.19	2.07	1.75

Table 4.1: Mean squared errors of area prediction for base (top panel) and augmented models (bottom panel). The rows are for the data set intensities $\rho \in \{50, 100, \ldots, 600\}$ and columns are for the models that is fitted for each $\rho^\star \in \{50, 100, \ldots, 600\}$. The case when $\rho_t = \rho_t^\star$ for $t = 1, \ldots, 7$ indicates the usage of the same model fitted for the data that has intensity ρ_t, and $\rho_t \neq \rho_t^\star$ indicates the usage of models fitted from different data intensities.

In this experiment, another consideration might be to evaluate the joint performance of models from other models that has the lower and higher intensities. For instance, the area prediction for $\rho_t = \rho_t^\star$ is $\hat{A}_{i,\rho_t^\star}$ for a specific intensity. As an alternative approach, we take the weighted average $(\hat{A}_{i,\rho_{t-1}^\star} + \hat{A}_{i,\rho_{t+1}^\star})/2$ to check how robust the prediction of $\hat{A}_{i,\rho_t^\star}$ is to misspecified intensity. The MSE of the weighted average of predictions from models $\rho_{t-1}^\star$ and $\rho_{t+1}^\star$ are given in Table 4.2. The results show that the weighted average is not very accurate. In particular, for augmented models, a lower MSE is obtained by just using a prediction model based on the higher intensity $\rho_{t+1}^\star$. However, for base models, this is not true and averaging the two predictions gives marginally lower MSE.

One possible violation of the modelling assumptions is the misspecification of the intensity which can lead to problems in the area prediction. Some of the cell properties such as the cell area and perimeter depend on ρ. When the model is fitted using a training set with ρ_0, and the test set has intensity $\rho \neq \rho_0$, the mean cell area and perimeter for the training and test data will be different. Therefore, the issue can be approached by modifying the variables (that depend on ρ) in the test

Data intensity ρ	**Model intensity**, $\rho^\star$						
	50	100	200	300	400	500	600
50	–	5.46	8.26	10.02	22.16	17.48	–
100	–	4.60	3.62	5.15	6.48	7.94	–
200	–	2.48	2.94	2.46	3.14	3.23	–
300	–	2.38	1.95	2.22	1.91	2.33	–
400	–	2.03	1.92	1.60	1.84	1.60	–
500	–	1.78	1.67	1.59	1.38	1.56	–
600	–	1.63	1.51	1.43	1.40	1.24	–
50	–	6.28	8.99	10.58	12.09	11.12	–
100	–	5.60	4.79	5.97	6.23	6.43	–
200	–	3.83	4.50	3.38	4.02	3.84	–
300	–	3.73	3.50	3.17	2.76	2.86	–
400	–	3.36	3.34	2.53	2.65	2.15	–
500	–	3.13	3.19	2.61	2.29	2.19	–
600	–	2.84	2.95	2.42	2.27	1.87	–

Table 4.2: Mean squared errors calculated from the weighted average of the predictions from models with $\rho^\star_{t-1}$ and $\rho^\star_{t+1}$ when the particular interest is the intensity ρ_t. Results are given for base (top panel) and augmented models (bottom panel) respectively. The smallest MSE in each row is highlighted in blue colour. The (–) symbol indicates no weighted average is calculated since there are no more columns on the left or right.

data appropriately with respect to ρ_0. The misspecified intensity can happen for the intensity of the homogeneous Poisson points as we considered in this section, or it is possible to obtain variable local intensities for different sub regions of clustered point patterns. The latter case will be considered in the next section by taking into account various regular and clustered point patterns.

4.2 Regular and clustered point patterns

The foundation sources to study the point patterns include Cox & Isham (1980); Cressie (2015); Cressie & Wikle (2015); Diggle (1983); Gelfand *et al.* (2010); Illian *et al.* (2008); Ripley (1988, 2005) and Baddeley *et al.* (2015). Various ways to generate different point patterns are explained and discussed in these references. We are interested in using a model to generate realizations of regular and clustered point patterns. The saturation process by Geyer (1999) explained in Section 1.5 has practical features that the different types of point patterns can be both analysed and simulated by the implementation of the method in a `R`.

The saturation process can be performed in `R` using the routines in the `spatsat` package as explained in Baddeley *et al.* (2015). There are two ways of choosing the parameters of the process, *i*) the values of the parameters β, γ, r, s can be predefined, or *ii*) the process can be fitted to a point pattern data, hence the parameters are estimated, then simulated realizations of the fitted point process are created using Metropolis-Hastings algorithm. There are also other well known processes such as the Poisson cluster process, Neyman-Scott, and Bartlett-Lewis cluster processes, or jittering the grid points, but the scheme we describe is more convenient in terms of the flexibility and practicality of the control of the parameters.

By the modification of the parameter γ it is possible to generate realizations of regular and clustered point patterns. The different values of γ decides the magnitude of regularity and clustering. For instance, the homogeneous point pattern case is obtained by setting $\gamma = 1$. However, setting of $\gamma < 1$ results more regular patterns and $\gamma > 1$ clustered points where the more regular and clustered points are achieved by departure from $\gamma = 1$ increases.

An example of different point patterns generated from the saturation process is shown in Figure 4.1. From top-left to bottom-right, n points are generated for the values $\gamma = 0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 2, 3$, where $n \sim Po(200)$. The centre plot where $\gamma = 1$ indicates homogeneity which can be considered as the baseline and realizations of regular and clustered points are shown for different values of $\gamma < 1$ and $\gamma > 1$.

4.3 The prediction of Voronoi cell area based on regular and clustered points

Area prediction for regular and clustered points is not straightforward as in the homogeneous Poisson point pattern case, but it is possible to use the models fitted for homogeneous data with modified covariates of regular and clustered points as we briefly mentioned. In this section, we explain the process of area prediction for regular and clustered points that aimed to be done by using the local intensities. Let the local intensities at the locations of the regular or clustered points $\{x_i\}_{i=1}^{n}$ be ρ_i which can be estimated as $\hat{\rho}_i$.

One method to estimate the local intensity is to use the kernel smoothed intensity from the point pattern. Given a point pattern data, the method computes a fixed-bandwidth kernel estimate of the intensity function of the related point process

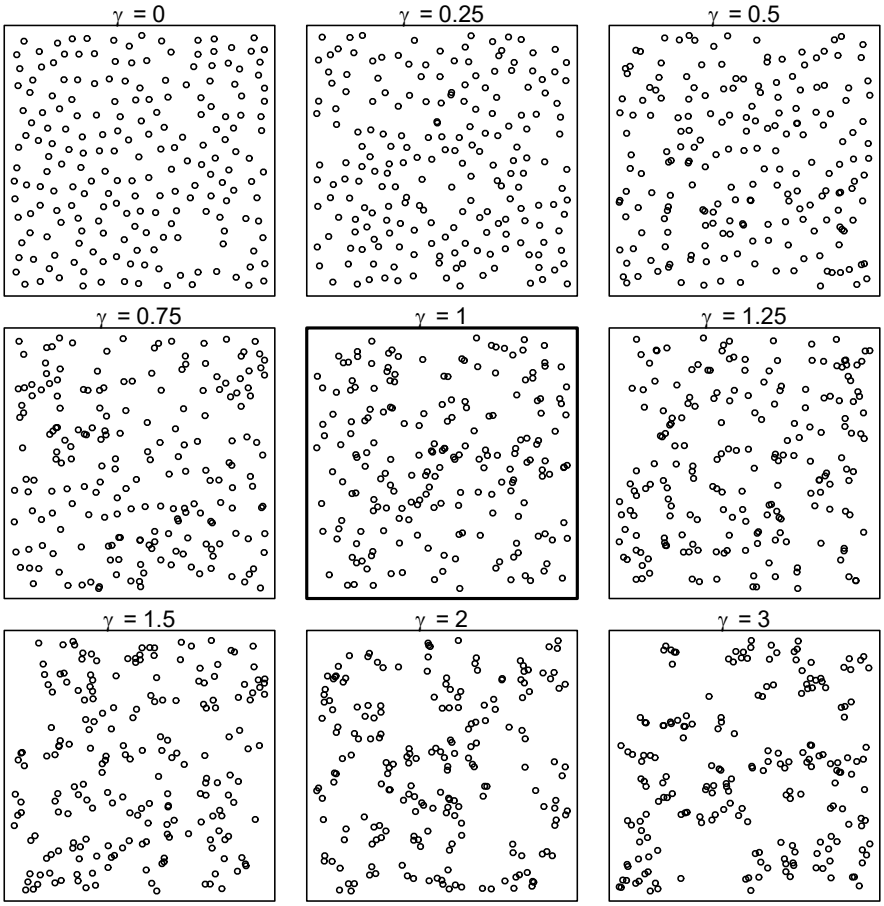


Figure 4.1: Simulated points from Geyer's saturation process (Geyer, 1999). Examples of inhibition or repulsion to homogeneity and to clustering or attraction are shown from top-left to bottom right with incremental magnitudes of repulsion and attraction where the plot with bold frame at the centre is the homogeneous case. The intensity parameter $\beta = 200$, interaction radius $r = 0.05$, and the saturation threshold $s = 2$ are fixed for all point patterns, and the interaction parameter γ takes values 0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 2, 3 where $\gamma < 1$ indicates regularity, and clustering if $\gamma > 1$.

using an isotropic Gaussian kernel as the default option (Diggle, 1985). The edge corrected intensity estimate at an arbitrary location u is

$$\hat{\rho}(u) = e(u) \sum_i \kappa(x_i - u)\omega_i \tag{4.1}$$

where κ is the kernel function based on isotropic Gaussian distribution, ω_i are the weights if assigned to the points, and $e(u)$ is the correction term for bias at the edges defined as

$$\frac{1}{e(u)} = \int_W \kappa(v - u)dv \tag{4.2}$$

where W is the observation window. The edge corrected estimate of the local intensity is obtained through dividing the convolution of the Gaussian kernel by the edge correction term $e(u)$.

An example of the local estimate of the intensity over the region with edge correction is shown in Figure 4.2 using the method shown in (4.1) for the point patterns from Figure 4.1. The `density.ppp` functions in the `spatstat` package is used to compute the kernel smoothed intensity given the point patterns.

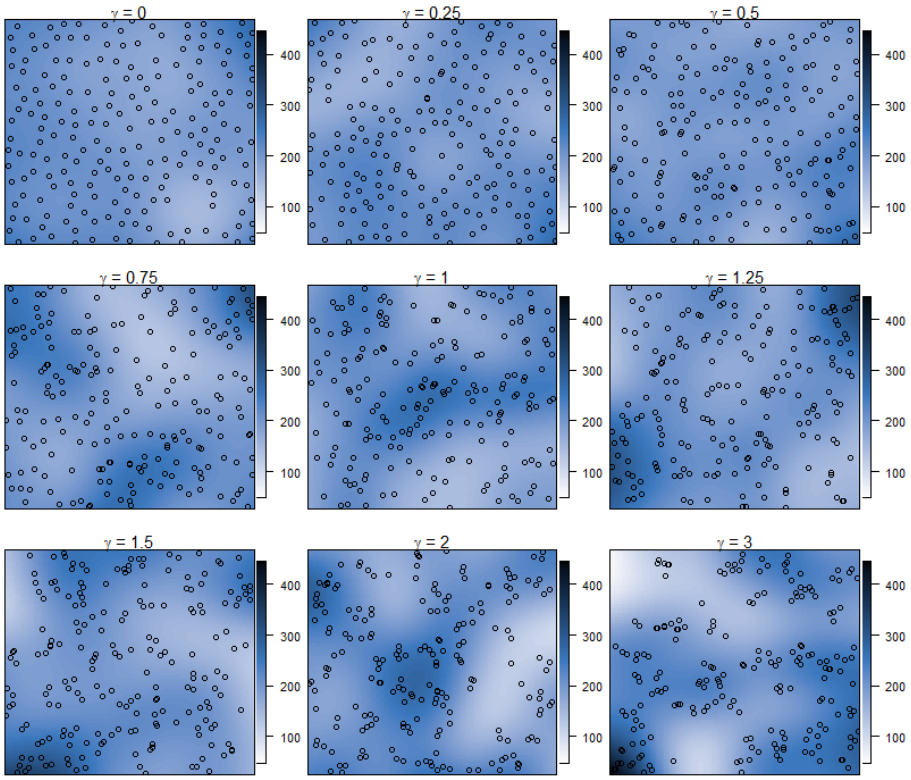


Figure 4.2: Kernel smoothed intensity of the point patterns.

The local intensities can be estimated at the data points rather than the entire region which is particularly useful for area prediction. Given a set of points $X^{\gamma} = \{x_i \in [0,1]^2;\ i = 1, \ldots, n\}$ generated based on the value of γ, and $n \sim Po(\rho_0 = 200)$, we have a local estimate of the intensity $\hat{\rho}_i$ at each point x_i. The area prediction for points is then performed as

$$\hat{A}_i = \sum_{j=1}^{p} f_{\rho_0, j}(\theta_{ij}) \tag{4.3}$$

where f_j are the unknown smooth functions fitted in the additive model for area prediction, p is the number of predictors, θ_j are the covariates $j = 1, 2, \ldots, p$ such

as the raw cell area, perimeter, number of edges due to induced boundaries, cell type and so on for p covariates in total. The model in (4.3) does not depend on the local intensity $\hat{\rho}_i$ and may have the issues highlighted in Section 4.1, however the model can be improved as

$$\hat{A}_i^\star = \sum_{j=1}^{p} f_{\hat{\rho}_i,j}(\theta_{ij}^\star). \tag{4.4}$$

The model in (4.4) uses the feature of the $\hat{\rho}_i$ to scale some of the covariates $\theta_j^\star$ that depend on the $\hat{\rho}_i$. For instance, the scaled cell area due to boundaries is $A_i^\star = A_i\hat{\rho}_i/\rho_0$, and the scaled perimeter is $P_i^\star = P_i\sqrt{\hat{\rho}_i/\rho_0}$. These scaling factors are defined based on the theoretical derivation of the expected cell area and perimeter from equation (2.1). Other covariates such as the number of cell edges do not depend on the data intensity ρ_0. Although we only considered the scaling of the cell area and the perimeter based on different global or local point intensities, other cell properties such as the closest distance between the point and the boundary can also be investigated to see whether it depends on the intensity.

The true area A_i' is determined by generating the point pattern in a larger region $\Omega^\star = [-1, 2]^2$, and studying the points inside the region $\Omega = [0, 1]^2$. The shifting approach in Chapter 2 is not used because the periodic boundary conditions are not very appropriate especially for clustered points. Consider clustered points at the corner of the region where the cluster is clipped by the boundary. We only observe some of the points that belong to the cluster. If shifting is applied, then the new neighbour points of the cluster may be far from the cluster. Therefore, the cell area A_i' obtained by shifting for a sampled point in the cluster may be very large. Instead, we use an approach that simulates points in a larger region, $\Omega^\star = [-1, 2]^2$ but samples from the points in $\Omega = [0, 1]^2$. The true cell area is calculated based on the Voronoi tesselation of all points in $\Omega^\star = [-1, 2]^2$. Therefore, irregularities such as clusters are not clipped, and the true area is calculated based on a continuum of the cluster rather than shifting the clipped cluster to the centre.

4.3.1 Results for simulated data

Area prediction is performed for the simulated data. In each simulation, sets of n points are generated for a specific value of γ, and the statistical properties of a randomly selected point is calculated. This is done for 10^4 realizations for each value of γ. The data is used to predict the cell area of the randomly selected points using models (4.3) and (4.4) and the results are summarized in the Table 4.3.

The mean squared error values are given in Table 4.3 where the values are for MSE$\times 10^6$, and their standard errors are in Table 4.4 for (SE$\times 10^8$). B and Ag notations indicate the area prediction results using model (4.3) for the *base* and *augmented* models. On the other hand, $B^\star$ and $Ag^\star$ are the base and augmented model prediction results using the model (4.4). The base and augmented model results are separated as the two main row panels (top and bottom) in Table 4.3. Each panel is also separated into three sub-panels which are for the *global*, *interior* and *edge* regions respectively. The interior region is defined as $\Omega_{in} = [0.15, 0.85]^2$ and edge region is for the points that are located within $\Omega_{ed} = \Omega'_{in}$ where $\Omega = \Omega_{in} \cup \Omega_{ed}$.

The MSE results in Table 4.3 show an overall better performance of model (4.4) in all irregular data situations both for base and augmented models. The MSE and the standard deviation is extremely small for interior points, which is expected, but differences are more apparent for the edge points. In terms of the values of γ, the MSE is smallest when $\gamma = 0$ that is the most regular point pattern and highest for the highly clustered points when $\gamma = 3$. The recommended model based on this experiment is to use $B^\star$ (base model that use the local estimate of intensity).

To check whether the differences between MSE values from models (4.3) and (4.4) are significant, consider the MSE values with confidence intervals in Figure 4.3. The black and red colours represent models (4.3) and (4.4) respectively for base models, and pale colours for the augmented models. Results are separated for global, interior and edge regions from left to right respectively. In the global case, $B^\star$ (•) always have the smallest MSE. We see the same pattern for edge region which is the case that is being of interest. Although all models give very small MSE values for the interior region, $B^\star$ (•) is the smallest except when $\gamma = 3$. Also, the confidence intervals between the base and augmented models generally do not overlap which suggests the base models are significantly better than the augmented models in area prediction. It is appropriate to keep relying on the model in (4.4) that uses the estimated local intensities. However, we do not suggest a strict usage of the base models since the augmented models reduce the maximum error, and the base models give the smallest global MSE. The preference between two models should be based on which of these criteria is more important in a particular application.

4.3.2 Results for real data

In this section, the area prediction method is applied to several real data sets, all are available in the `spatstat` library in `R`. We selected four data sets `finpines`,

		γ								
Cases		0	0.25	0.50	0.75	1	1.25	1.50	2	3
Global	B	0.186	0.226	0.298	0.411	0.529	0.628	0.807	0.907	1.254
	$B^\star$	0.163	0.196	0.243	0.334	0.429	0.520	0.631	0.758	1.173
	Ag	0.698	0.691	0.690	0.800	0.863	0.960	1.092	1.170	1.478
	$Ag^\star$	0.639	0.629	0.614	0.717	0.749	0.840	0.934	1.059	1.477
Interior	B	0.001	0.001	0.001	0.002	0.004	0.004	0.011	0.015	0.023
	$B^\star$	0.000	0.000	0.001	0.001	0.003	0.002	0.006	0.013	0.038
	Ag	0.019	0.021	0.026	0.035	0.043	0.048	0.066	0.073	0.095
	$Ag^\star$	0.020	0.017	0.015	0.015	0.017	0.019	0.030	0.050	0.112
Edge	B	0.365	0.442	0.592	0.805	1.021	1.222	1.566	1.742	2.487
	$B^\star$	0.321	0.385	0.483	0.656	0.830	1.016	1.227	1.458	2.314
	Ag	1.353	1.337	1.348	1.536	1.633	1.831	2.070	2.198	2.865
	$Ag^\star$	1.239	1.219	1.210	1.393	1.437	1.625	1.798	2.006	2.849

Table 4.3: Mean squared error of the predicted area using base B and augmented Ag models. B indicates the base model, whereas $B^\star$ is a base model that uses the scaled covariates based on the estimated local intensities $\hat{\rho}_i$ at the data points. The same case applies for the Ag and $Ag^\star$. Results are given in three row panels that are for global, interior and edge parts respectively. Columns show the MSE results for each point pattern type based on the value of γ. In each column, results are obtained from 10^4 data points each of which is sampled from 10^4 realizations of independent data sets. Results are for MSE $\times 10^6$.

		γ								
Cases		0	0.25	0.50	0.75	1	1.25	1.50	2	3
Global	B	0.670	0.850	1.332	2.167	2.737	2.947	4.182	4.759	7.166
	$B^\star$	0.532	0.674	0.903	1.235	1.653	2.098	2.741	3.595	5.940
	Ag	2.065	2.195	2.297	2.781	3.132	3.400	4.111	4.498	6.828
	$Ag^\star$	1.957	2.000	2.036	2.401	2.560	2.873	3.253	3.922	5.897
Interior	B	0.001	0.002	0.033	0.033	0.144	0.046	0.200	0.218	0.529
	$B^\star$	0.001	0.001	0.031	0.007	0.127	0.024	0.156	0.209	0.581
	Ag	0.032	0.038	0.062	0.089	0.188	0.131	0.728	0.296	1.022
	$Ag^\star$	0.018	0.018	0.038	0.036	0.126	0.054	0.528	0.562	0.939
Edge	B	1.267	1.612	2.587	4.179	5.213	5.637	8.024	9.064	14.126
	$B^\star$	0.998	1.270	1.735	2.340	3.105	3.983	5.222	6.827	11.671
	Ag	3.909	4.190	4.401	5.355	6.060	6.563	7.944	8.741	13.335
	$Ag^\star$	3.661	3.748	3.880	4.520	4.773	5.396	6.100	7.347	11.459

Table 4.4: Standard error of the MSE values from Table 4.3. Results are for SE$\times 10^8$.

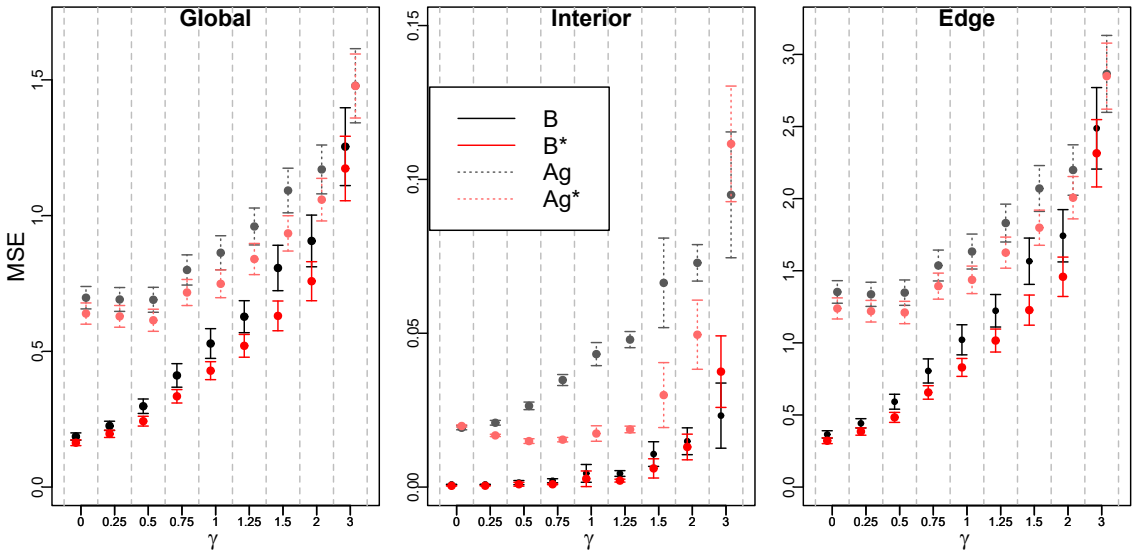


Figure 4.3: Confidence intervals for MSE values. The points (of all colours) show the MSE and the lines (of all colours) show the confidence intervals based on different methods that are shown in the centre plot legend.

`longleaf`, `spruces`, and `waka` that have different spatial features. These data sets are created from different types of trees within specific sampling regions and are examples of point pattern data. The locations of trees are marked by the height and the diameter of trees which make it a marked point pattern. However, we will only use the tree locations as points in this chapter.

We additionally use a data set that contain the chemicals measured in the soil in Barro Colorado Island (BCI) at sampled locations. The BCI data has two-thirds of the locations from the equidistant grid points and one-thirds are sampled at a random isotropic direction with some distance from the regular points. Therefore, this data set has a completely different nature compared to the tree data sets mentioned earlier. The BCI data is a geo-referenced data that has coordinate-based sampled points where chemical levels are measured. The BCI data, is collected by a part of the *Effects of soil-borne resources on the structure and dynamics of lowland tropical forests* project by principal investigators: Jim Dalling, Robert John, Kyle Harms, Robert Stallard and Joe Yavitt. The data that are publicly available in `http://ctfs.si.edu/webatlas/datasets/bci/soilmaps/BCIsoil.html` only contains sampling locations and kriging estimates of the soil data. However, the raw soil data was obtained from Dalling *et al.* (2021) by personal communication.

Descriptive information about the data sets is given in Table 4.5 and the locations are shown in Figure 4.4. In Table 4.5, the estimated parameter $\hat{\gamma}$ is also given for

each data set; these fall in the range $0 \leq \gamma \leq 3$ which we used in the simulations. The $\hat{\gamma}$ values indicate that `spruces` and BCI are the data sets that have regular pattern, `waka` is almost completely homogeneous data, and `finpines` and `longleaf` have clustering. Hence the presentation order of the data sets are decided based on the $\hat{\gamma}$.

Data set	n	$\hat{\gamma}$	Ω	Description
`spruces`	134	0.28	56 × 38 meter	Locations of Norwegian spruces trees and diameters in a rectangle sampling region in Saxony, Germany.
BCI	300	0.32	1000 × 500 meter	Soil nutrient data for 13 different chemicals at the sampled locations in a rectangular sampling region in Barro Colorado Island.
`waka`	504	1.04	100 × 100 meter	Locations and diameters of trees in square sampling region at Waka National Park, Gabon.
`finpines`	126	1.25	10 × 10 meter	Locations and diameters of pine saplings in a Finnish forest.
`longleaf`	584	1.60	200 × 200 meter	Locations and diameters of longleaf pine trees in southern Georgia, USA.

Table 4.5: Data set name, number of points n, estimated parameter $\hat{\gamma}$, sampling region Ω, and the description of the data sets.

Ripley's K statistic

The real data sets can be diagnosed using the Ripley's K function (Ripley, 1976, 1977) that checks the spatial homogeneity (complete spatial randomness) in the data. Let X be a set of points $X = \{x_1, x_2, \ldots, x_n\}$ in two-dimensional region, then the general form of the K statistic is defined as

$$\hat{K}(r) = \frac{1}{\rho} \sum_{x_i \neq x_j \in X} \frac{\mathbb{1}\{d(x_i, x_j \leq r)\}}{n}, \tag{4.5}$$

where $d(x_i, x_j)$ is the Euclidean distance between the i-th and j-th points, $\mathbb{1}$ is the indicator function that takes values 1 if the condition is true and 0 otherwise, with a search radius r, ρ is the intensity of the points estimated as $\hat{\rho} = n/|\Omega|$ where $|\Omega|$ is the area of the region Ω within which all points are located. If the process is a homogeneous Poisson point process then $\hat{K}(r) = \pi r^2$ which indicates a complete spatial randomness whereas departure form πr^2 means clustered or dispersed pattern (Kiskowski *et al.*, 2009).

The following standardization is recommended in Besag (1977)

$$\hat{L}(r) = \sqrt{\hat{K}(r)/\pi} \tag{4.6}$$

so the expected value of L function is r for homogeneous data. The randomness of the points are tested using the hypothesis $H_o : \hat{L}(r) - r = 0$ (X follows a homogeneous Poisson process with intensity ρ). In the violation of H_o, positive values of $\hat{L}(r) - r$ indicate clustering and negative value indicate dispersion. In Chapter 4, we will consider regular and clustered points both from simulations and examples from real life data. The K function and the hypothesis would be useful to examine the pattern of real data examples particularly.

Results for the K function are shown in Figure 4.5 for the real data sets. The top-left plot in Figure 4.5 is an example of the K function from the simulated data for $\gamma = 0.25, 1, 1.5$. Plots from top-centre to the bottom-right are the K functions for real data sets with the same presentation order as in Table 4.5. In Figure 4.5 (a), which is obtained from the simulated data, the red dashed line is the expected $K(r)$ for an independent simulated homogeneous Poisson points. The black lines are obtained using point patterns when $\gamma \in \{0.28, 1, 1.50\}$. The black line above the red line when $\gamma = 1.50$ is the expected $\hat{K}(r)$ for observed data locations, indicating that the number of expected points within the search region (isotropic distance r from the data locations) is higher compared to the red line. In this case the, the line for $\gamma = 1.50$ indicates clustering. On the other hand, the black solid line obtained for $\gamma = 1$ almost overlap with the red line since $\gamma = 1$ indicates homogeneity of points. The case when $\gamma = 0.28$, the black line is always below the red line that is interpreted as the regularity.

The K function plots for each real data set are shown in Figure 4.5 (b – e) separately. The black line in Figure 4.5 (b) which is for `spruces` is under the red line that indicates regularity. The `waka` data set (c) is a clear example of homogeneous points since black line follows exactly the same pattern as the red line at different r. The `finpines` data in (d) is slightly clustered since the black curve is above the red curve. The last plot (e) indicates more clustering of locations in the `longleaf` data. The BCI data set is not included in Figure 4.5 since it is a geo-referenced data and the K function is suitable for point pattern data.

The advantage of using various types of data sets is to see how the area prediction works for such different scenarios. Therefore, the validity and the limitations of the modeling approaches that we suggested can be evaluated.

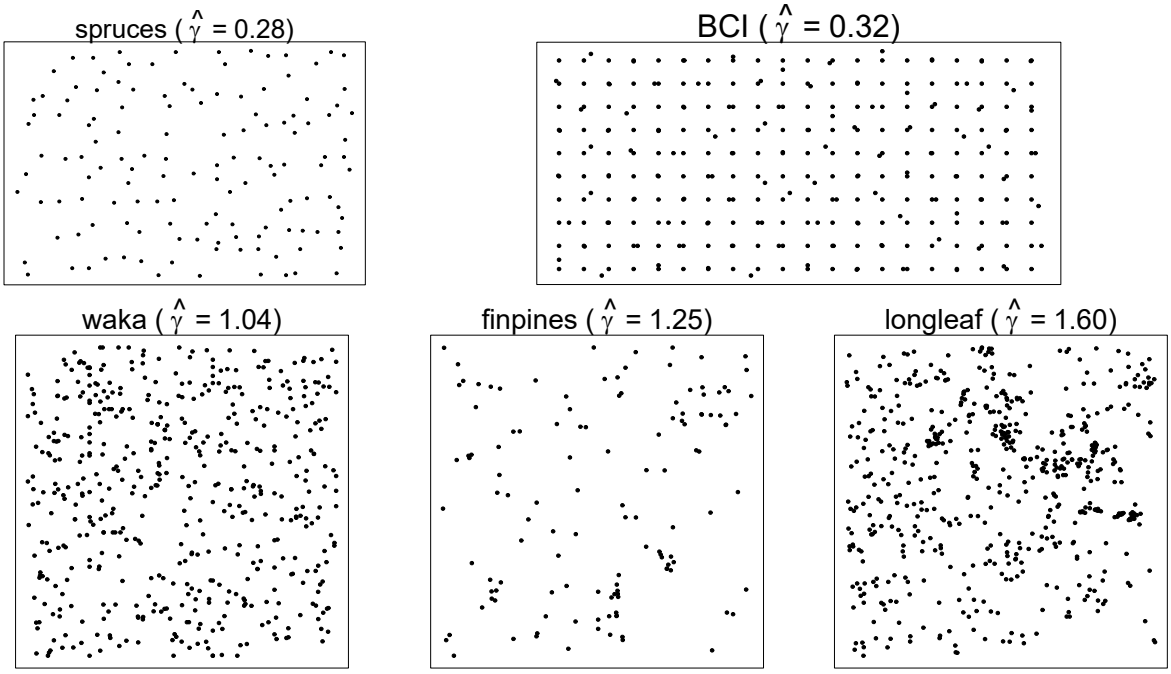


Figure 4.4: Locations of the data points in the real data sets. From top-left to bottom-right, `spruces`, Barro Colorado Island, `waka`, `finpines`, and `longleaf` data are shown. The estimated parameter $\hat{\gamma} = 0.28, 0.32, 1.04, 1.25, 1.60$ is given for each data set respectively.

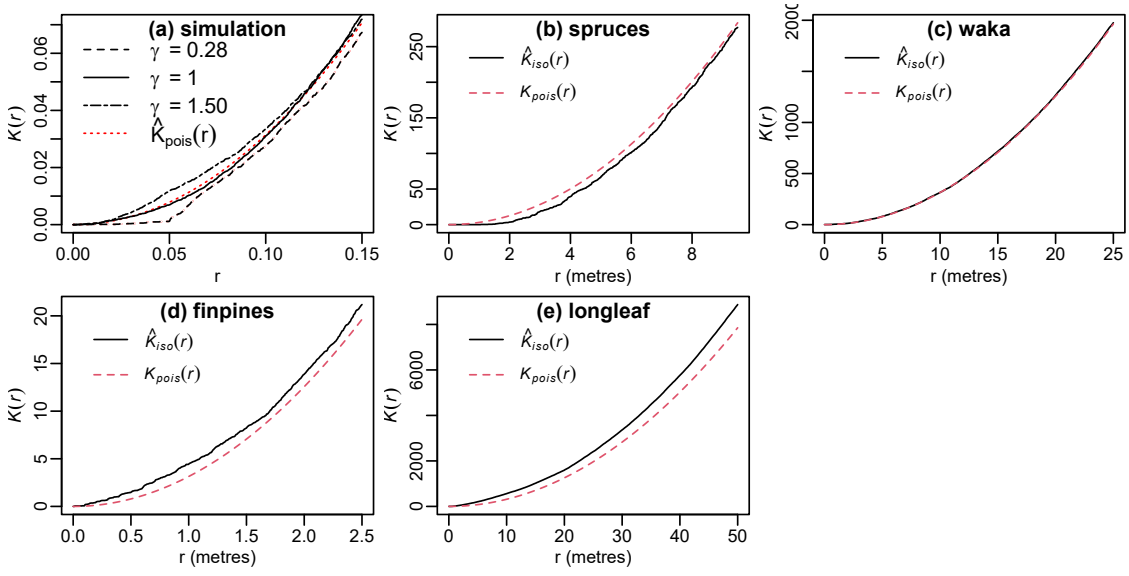


Figure 4.5: Ripley's K function plots for (a) simulated data for different values of γ, (b) `spruces`, (c) `waka`, (d) `finpines`, and (e) `longleaf` data sets. The red line is known analytically for the K-function. The black line is the expected $\hat{K}(r)$ from observed locations.

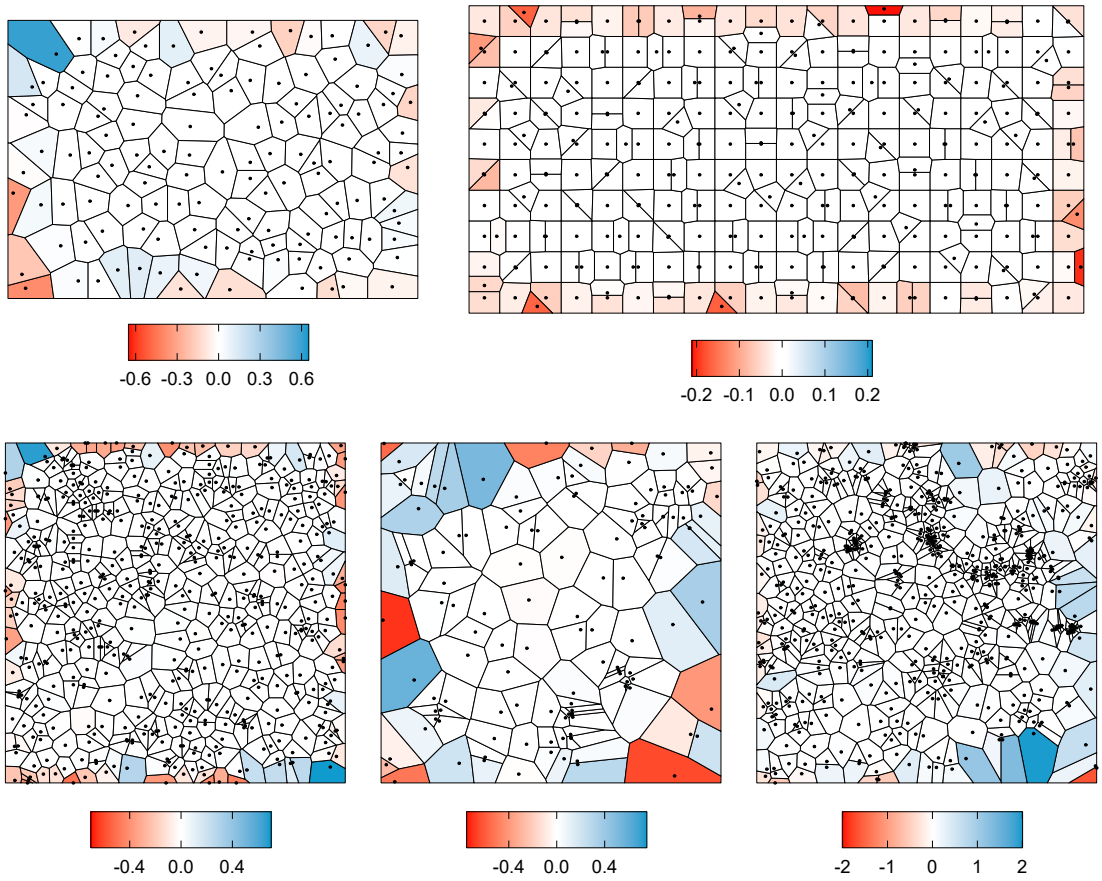


Figure 4.6: The adjustment pattern on the cell area using base $B^{\star}$ models. The difference between the observed and adjusted area is calculated as $A_i - \hat{A}_i^{\star}$ where A_i is the calculated area due to the given rectangular boundary and $\hat{A}_i^{\star}$ is the predicted area. From top-left to bottom-right, the data sets follow the same order.

Area prediction results for real data are presented in Figure 4.6 using base models, and in Figure 4.7 using the augmented models by illustrating how the cells are adjusted. It looks that the predicted area for the cells near the edges are different than the observed cell area, and the predicted area for interior cells is similar to the observed cell area. That means the edge cells are likely to be adjusted when the models are used. The blue and red coloured cells indicate shrinkage and expansion respectively. Some very large cell areas are reduced and the small ones are expanded in the prediction. As expected, interior cells are white indicating that no adjustments are happening to interior cells.

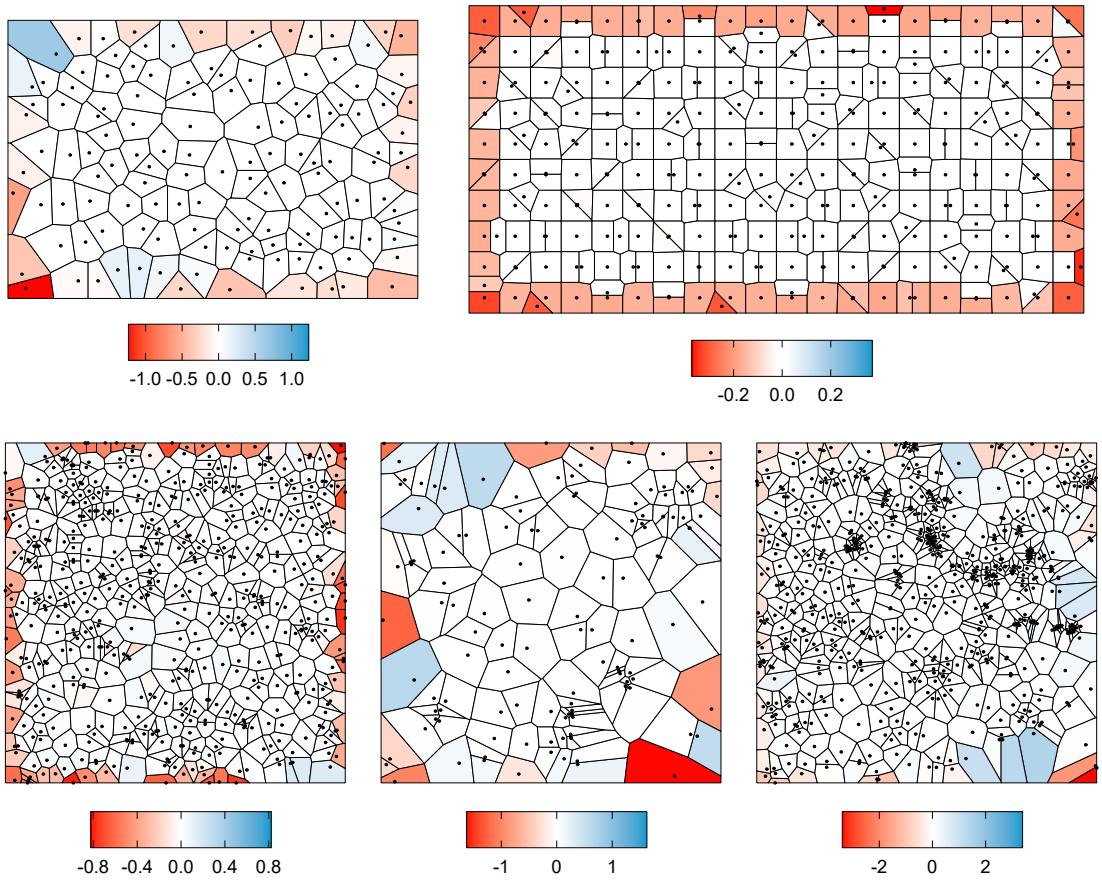


Figure 4.7: The adjustment pattern on the cell area using augmented $Ag^{\star}$ models. The difference between the observed and adjusted area is calculated as $A_i - \hat{A}_i^{\star}$ where A_i is the calculated area due to the given rectangular boundary and $\hat{A}_i^{\star}$ is the predicted area. From top-left to bottom-right, the data sets follow the same order.

4.4 Conclusion

This chapter investigates the robustness of the area prediction by testing the performances of models on data sets with misspecified intensities. The weak performance of the models are improved by using the local estimate of the intensity at data locations. The local estimate of intensity is used in the models to scale the covariates of the cells. The improvement is achieved in different degrees of regularity, and clustering of points that contain the extreme examples as well. The base models give the smallest overall MSE compared to the augmented models. However, if one would wish to reduce the maximum error, then augmented models may be preferred.

It is important to decide on the best model based on the conclusions from the simulated data and use the suggested model in the further studies where the Voronoi tessellation cell area is useful. $B^\star$ is selected as the best model to predict the area and it may be more appropriate for the real data sets. The real data sets are examples of homogeneous, regular and clustered point patterns. It is useful to apply the method on such data sets that pushes the assumptions such as having large number of points and rectangle boundary. However, the area prediction method give reasonable results in the real data sets such as expanding very small edge cells or shrinking very large edge cells. These leads to the usage of the adjusted area as an alternative weight method in lifting. We will explore this application in Chapter 6, after first explaining the lifting method in Chapter 5.

Consequently, the area prediction for Voronoi cells for homogeneous points in Chapter 3 and for the regular and clustered points in the current chapter aims to treat the Voronoi cells in the bounded region as if they are in an infinite plane by the adjustments in the cell areas. The approach we devised in Chapter 3 and 4 might be useful for the methods such as the lifting scheme that uses the Voronoi cell area. In the next chapters the area prediction approaches will be combined with the lifting scheme.

Chapter 5

Lifting scheme

In this chapter, we explain the background of the lifting scheme, and Voronoi tessellation-based lifting in the two-dimensional case particularly. Lifting is an extension of wavelet methods and has grown out for the need of a generalized version of wavelet decomposition. The lifting scheme is an instrument that we use in the remainder of this thesis as an application of the ideas and approaches we developed in the previous chapters based on Voronoi tessellations. In this background chapter, we highlight some important ideas of wavelet methods and their limitations, and explain the need for a generalized version. We explain the general framework of the lifting scheme, discuss thresholding methods and why we use them, and illustrate a few initial steps of Voronoi tessellation-based lifting. In the following chapters, we will consider using various weight methods in lifting such as the observed cell area using boundaries, and predicted cell area using the methods we discussed in Chapters 3 and 4 to reduce the boundary effects.

It is possible to understand the mathematical framework of lifting without any prior knowledge of wavelet methods, but since the wavelet theory includes the fundamental concepts that the lifting scheme is built upon, we give a short introduction to wavelet methods and explain how the lifting scheme aims to improve some of its aspects in Section 5.1. Moreover, we describe the lifting scheme in two dimensions in Section 5.3, which we will use in the remainder of this thesis in conjunction with Voronoi tessellations. We describe some important thresholding methods, and outline their usage in the context of wavelet analysis and lifting in Section 5.4. We finally give an example in Section 5.5 to show the steps of the algorithm and how the Voronoi tessellation-based lifting scheme works in two dimensions.

5.1 Background

The lifting scheme is referred to as a *second-generation wavelet method*. The general form of lifting is explained in Sweldens (1998) who defined the idea behind lifting as an iterative transformation of the data starting with localised or fine-scale details and working up to broader or coarse-scale patterns. Unlike conventional wavelet methods, lifting can be applied to irregularly spaced data with an arbitrary sample size which is often the case in reality. Lifting also relaxes the requirement of equidistant data with size $n = 2^J, J \in \mathbb{N}$ of the wavelet methods. For the preliminary studies that lead to the construction of lifting scheme, earlier work of Sweldens (1995, 1996) can be reviewed.

Lifting also has advantages over well known methods to analyze spatial data such as Gaussian process regression (kriging) or model based spatial methods. Lifting is capable of modeling irregularities such as sharp discontinuities or spikes which other methods tend to do poorly on, such as over-smoothing at the boundary of the discontinuity. Pope *et al.* (2021) aimed to reduce such issues by partitioning the sampling region using Voronoi tessellations and fitting a Gaussian process in each sub-region (Voronoi cell) separately. The method does a good job if the data has a step change, but less well in the case of repeated wiggles. However, wavelet-based or lifting-based approaches have been demonstrated to be effective approaches to deal with these kind of situations.

5.2 Discrete wavelet transform

Wavelet methods are commonly used in the estimation of functions corrupted by noise. For an introduction to wavelets, good resources include Daubechies (1992) and Vidakovic (1999). The estimation of the true function involves the transformation of the noisy data into a set of coefficients, and shrinkage/thresholding procedures to remove noise followed by the inverse transform on the modified coefficients. The lifting scheme serves the same purpose as an extension of the idea of multiresolution analysis and discrete wavelet transform (DWT) introduced by Mallat (1989). The multiresolution analysis allows the decomposition and reconstruction of noisy data.

Wavelet-based function estimation methods often assume the following model setting:

$$y_i = f(t_i) + \epsilon_i,$$

where $f(t_i)$ is the function that we are interested in at point t_i which may be a time (in one-dimensional case) or location (two-dimensional case), ϵ_i are Gaussian noise, assuming $\epsilon_i \sim N(0, \sigma^2)$ independently, and y_i are the observed noisy data.

Consider we have observations $y(t_i)$, the discrete wavelet transform assumes that $\{t_i = i/N : i = 0, 2, \ldots, N\}$ are discrete equispaced points in time or space. DWT requires $N = 2^J$ for some positive integer J hence the full transform can be carried out. If the full transform is not desired, this transform can be performed for the first J_0 steps which is called as the non-decimated DWT.

The idea of the discrete wavelet transform is to transform a vector of noisy data $\mathbf{y}$ into a vector of coefficients $\mathbf{d}$ using the low pass filter $\mathcal{H} = \{h_k\}$ and high pass filter $\mathcal{G} = \{g_k\}$ where h_k and g_k are the coefficients of filters. We first define the scaling coefficients as $c_{J,i} = y(t_i)^\top$ at level J and perform the discrete wavelet transform at levels $j = J - 1, \ldots, 0$ and by calculating

$$c_{j,i} = \sum_n h_{n-2i} c_{j+1,n} \tag{5.1}$$

$$d_{j,i} = \sum_n g_{n-2i} c_{j+1,n} \tag{5.2}$$

that gives a collection of wavelet coefficients $d_{j,i}$ and an individual coefficient $c_{0,0}$ at the end of the full transform. This transform is an orthogonal transform of the observed data y_i of length N into the wavelet domain. The y_i can be reconstructed by applying the inverse transform.

Since the discrete wavelet transform is a linear transformation of the noisy data, it can be expressed as

$$\mathbf{d} = \mathbf{W}\mathbf{y},$$

where $\mathbf{W}$ is an orthogonal matrix and multiplying the noisy data $\mathbf{y}$ by $\mathbf{W}$ gives the collection of coefficients $\mathbf{d}$.

The lifting scheme can also be explained in the same way which we will show in Section 5.3 by describing the calculation of the transform matrix in the lifting scheme context. Jansen & Oonincx (2005) explains the usage of filter banks in the lifting scheme. Further details of the wavelet transform are given in Mallat (1989), and Nason (2008) but are beyond the scope of our core mechanism *lifting one coefficient at a time* which will be discussed later.

5.3 Lifting in two dimensions

In this section, we explain and use the lifting one coefficient at a time (LOOCAT) method in two dimensions based on the description in Jansen *et al.* (2009). Another related study based on LOOCAT by Jansen *et al.* (2001) used the lifting scheme as a smoothing method for irregularly spaced data which was one of the preliminary studies after the lifting scheme is introduced in Sweldens (1998). We adopt the lifting one coefficient at a time technique introduced in (Jansen *et al.*, 2009), where data points are lifted and coefficients are calculated sequentially. The general form of lifting consists of three major steps: *split, predict* and *update*, and the lifting one coefficient at a time technique differ from earlier versions of lifting in the way that it splits the data. The standard wavelet and lifting methods use dyadic splitting based on the odd and even indices of the data such as in Claypoole *et al.* (1998) which is why the $n = 2^J$ condition is required. LOOCAT relies on a method to decide on the order of the calculations of the coefficients one by one.

There are other approaches such as the adaptive lifting in Nunes *et al.* (2006) who used the local features of the data in the prediction step when calculating the coefficients. The calculation of the coefficients includes different regression methods and ways of defining neighbours, and they choose the configuration that gives the smallest absolute value of the coefficient. Non-decimated lifting, introduced in Knight & Nason (2009), considers paths or trajectories that are $n!$ possible lifting orders rather than relying on one. However, they sub-sample a smaller number of paths from $n!$ trajectories and obtain a set of coefficients at each point rather than one coefficient. The lifting scheme is also used in Heaton & Silverman (2008) in the context of imputation.

Now let us explain the lifting scheme based on Voronoi tessellations proposed by Jansen *et al.* (2009). Recall the settings where $\mathbf{x}$ is a set of data locations $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ which are n irregularly spaced points in a two-dimensional space, that is $\{x_i\}_{i=1}^n \in \mathbb{R}^2$. At each point x_i, we observe noisy data y_i. Now we assume the model

$$y_i = f(x_i) + \epsilon_i \tag{5.3}$$

where f_i are the values of an underlying true function which we are interested in but only the data y_i are available, and $\epsilon_i \sim N(0, \sigma^2)$ are independent Gaussian noise. LOOCAT aims to transform the vector of noisy data values into a set of coefficients by calculating each coefficient at a time. The operation of transforming an individual data point into a coefficient is referred to as *lifting* that observation.

The lifting method has two important aspects: the order of lifting the observations and the neighbourhood structure of the points. Therefore, the decision of the lifting order and the determination of the neighbourhood structure is crucial. This is the first part where Voronoi tessellation is used in the lifting scheme. More importantly, using the Voronoi tessellation cell areas attributed to points $\{x_i\}_{i=1}^n$, we perform the split, predict and update stages in lifting that are explained in the next section in detail.

5.3.1 Steps of the lifting transform

The steps of the lifting scheme are explained in this section. These steps are the intermediate calculations in the lifting transform that eventually maps the noisy data into a set of coefficients in the lifting domain. Since the transform is linear and can be expressed as a non-singular matrix, it is easily invertible. That means the noisy values can be recovered exactly from the coefficients. The inversion can also be accomplished by following the lifting algorithm in reverse.

The lifting transform is an iterative process, and we repeatedly perform the steps until we have a small number of non-lifted points. We start the (forward) transform by selecting the first data location to be lifted; this selection process is called the *splitting* step. Then we predict the observed value at the selected point from its neighbours, which is the *prediction* step. Finally the values of the neighbours are updated in the *update* step, and the selected point is *lifted*.

Let r be the current stage of the lifting transform. We first set $r = n$ and increment by -1 at each step until $r = l + 1$, where l is the number of points to keep (not to lift). The general form of the Voronoi tessellation-based lifting scheme has the following steps:

1. At stage r of the lifting transform, we identify the next point to be lifted, which is the point with the smallest Voronoi cell area. This ensures we lift the point with the finest level of detail in the data. The point with the smallest cell area is likely to have nearby neighbours that tend to have similar characteristics with the neighbours unless there is a large local feature in the data. Hence the noise at the selected point can be well detected by the values of its neighbours. Also, the coefficient that we obtained for that point has the information that is representative only for a small region. At stage $r = n$ we

select the smallest cell by

$$i^r = \underset{i \in \{1,\ldots,n\}}{\operatorname{argmin}} I_{r,i} \tag{5.4}$$

where i^r is the index number of the selected point at stage r, and $I_{r,i}$ is the area of the Voronoi cell associated with point x_i at stage r. Then, let J_r denote the set of indices of neighbours of the selected point x_{i^r}.

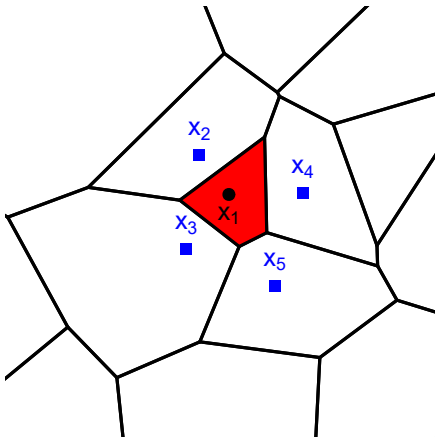


Figure 5.1: An illustration of the neighbourhood structure of a selected point x_1 given that $i^r = 1$, and its neighbours x_2, x_3, x_4, x_5 such that $J_r = \{2, 3, 4, 5\}$.

This first step is illustrated in Figure 5.1. The selected point $x_{i^r} = x_1$ at stage r is shown as (•) and its neighbours $J_r = 2, 3, 4, 5$ with (■) points. The index numbers of the points in Figure 5.1 are arbitrarily chosen to illustrate the neighbourhood more clearly. In this step, the red cell has the smallest area and we predict the value at x_{i^r} using its neighbours J_r at the next step.

2. Now, we predict the value y_{i^r} of the selected point x_{i^r} by $\hat{y}_{i^r} = \mathbf{a}^\top \mathbf{y}_{J_r}$ where $\mathbf{y}_{J_r}$ is a vector of values of the neighbours J_r, and $\mathbf{a}$ is the vector of prediction weights obtained from a regression procedure over the neighbours J_r which will be discussed later in detail in Section 5.3.2. We then calculate the detail coefficient d_{i^r}, which is the difference between the observed and predicted value,

$$d_{i^r} = y_{i^r} - \hat{y}_{i^r}. \tag{5.5}$$

3. We update the values of the neighbours of the removed point, using the detail coefficient d_{i^r} by setting

$$\mathbf{y}^\star_{J_r} = \mathbf{y}_{J_r} + d_{i^r}\mathbf{b}, \tag{5.6}$$

where the elements of vector $\mathbf{b}$ are calculated by

$$b_j = \frac{I_{r,i^r} I_{r-1,j}}{\sum_{k \in J_r} I_{r-1,k}^2}. \tag{5.7}$$

Here $I_{r-1,j}$ is the cell area of the neighbours after the point x_{i^r} is lifted, so the area of I_{r,i^r} will be shared by its neighbours. We define $I_{r-1,j} = I_{r,j} + a_j I_{r,i^r}$ for all $j \in J_r$, where $r-1$ indicates the next stage of lifting transform. The a_j are the values of the vector of weights $\mathbf{a}$ in Step 2.

4. Finally, we remove x_{i^r} from the entire data set and return to the first step, recalculating the Voronoi tessellation of points. From the remaining data, we choose the next point that has the smallest cell area and follow the predict and update steps accordingly until we are left with l non-lifted points.

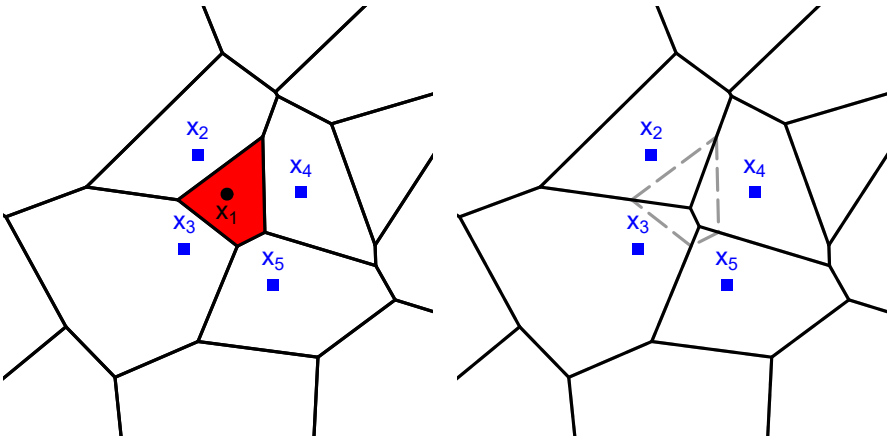


Figure 5.2: An illustration of the neighbourhood structure of a selected point x_1 (left), and the change in the cells of the neighbours x_2, x_3, x_4, x_5 after x_1 is removed (right).

We illustrate the appearance of the cells of the neighbours x_2, x_3, x_4, x_5 after x_1 is removed from the data set in Figure 5.2 (right). We show the cell edges of x_1 in gray, and how its area is shared by its neighbours after x_1 is removed. The neighbouring cells are expanded in the new tessellation in the absence of x_1. The expansion of neighbouring cells is denoted as $I_{r-1,j} = I_{r,j} + a_j I_{r,i^r}$ where the first term indicates the original cell area, and the second term is the part gained from I_{r,i^r} which will be clarified in Section 5.3.2. Algorithm 1 shows the pseudo code of the forward transform.

Algorithm 1: Lifting transform

Input: Points $\mathbf{x} = x_i$, function values $\mathbf{y} = f(x_i) + \epsilon_i$ for $i \in \{1, ..., n\}$ and $\epsilon_i \sim N(0, \sigma^2)$ is the Gaussian noise.

Decide l, the number of points to keep

Let the stages of the transform be $r = \{n, n-1, ..., l+1\}$

for $r = n$ **to** $l + 1$ **do**

- Partition the space into Voronoi cells V_i
- Calculate cell area $I_{r,i}$ for cells V_i
- **Splitting step:** Find the cell with smallest area
  - Choose $i^r = \text{argmin}_{i \in \{1,...,n\}} I_{r,i}$
  - Determine the set of neighbours J_r
- **Prediction step:** Calculate the detail coefficient
  - $d_{i^r} = y_{i^r} - \mathbf{a}^\top \mathbf{y}_{J_r}$ where $a_j = \frac{I_{i^r,j}}{I_{i^r}}$, for all $j \in J_r$
- **Update step:** Update the function values of neighbours
  - $\mathbf{y}^\star_{J_r} = \mathbf{y}_{J_r} + d_{i^r}\mathbf{b}$ where $b_j = \frac{I_{r,i^r} I_{r-1,j}}{\sum_{k \in J_r} I^2_{r-1,k}}$ and $I_{r-1,j} = I_{r,j} + a_j I_{ri^r}$
- Remove x_{i^r}

Output: Detail coefficients $\mathbf{d} = \{d_{i^r}, d_{i^{r-1}}, ..., d_{i^{l+1}}\}$, lifting order $\mathbf{s} = \{i^r, i^{r-1}, ..., i^{l+1}\}$, remaining points x_i and function values f_i for all $i \notin \mathbf{s}$, and the transform matrix $\mathbf{L}$.

5.3.2 Methods of prediction

For the LOOCAT algorithm based on Voronoi polygons in two-dimensional space, Jansen *et al.* (2009) discussed two prediction schemes, natural neighbour interpolation and local least squares prediction. As mentioned at Step 2 in Section 5.3.1, for the selected point x_{i^r} at stage r, we aim to predict y_{i^r} by a weighted average of the values of its neighbours y_{J_r} specifying the prediction weights $\mathbf{a}$ at each stage.

When the point x_{i^r} is removed, the Voronoi tessellation of the remaining points at the next stage $r-1$ can be recomputed. However, Voronoi cell V_{i^r} will disappear and its area is shared by its neighbours J_r. Let us carry on the explanation assuming the selected point is x_1 and its neighbours are x_2, x_3, x_4, x_5 as previously mentioned. In a finite region Ω, let $V_{i^r,j}$ be the part of V_{i^r} which joins to neighbour $j \in J_r$. We adopt the natural neighbourhood interpolation explained in Jansen *et al.* (2009) that works by setting

$$a_j = \frac{|V_{i^r,j}|}{|V_{i^r}|} \tag{5.8}$$

where $|.|$ denotes the area of the Voronoi cell. Note that $\sum a_j = 1$ by definition and $0 < a_j \leq 1$ for all $j \in Jr$.

In the examples from Figure 5.1 and 5.2, let $V_1 = V_{1,2} \cup V_{1,3} \cup V_{1,4} \cup V_{1,5}$ be the Voronoi cell of x_1, and let $V_{1,j}, j \in 2, \ldots, 5$ be the divided parts of V_1, and $V_2, \ldots, V_5$ be the

cells of the neighbours. The division of the cell V_1 by the neighbours is illustrated in Figure 5.3.

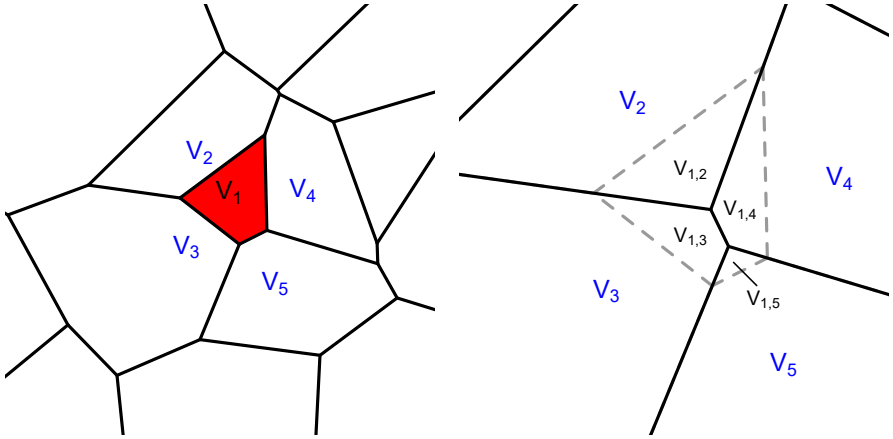


Figure 5.3: An illustration of the calculation of weights based on partitioned cell of the removed point.

In this case, the prediction weights are calculated as

$$\mathbf{a}^\top = \left(\frac{|V_{1,2}|}{|V_1|}, \frac{|V_{1,3}|}{|V_1|}, \frac{|V_{1,4}|}{|V_1|}, \frac{|V_{1,5}|}{|V_1|} \right). \tag{5.9}$$

Using the weights a_j from the natural neighbour interpolation, the detail coefficient for x_1 is calculated as

$$d_1 = y_1 - (a_2 y_2 + a_3 y_3 + a_4 y_4 + a_5 y_5).$$

Natural neighbour interpolation can also be expanded to $d = 1$ or $d \geq 3$ dimensional cases. Computational intensity is the only disadvantage of this method.

Another prediction method called local least squares is a computationally simpler approach. In stage r, a least squares plane is fitted to the selected site i^r and its neighbours J_r, however, this method is not interpolating. Limitations arise when a point is very close to one of its neighbours; more distant neighbours will still have a high influence. Hence, as a more stable method, natural neighbourhood interpolation is recommended when calculating the detail coefficients in Jansen *et al.* (2009).

5.3.3 Derivation of transform matrix

The lifting transform can be represented by a transform matrix which is independent of the observations or the function values and only depends on the data locations.

Therefore, once the transform matrix is obtained, it can be reused for different observed values at the same locations. In this section, the construction of the transform matrix will be explained. Recall that we have data locations $x_i \in \mathbb{R}^2$ for $i = 1, \ldots, n$, and observe y_i at each location. While performing the lifting transform as explained in Section 5.3.1, a transform matrix $\mathbf{L}$ can be constructed simultaneously. Then the split, predict, and update steps of the lifting transform can be achieved through pre-multiplication of the data vector $\mathbf{y}$ by the transform matrix $\mathbf{L}$ to obtain the vector of detail coefficients $\mathbf{d}$:

$$\underbrace{\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}}_{\mathbf{d}} = \underbrace{\begin{bmatrix} w_{11} & w_{12} & w_{13} & \ldots & w_{1n} \\ w_{12} & w_{22} & w_{23} & \ldots & w_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & w_{n3} & \ldots & w_{nn} \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}}, \tag{5.10}$$

where the elements w_{ij} of $\mathbf{L}$ depend on the cell areas and data locations. This is the general form of $\mathbf{L}$ that allows us to calculate detail coefficients as $\mathbf{d} = \mathbf{L}\mathbf{y}$ for any observed data vector $\mathbf{y}$. The transform matrix $\mathbf{L}$ can also be used for the inverse transform to reconstruct the observed data $\mathbf{y} = \mathbf{L}^{-1}\mathbf{d}$. The inverse transform is more useful to invert the adjusted or thresholded detail coefficients which will be discussed in Section 5.4.

We start the construction of the transform matrix by initializing the transform matrix as being an identity matrix $\mathbf{L} = \mathbf{I}_{n\times n}$ and hence $\mathbf{y} = \mathbf{L}\mathbf{y}$ which returns the same vector of observed data $\mathbf{y}$ as

$$\overbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}^{\mathbf{d}^n} = \overbrace{\begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix}}^{\mathbf{L}^n} \overbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}^{\mathbf{y}^n}. \tag{5.11}$$

Assume that the lifted point is x_1 at the first stage $r = n-1$ of the lifting transform and x_2, x_3, and x_4 are its neighbours. Then the transform matrix $\mathbf{L}^{n-1}$ at stage

$r = n - 1$ takes the form

$$\overbrace{\begin{bmatrix} d_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_n \end{bmatrix}}^{\mathbf{d}^{\star(n-1)}} = \overbrace{\begin{bmatrix} 1 & -a_2 & -a_3 & -a_4 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix}}^{\mathbf{L}^{\star(n-1)}} \overbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_n \end{bmatrix}}^{\mathbf{y}^{(n)}} \tag{5.12}$$

where the vector $\mathbf{d}^{\star(n-1)}$ only includes the detail coefficient d_1 which is calculated from the first row of $\mathbf{L}^{\star(n-1)}$ and vector $\mathbf{y}^{(n)}$ as $d_1 = y_1 - (a_2 y_2 + a_3 y_3 + a_4 y_4)$ which is the first value of $\mathbf{d}^{\star(n-1)}$ and the remaining values in $\mathbf{d}^{\star(n-1)}$ are the original observed values $y_2, \dots, y_n$.

To update the values of neighbours y_2, y_3, y_4, we make the following calculation

$$\overbrace{\begin{bmatrix} d_1 \\ y_2^\star \\ y_3^\star \\ y_4^\star \\ \vdots \\ y_n \end{bmatrix}}^{\mathbf{d}^{(n-1)}} = \overbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ b_2 & 1 & 0 & 0 & \dots & 0 \\ b_3 & 0 & 1 & 0 & \dots & 0 \\ b_4 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix}}^{\mathbf{L}^{(n-1)}} \overbrace{\begin{bmatrix} d_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_n \end{bmatrix}}^{\mathbf{d}^{\star(n-1)}} \tag{5.13}$$

hence the vector $\mathbf{d}^{(\mathbf{n-1})}$ includes the first detail coefficient d_1 and the updated values $y_2^\star, y_3^\star, y_4^\star$ and remaining values $y_5, \dots, y_n$. The a_j and b_j are the coefficients obtained from the calculation explained in Section 5.3.1. The point x_1 is lifted and can no longer be a lifted point or a neighbour in the further stages.

Now we move on to the next step $r = n - 2$ of the transform. Let us assume the next point to be lifted is x_2 and its neighbours are x_3, x_4, and x_5 at stage $r = n - 2$. This stage is similarly performed by calculating $\mathbf{d}^{\star(n-2)}$ as

$$\overbrace{\begin{bmatrix} d_1 \\ d_2 \\ y_3 \\ y_4 \\ y_5 \\ \vdots \\ y_n \end{bmatrix}}^{\mathbf{d}^{\star(n-2)}} = \overbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & -a_3 & -a_4 & -a_5 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix}}^{\mathbf{L}^{\star(n-2)}} \overbrace{\begin{bmatrix} d_1 \\ y_2^\star \\ y_3^\star \\ y_4^\star \\ y_5 \\ \vdots \\ y_n \end{bmatrix}}^{\mathbf{d}^{(n-1)}}$$

where $\mathbf{d}^{(n-1)}$ include the information from the previous stage $r = n-1$ but we overwrite it and do not need to keep in the memory. Then the updated values of the neighbours $y_3^\star, y_4^\star, y_5^\star$ are calculated as

$$\overbrace{\begin{bmatrix} d_1 \\ d_2 \\ y_3^\star \\ y_4^\star \\ y_5^\star \\ \vdots \\ y_n \end{bmatrix}}^{\mathbf{d}^{(n-2)}} = \overbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & b_3 & 1 & 0 & 0 & \dots & 0 \\ 0 & b_4 & 0 & 1 & 0 & \dots & 0 \\ 0 & b_5 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix}}^{\mathbf{L}^{(n-2)}} \overbrace{\begin{bmatrix} d_1 \\ d_2 \\ y_3 \\ y_4 \\ y_5 \\ \vdots \\ y_n \end{bmatrix}}^{\mathbf{d}^{\star(n-2)}}$$

Now we have the vector $\mathbf{d}^{\star(n-2)}$ that contain the detail coefficients d_1, d_2 and updated values $y_3^\star, y_4^\star, y_5^\star$. The transform continues by the selection of the next point to lift and so on until we have l points left. The idea is the same as in the first two steps and the full transform is performed by the following matrix multiplications. Collapsing the first two stages $r = n-1, n-2$ into a more compact from, we can show how the transform matrix $\mathbf{L}$ is constructed as

$$\begin{aligned}
\mathbf{d}^{(n-1)} &= \mathbf{L}^{(n-1)} \underbrace{\mathbf{L}^{\star(n-1)}\mathbf{y}}_{\mathbf{d}^{\star(n-1)}} \\
\mathbf{d}^{(n-2)} &= \mathbf{L}^{(n-2)} \underbrace{\mathbf{L}^{\star(n-2)}\mathbf{L}^{(n-1)}\mathbf{L}^{\star(n-1)}\mathbf{y}}_{\mathbf{d}^{\star(n-2)}} \\
\mathbf{d}^{(n-3)} &= \mathbf{L}^{(n-3)} \underbrace{\mathbf{L}^{\star(n-3)}\mathbf{L}^{(n-2)}\mathbf{L}^{\star(n-2)}\mathbf{L}^{(n-1)}\mathbf{L}^{\star(n-1)}\mathbf{y}}_{\mathbf{d}^{\star(n-3)}} \\
&\vdots \\
\mathbf{d}^{(l)} &= \mathbf{L}^{(l)}\mathbf{L}^{\star(l)} \ \dots \ \mathbf{L}^{(n-1)}\mathbf{L}^{\star(n-1)}\mathbf{y}.
\end{aligned} \tag{5.14}$$

We obtain the final vector of detail and scaling coefficients (updated values) $\mathbf{d}^{(l)}$ on the left hand side of (5.14), and the final form of the transform matrix $\mathbf{L}$ in equation (5.10) is calculated as

$$\mathbf{L} = \mathbf{L}^{(l)}\mathbf{L}^{\star(l)}\mathbf{L}^{(l+1)}\mathbf{L}^{\star(l+1)} \ \dots \ \mathbf{L}^{(n-2)}\mathbf{L}^{\star(n-2)}\mathbf{L}^{(n-1)}\mathbf{L}^{\star(n-1)}. \tag{5.15}$$

As mentioned previously, $\mathbf{L}$ only depends on the locations x_i. One would prefer the matrix calculation $\mathbf{d} = \mathbf{L}\mathbf{y}$ to calculate the detail coefficients instead of running the full lifting transform each time. This is useful in the case of multiple vectors of observed data $\mathbf{y}^i, \mathbf{y}^{ii}, \dots$ observed at the same data locations x_i. This choice reduces the computational cost since the multiplication of the matrix with a vector

will be faster than running the steps of the lifting transform each time. However, note that we have to run the full transform once to obtain the matrix **L**.

5.3.4 Implementation of 2D lifting in R

In this section, we explain the implementation of two-dimensional lifting in R programming language, (R Core Team, 2021). Available libraries to perform lifting in R do not have the option of Voronoi tessellation-based lifting in two-dimensional cases. Current packages allow adaptive lifting, using the `adlift` package (Nunes & Knight, 2018), and nondecimated lifting transform using `nlt` package (Knight & Nunes, 2018), however, they are only useful for lifting in one-dimensional cases and their mathematical framework is not suitable for our two-dimensional case. Therefore, we wrote our own function `lift2D` to implement lifting on two-dimensional data. The `lift2D` function has various options to choose the type of the boundary, and to assign any vector of weights including the Voronoi cell area-based weights.

The routines to perform the two-dimensional lifting in `lift2D` function requires the key packages `deldir` and `tripack` introduced by Turner (2021) and Gebhardt *et al.* (2020) respectively to compute the Voronoi tessellation of points and to extract information such as the identification of neighbourhood structure, calculation of cell area, etc. Another package `rgeos` by Bivand & Rundel (2020) is used to intersect the polygons with the boundaries during the intermediate steps of the lifting transform.

The `lift2D` function has the following input structure:

```
lift2D(x, y, f, nleft, stage, keepnbrs, Lmat, rw, method, ... )
```

where vectors `x` and `y` are the coordinates of the data locations, and `f` is the vector of observed values at the locations. The number of points to leave (not to lift) l is defined as `nleft`. The `stage` option is to save the data in the selected stages of lifting such as `stage=c(90, 70, 50, 25, 12)` that saves the coordinates of the remaining points and the updated values of `f` at the specified stage. This is useful to check and illustrate the different stages of the algorithm. If `keepnbrs = TRUE`, then the indices of the neighbours of the lifted point at each stage is recorded. An important option `Lmat` specifies whether to calculate and output the transform matrix **L**. The `rw` is the window of observed locations such as the rectangular boundary. If known, it can be specified, otherwise convex hull of points may be used. Finally the `method` option is used to specify the weights to be used in the transform. The weights we use in this thesis are based on the Voronoi cell area which are discussed in Section 6.2 but any vector of weights can be specified.

We designed the `lift2D` function to perform the Voronoi tessellation-based lifting given a set of irregularly spaced data in two dimensions when the boundaries are taken into account. Cell area-based weights can be specified in multiple ways such as the cell area calculated directly from the Voronoi tessellation. However, the method we introduced in Chapter 3 allows us to reduce the boundary affect by adjusting the cell area. The adjusted cell area can also be used in the lifting framework as an alternative weight method; we shall investigate this in Chapter 6 and 7. Therefore, we can investigate the performances of different weight methods in lifting. The `method` in the function option allows us to specify these different weights. However, it is important to note that the use of different weights will create different transform matrices $\mathbf{L}$ since the weights are used in the steps of the lifting algorithm to calculate the coefficients a_j and b_j and to determine the lifting order.

5.4 Shrinkage in lifting

Wavelet methods and lifting are often used in situations where one would like to analyze a data corrupted by noise and estimate the underlying true patterns in the data. The purpose of thresholding is to identify the coefficients that represent only noise, and hence follow a $N(0, \sigma^2)$ distribution. Thresholding schemes assume that the small empirical coefficients are due to the small variations in the data (noise), hence we assume their true values to be zero. The large coefficients are kept or adjusted depending on the thresholding technique since they are considered to be due to activity in the true function f, and are referred to as the signal.

We have discussed the lifting scheme that transforms data that contain iid Gaussian noise $y_i = f(x_i) + \epsilon_i$ into a vector of coefficients $\mathbf{d}$ in the lifting domain. The resulting transform gives a vector $\mathbf{d}$ which includes a collection of detail and scaling coefficients. Usually, $\mathbf{d}$ is a sparse representation of $\mathbf{y}$ in the wavelet or lifting domain. However, we are interested in the estimation of $\hat{f}_i$ which is achievable using the shrinkage techniques. This is done by the inverse transform of the thresholded coefficients to have an estimate $\hat{f}$ of the function f.

Donoho & Johnstone (1994) and Donoho *et al.* (1995) suggested thresholding or shrinking the coefficients in $\mathbf{d}$. Thresholding methods aim to identify and modify the coefficients that represent noise, and preserve the coefficients that represent the actual activity in the underlying pattern. The coefficients that are smaller than a threshold are assumed to be due to the noise in the data and these coefficients are shrunk to zero. Larger coefficients are either kept unchanged or also adjusted but

not shrunk to zero depending on the thresholding technique. The estimation of $\hat{f}_i$ is then followed by the inversion of the modified coefficients.

The shrinkage techniques in Donoho & Johnstone (1994) and Donoho *et al.* (1995) are known as the wavelet shrinkage, but it can be easily extended to the lifting context. Let us consider the setting in (5.3) such that we have the noisy data $y_i = f(x_i) + \epsilon_i$ observed at the data locations $x_i \in \mathbb{R}^2$ for $i = 1, ..., n$, where the underlying true function values $f(x_i)$ are corrupted by the Gaussian noise term $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$.

Using the transform matrix $\mathbf{L}$, we are able to calculate the detail coefficients from the noisy observations y_i. Let us show how the noise component is translated into the lifting domain. Let us expand the notation $\mathbf{d} = \mathbf{L}\mathbf{y}$ as

$$d_i = \mathbf{L}y_i = \mathbf{L}(f(x_i) + \epsilon_i) = \mathbf{L}(f(x_i)) + \mathbf{L}(\epsilon_i) = d_i^* + \varepsilon_i$$

where ε_i is the lifting transform of the noise component and $\varepsilon \sim N(0, \sigma_\varepsilon^2)$ where

$$\varepsilon = \mathbf{L}\epsilon \sim N(\mathbf{L}0, \mathbf{L}^\top \sigma^2 \mathbf{L}) = N(0, \sigma^2 \mathbf{L}^\top \mathbf{L}).$$

Hence the variance of the noise component is denoted $\sigma_\varepsilon^2 = \sigma^2 \mathbf{L}^\top \mathbf{L}$ and it is $\sigma_\varepsilon^2 = \sigma^2 \mathbf{I}$ if the lifting transform is orthogonal $\mathbf{L}^\top \mathbf{L} = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix. The wavelet transform is orthogonal but it is not always the case for the lifting.

Thresholding methods are applied on d_i to have an estimate $\hat{d}_i = t(d_i)$ where t stands for the thresholding scheme being used. Next, the small coefficients are assumed to be zero and coarser coefficients are kept unchanged that is the stage where the noise is suppressed. The threshold δ is estimated from the data itself which will be discussed later. Three thresholding rules are widely used in the wavelets and lifting literature; hard, soft and empirical Bayes thresholding.

5.4.1 Hard and soft thresholding

The hard and soft thresholding methods are introduced in Donoho & Johnstone (1994). In the hard thresholding method, a '*kill or keep*' strategy is adopted. If the absolute value of a detail coefficient is larger than the threshold, it is not changed, otherwise, it is set to zero. A detail coefficient d is thresholded based on the threshold δ as

$$\hat{d} = t_{H,\delta}(d) = \begin{cases} d & \text{if} \quad |d| \geq \delta \\ 0 & \text{if} \quad |d| < \delta \end{cases}. \tag{5.16}$$

On the other hand, the soft thresholding applies an adjustment on the coefficients larger than the threshold by reducing them by δ,

$$\hat{d} = t_{S,\delta}(d) = \begin{cases} (|d| - \delta)sgn(d) & \text{if} \quad |d| \geq \delta \\ 0 & \text{if} \quad |d| < \delta \end{cases}. \tag{5.17}$$

One popular choice for δ is the universal threshold defined as

$$\delta = \sqrt{2\log(N-l)\zeta^2} \tag{5.18}$$

in Donoho & Johnstone (1994) where $(N - l)$ is the number of coefficients. The variance of the noise ζ is usually estimated from the median absolute deviation of finest-scale detail coefficients from zero.

5.4.2 Empirical Bayesian thresholding

Empirical Bayesian threshold (Johnstone & Silverman, 2004, 2005a) has great adaptivity features and has been widely used in the lifting scheme. We assume the model of observations $Z_i = h(t_i) + \varepsilon_i$ where the noise is $\varepsilon \sim N(0, \sigma^2)$ independently. Consider we have a parameter θ and an observation $Z \sim N(\theta, 1)$ where the parameter θ is the lifting coefficient of h hence the variance can be scaled if $\sigma^2 \neq 1$ to have unit variance. In the context of this approach, sparsity is modeled through a suitable prior distribution of independent θ_i as

$$f_{prior}(\theta) = (1 - \omega)\delta_0(\theta) + \omega\gamma(\theta)$$

where ω is the mixing weight and γ is a symmetric, uni-modal density. One way to estimate thresholded coefficient is the posterior median $\hat{\theta}(z;\omega)$ which is a monotonic function of z and there exist a function $t(\omega) > 0$ such that $\hat{\theta}(z;\omega) = 0$ if and only if $|z| \leq t(\omega)$. Hence for each observation $Z_i = z_i$, the posterior distribution, $f_{post}(\theta_i|Z_i = z_i)$ can be calculated.

The mixing weight w, or the threshold $t(\omega)$ can be specified by letting $\phi(z)$ be the standard normal distribution and defining $g = \gamma \star \phi$ where $\star$ denotes convolution. The marginal density of Z is

$$Z \sim (1 - \omega)\phi(z) + \omega g(z). \tag{5.19}$$

The maximum likelihood estimator of $\hat{\omega}$ of ω can be obtained by maximizing the

marginal log-likelihood

$$l(\omega) = \sum_{i=1}^{n} \log\{(1-\omega)\phi(z) + \omega g(z_i)\}. \tag{5.20}$$

In order to prevent the empirical Bayesian threshold being greater than the universal threshold, we set a restriction $t(\omega) \leq \sqrt{2\log n}$ which assures the removal of all pure-noise coefficients. The posterior distribution of $\theta|X = x$ can be expressed as

$$f_{post}(\theta|X = x) = (1 - \omega_{post})\delta_0(\theta) + \omega_{post} f_1(\theta|x), \tag{5.21}$$

where $\omega_{post}(x) = P(\theta \neq 0|X = x)$ and $f_1(\theta|X = x) = f(\theta|X = x, \theta \neq 0)$. Let $\beta(x) = \frac{g(x)}{\phi(x)} - 1$, then the posterior probability is defined as

$$\omega_{post}(x) = \frac{1 + \beta(x)}{\omega^{-1} + \beta(x)}.$$

We define the posterior median $\hat{\theta}(x; w)$ by considering

$$\tilde{F}_1(\theta|x) = \int_{\theta}^{\infty} f_1(u|x)du.$$

Thus, if $x \geq 0$

$$\begin{cases} \hat{\theta}(x; w) = 0 & \text{if} \quad \omega_{post}(x)\tilde{F}_1(0|x) \leq 1/2 \\ \tilde{F}_1\big(\theta(x; w)|x\big) = 1/2\omega_{post} & \text{otherwise.} \end{cases} \tag{5.22}$$

If $x < 0$, then $\theta(x; w) = -\theta(-x; w)$ by the anti-symmetry property.

There is an implementation of empirical Bayes thresholding in the `EBayesThresh` package in `R` (Johnstone & Silverman, 2005b). The `EBayesThresh` function thresholds each coefficient using an empirical Bayesian procedure instead of fixing the threshold.

These thresholding methods are widely used in the lifting literature. However, there are various other methods available such as block thresholding in Cai (1999, 2002) and Hall *et al.* (1999) that considers thresholding the coefficients in groups instead of individual adjustments, NeighBlock and NeighCoeff (Cai & Silverman, 2001) that is built upon block thresholding and takes neighbouring coefficients into account, SureShrink, based on Stein's Unbiased Risk Estimator (SURE) introduced in Donoho & Johnstone (1995) based on (Stein, 1981) that chooses the threshold by minimizing the SURE, cross validation based thresholding to find the optimal

threshold parameter in Nason (1996) and Jansen & Bultheel (1999). Antoniadis *et al.* (2001) gave an extensive comparative study considering many alternative methods in addition to the ones listed above.

5.5 Example

In this section, an example is given to show the calculations of the coefficients in the first few steps of the lifting transform, and to perform the thresholding and the estimation of the underlying function. In this example, we first generate $n = 100$ uniform random points inside a unit square sampling region $\Omega = [0, 1]^2$. Locations of data points $\mathbf{x} = \{x_1, x_2, \ldots, x_{100}\}$ are shown in Figure 5.4. If the points were considered in an infinite plane where no points were outside the unit square, some of the edge cells would have infinite areas, and some would have finite but extremely large areas. This would affect the steps of the lifting and misguide us on the calculations. Since the cell areas are used in the lifting scheme, use of boundaries is important. Here we show two simple types of boundaries. The unit square is shown as the black square, and the convex hull of points is shown as the large red polygon. These options are considered as the boundaries and the cell areas due to these imposed boundaries are calculated. In this example we only use the unit square boundary as an illustration but we will consider various options later in Chapters 6 and 7.

At each point x_i, we observe some function value $\mathbf{f} = \{f_1, f_2, ..., f_{100}\}$ by taking $f_i = f(x_i)$ based on a two-dimensional function called Doppler. The formula of the Doppler test function is given in (B.1) and it is explained with other test functions in Section 6.1 in detail. We artificially add iid Gaussian noise $\epsilon \sim N(0, 0.2)$ to the test function, and obtain the noisy observations $y_i = f(x_i) + \epsilon_i$. The vector of noisy function values are $\mathbf{y} = \{0.741, 0.019, \ldots, -0.765, 1.133\}$. The locations and the noisy function values at the data locations are shown in Figure 5.5. The `R` package `ggvoronoi` by Garrett *et al.* (2021) is used to create the Voronoi tessellation plots with coloured cells with the colour scheme throughout the thesis.

The index numbers $i = 1, 2, \ldots, 100$ help us to keep track of the points we lift and identify their neighbours. We shall start the algorithm by identifying the cell with the smallest area.

1. As the first step $r = 100$, the smallest area calculated at the site $i^{100} = 34$ and its area is $I_{100,34} = 0.00195$. The set of neighbours of x_{34} is found as

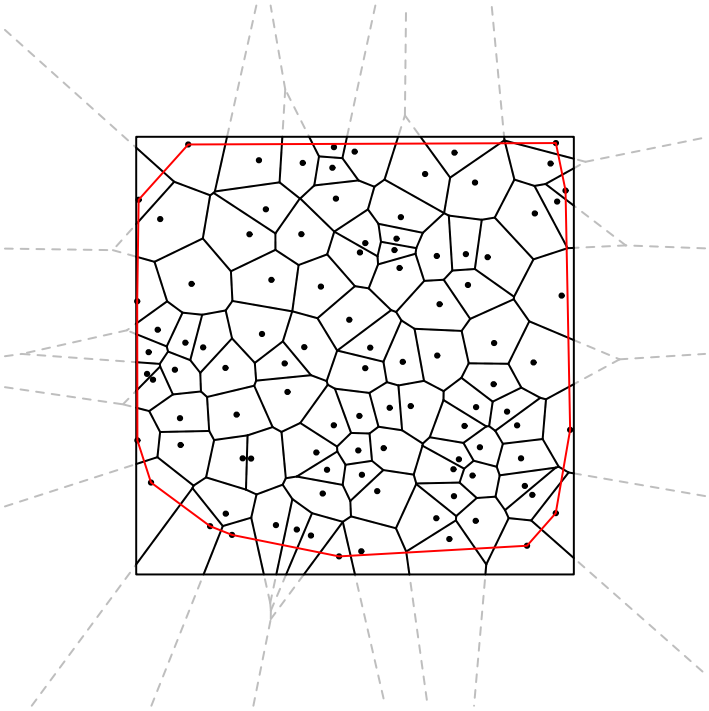


Figure 5.4: Voronoi tessellation of 100 uniform random points generated in a unit square. The black square is the unit square boundary and the red polygon is the convex hull of points. Gray dashed lines show the shapes of the polygons if no boundary was imposed.

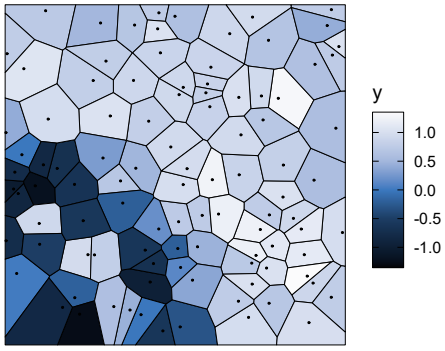


Figure 5.5: Voronoi tessellation of 100 uniform random points generated in a unit square. Cells are coloured based on the noisy function value at the locations.

$J_{100} = \{x_8, x_{28}, x_{99}\}$. Also, the observed noisy value calculated for x_{34} is $y_{34} = -1.04$, and $y_{J_{100}} = \{-0.869, -1.215, -0.765\}$ for its neighbours.

2. In order to calculate the detail coefficient d_{34} for the point x_{34} using equation (5.5), we need to calculate the vector of weights $\mathbf{a}$ first. After removing the point to be lifted at this stage, its area is shared by the neighbours and

each piece of the shared area allows us to calculate the weights taking the ratio over the entire area. Figure 5.6 shows an illustration of this process. We see a zoomed version of the Voronoi tessellation of the point x_{34} along with its neighbours x_8, x_{28} and x_{99}. After lifting point x_{34}, the Voronoi cells of the neighbours take the form of the solid lines. Dashed gray lines are the former lines of x_{34}. Note that all other cells will remain the same since the lifting only affect the neighbouring cells.

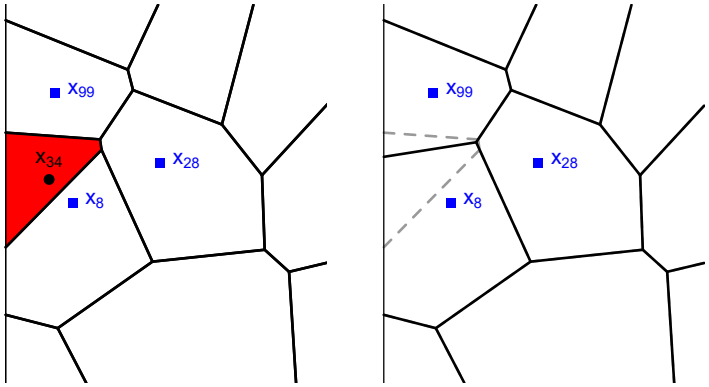


Figure 5.6: Zoomed in plot of Voronoi tessellation of the lifted point x_{34} and its neighbours x_8, x_{28} and x_{99} (left), and the partition of the V_{34} by the neighbours (right).

By equation (5.8), the weights are calculated as

$$a_8 = \frac{0.00154}{0.00195} = 0.788, \quad a_{28} = \frac{0.00003}{0.00195} = 0.002, \quad a_{99} = \frac{0.0004}{0.00195} = 0.210,$$

and using (5.5), the detail coefficient is

$$d_{34} = 1 - (0.788, 0.002, 0.210)^\top (-0.869, -1.215, -0.765) = -0.193.$$

3. Note that to update the function values of the neighbours, we need to calculate the weights b first. By equation (5.7),

$$b_8 = \frac{0.00195 \times 0.0648}{0.000098} = 0.129, \quad b_{28} = \frac{0.00195 \times 0.0637}{0.000098} = 0.127$$

$$b_{99} = \frac{0.00195 \times 0.0039}{0.000098} = 0.078,$$

hence we update the function values of neighbours using (5.6)

$$\begin{aligned} y^{\star}_{J_{100}} &= (-0.869, -1.215, -0.765) + (-0.193)(0.129, 0.127, 0.078) \\ &= (-0.894, -1.239, -0.780). \end{aligned}$$

4. Finally, x_{34} is removed at stage $r = 100$ and the same procedure is repeated at the next stage $r = 99$.

In the stage $r = 99$, $i^{99} = 49$, so we lift x_{49} whose function value is $y_{49} = 0.804$ and area is $I_{99,49} = 0.00275$. Now, $J_{99} = \{x_{15}, x_{30}, x_{67}, x_{80}, x_{83}\}$ are the neighbours of x_{49}. The detail coefficient at x_{49} is found to be $d_{49} = -0.048$. Also, updated function values for $x_{15}, x_{30}, x_{67}, x_{80}, x_{83}$ are calculated as

$$y^{\star}_{J99} = (0.852, 0.907, 0.834, 1.007, 0.910),$$

respectively. For the next stage $i^{98} = 7$, so the site x_7 will be lifted and the process will be repeated until $r = l + 1$.

Now we shall perform the full lifting transform using the `lift2D` function we created. Based on the same data locations and function values, the lifting transform is performed setting $l = 12$. In Figure 5.7, the progression of the forward transform of lifting is illustrated with snapshots of the updated function values of the remaining points in the intermediate steps. The top-left plots is the Voronoi tessellation of all points where the cells are coloured based on the observed noisy function values. The remaining plots show the updated function values of the non-lifted points at different stages of lifting $r = 80, 60, 50, 40, 30, 20, 13$.

The plots except the top-left one, are colored based on the updated function value where the mutual color scheme is given on the right end of the figure. The points appear on the plots are the points that remain in the data set. The function has activity near the bottom-left corner and is smooth otherwise. The initial data at the top-left plot is smoothed over the stages of the lifting. The Voronoi cells of the removed points join the areas of their neighbours hence the cell areas increase as the number of points decrease. Also, the updated function values of the removed points include averaging over the neighbours hence we obtain the smoothest pattern at the final stage.

We now show the detail coefficients d_i and estimated function values $\hat{f}_i$ in Figure 5.8 at the data points. The detail coefficients are calculated for each removed point x_i and the cells are colored based on the value of the coefficient. The color scheme is

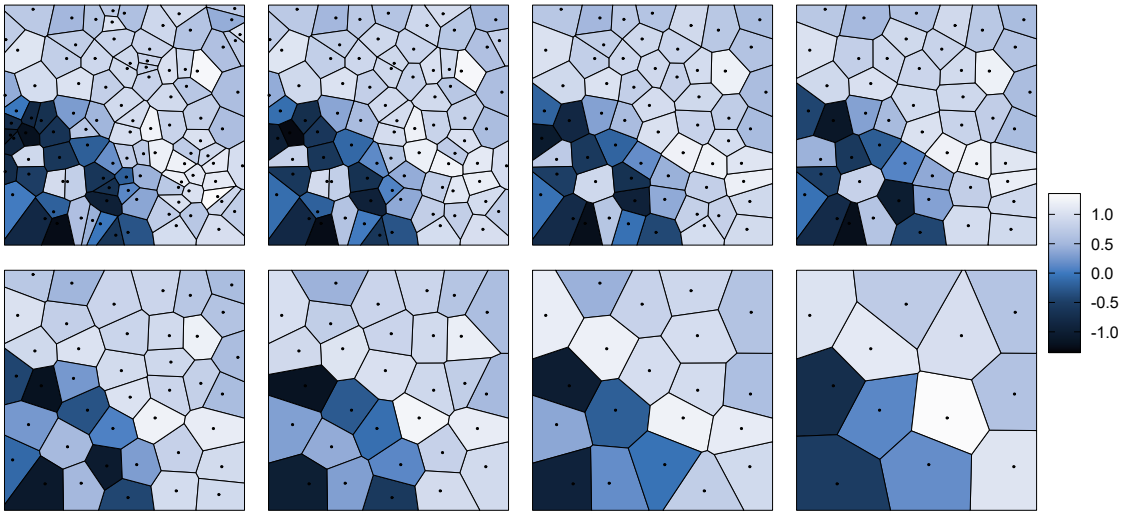


Figure 5.7: Progression of the lifting transform. Top-left plot is the Voronoi tessellation of all points where the cells are coloured based on the noisy function values. Remaining plots show the updated function values for the non-lifted points at different stages of the lifting transform.

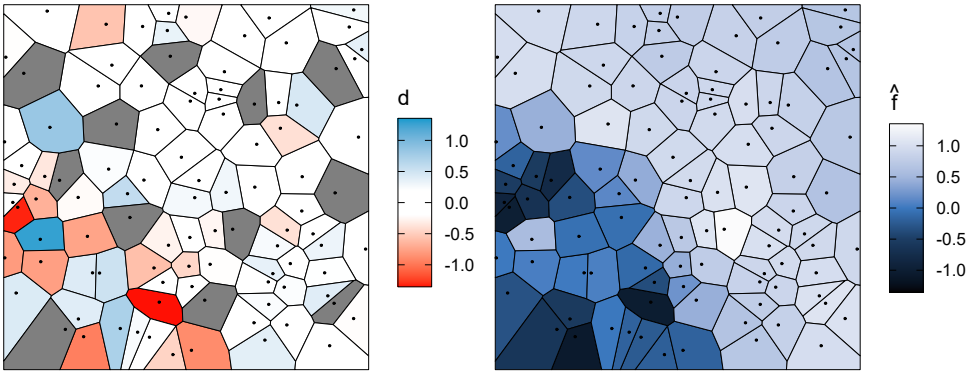


Figure 5.8: Detail coefficients calculated at each lifting stage (left), and the estimated function values for the points (right). Cell areas are coloured based on the detail coefficient or the estimated function value. Gray polygons on the left plot indicate the non-lifted points for which the detail coefficients are not calculated

given at the right side of both plots. Gray polygons signify points that are not lifted hence no detail coefficients are calculated. The negative coefficients are colored in red and the positive ones are in blue. The coefficients very close to zero are colored in white. We observe the majority of the detail coefficients that are close to zero are located in the less active parts of the function. Large detail coefficients (both positive and negative) occur for the points where the function has high activity. The inverse transform is performed on the thresholded detail coefficients to estimate the underlying function $\hat{f}_i$ at points x_i. We used the empirical Bayes threshold in this

example. The estimated function values do not contain the variations caused by the Gaussian noise as in the top-left plot in Figure 5.7 since the noise is separated from the y_i by thresholding the detail coefficients d_i. Hence, the inverse transform of the thresholded coefficients gives an estimate of the underlying true function that is denoised, and the active parts in the true function are not over-smoothed which is one of the main advantages of using lifting as a smoothing technique..

Chapter 6

Lifting results for homogeneous data

We report the lifting results for homogeneous simulated data in this chapter. The lifting scheme explained in Chapter 5 is used for function estimation using simulated data comprising two-dimensional tests functions with artificial noise. The test functions used in this thesis are explained in Section 6.1. One of the important aspects of the lifting scheme is a set of weights which are used in the calculation of the lifting coefficients and to decide on the lifting order. The Voronoi tessellation cell area-based weight methods we propose are described in Section 6.2. The simulated data is generated based on the design in Chapter 2 that considers points from the homogeneous Poisson process with $\rho = 200$ within a unit square. However, we generate independent replications of data sets each of which contain a set of test function values at a set of Poisson points. The design of the simulation is explained in Section 6.3. Finally, function estimation results are presented for each test function in Section 6.4 and conclusions are in Section 6.5.

6.1 Test functions

The function estimation performance of the lifting scheme is evaluated using two-dimensional test functions: Doppler, Heavisine, Blocks, Bumps and Maartenfunc that are shown in Figure 6.1. The test functions in (a)-(d) were first introduced by Donoho & Johnstone (1994) in one dimensional form, and Maartenfunc in (e) is designed to be used in lifting by Jansen *et al.* (2009). Two-dimensional analogues of these test functions are used in Nason *et al.* (2004). The formulae and `R` implementation of the test functions are provided in Appendix B.1. The true function

values $f(x)$ is obtained from the formulae of the functions given in Appendix B.1 and the noise is added artificially. Therefore, the estimated function values $\hat{f}_i$ are compared to the true function values f_i.

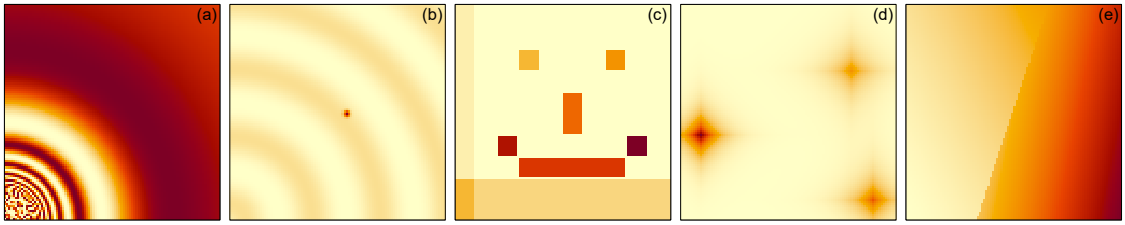


Figure 6.1: Test functions: (a) Doppler, (b) Heavisine, (c) Blocks, (d) Bumps, (e) Maartenfunc.

The test functions have different spatial characteristics. The Doppler in Figure 6.1 (a) is spherically symmetric function around the origin and has higher frequency oscillation closer to the origin and lower frequency activity otherwise. The Heavisine (b) has spherically symmetric with regular sinusoidal waves, and has a sharp spike around the centre at coordinates $(0.55, 0.5)$. The Blocks function (c) generates blocks that form a smiling face where the block heights take different integer values. The Bumps function (d) has three spikes where each spike has different heights and width. Lastly, the Maartenfunc (e) is a piecewise function that has a planar discontinuity at the intersection of the two planes.

6.2 Weight methods

The usage of the Voronoi cell area-based weights in the steps of lifting scheme is explained in Section 5.3.1. The basic idea is to use the cell area to decide on the lifting order as shown in (5.4), and in the calculations of the predict and update stages in (5.5) and (5.6) respectively. Also, the neighbourhood structure is determined using the Voronoi tessellation of data points as shown in Figure 5.1. Hence, the weights play a key role in the function estimation since they affect the calculations of the lifting coefficients that are then transformed to the estimated function values. Given that the statistical properties of cell area differ for the cells near the boundaries as demonstrated in Chapter 2 regardless of the boundary types we used, we expect to see differences in the estimated function values if we were to use different methods to calculate cell area. In this chapter, we will use various available options to calculate the cell area that correspond to *observed weights*, and the prediction of cell area as we devised in Chapter 3 which we call *adjusted weights*.

Two main groups of weight methods are considered. The *observed weights* contain the cell area calculated using unit square and convex hull boundaries. On the other hand, *adjusted weights* includes the predicted cell area using base and augmented models, and doubled edge cell area. The observed weights are the standard ones used in the literature. However, using the adjusted weights, particularly the prediction of cell area, is the novel approach we propose. We introduced this method in Chapter 3 for homogeneous data and expanded it in Chapter 4 for the regular and clustered data cases. Now we combine this method with the lifting scheme in the context of the weights. The full list of weight methods we consider and their explanations is:

i. *convex:* Cell area using the convex hull boundary.

ii. *unit:* Cell area using the unit square boundary.

iii. *double:* Edge cell area using the unit square boundary is doubled, and interior cells area is kept the same.

iv. *base:* The ensemble prediction of cell area is calculated using the base models from Chapter 3.

v. *augm.:* The ensemble prediction of cell area is calculated using the augmented models.

These five weight methods are used separately in the lifting transform in the calculation of detail coefficients which are thresholded and inverted later for function estimation. We aim to investigate whether the usage of different weight methods matters in function estimation, highlight which methods are accurate and robust for different test functions, and look at local performance such as in the places where discontinuities happen, edges, corners, etc. Whilst the weight methods from observed cell area are more rigid methods, adjusted weights, especially the base and augmented model predictions are novel approaches that also aim to advance the performances of the other methods.

6.3 Design of the simulation

The simulation follows these steps:

i) Generate a set of n homogeneous Poisson points where $n \sim Po(200)$ in the unit square sampling region $\Omega = [0, 1]^2$.

ii) At each point x_i for $i = 1, \ldots, n$, calculate $y_i = f(x_i) + \epsilon_i$ using the selected test function. The ϵ_i are iid Gaussian noise assuming $\epsilon \sim N(0, \sigma^2)$ and the variance is determined from $\sigma = \sigma_t / z$ where σ_t is the standard deviation of the true function values $f(x_i)$ and z is the root signal to noise ratio to define the magnitude of the noise, which is taken as $z = 3$ in this experiment.

iii) Perform the forward lifting transform using the selected weight method to determine the vector of detail coefficients $\mathbf{d} = \mathbf{L}\mathbf{f}^{\star}$.

iv) Shrink the detail coefficients using the hard, soft, and empirical Bayes thresholding methods to obtain $\hat{d}_i = t(d_i)$.

v) Perform the inverse transform $\hat{\mathbf{y}} = \mathbf{L}^{-1}\hat{\mathbf{d}}$ on the vector of thresholded coefficients $\hat{\mathbf{d}}$ to obtain the vector of function estimates $\hat{\mathbf{y}}$ (denoised values).

vi) Record the locations x_i and estimates $\hat{y}_i$ for all points, and repeat the process for $r = 250$ independent realizations.

6.4 Results for simulated homogeneous data

The simulation study generates 250 data sets and each data set has $\{n_j\}_{j=1}^{250} \sim Po(200)$ locations. In total, there are expected to be $\rho \times 250 \approx 5 \times 10^4$ points generated in the unit square. Data sets are independent from each other. However, we do not change the 250 data sets for different configurations such as the test functions, weight methods, and the thresholding rule. Therefore, function estimation is made using the same data locations for different techniques that makes the function estimation results comparable based on the weight methods.

The way of presentation and discussion of the results is important. We combine the results from all data sets. The accuracy of the function estimation may be discussed for the unit square globally. However, we are actually interested in the local details as well. Therefore, different parts such as the interior and edge regions, and diagonal, vertical and horizontal transects, especially near the boundaries, may be used. We focus on the edges particularly because different weight methods usually have different values for edge cells and minor or no differences for interior cells. To define the transects and for the summary results, the entire unit square is divided into equal-sized square bins as shown in Figure 6.2. The zoomed-in plot of the bins shows how the points are scattered into each bin. Checking the data reveals that there are no bins missing a data point.

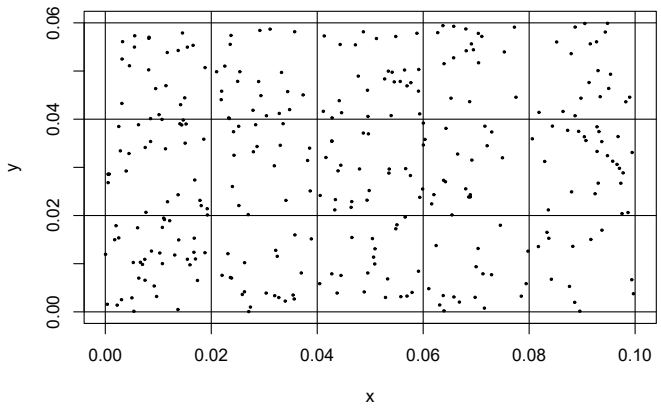


Figure 6.2: Zoomed in bottom left corner of the unit square divided into a 50×50 grid of square bins, showing how the points fall into the first few.

In the remainder of this chapter, the function estimation using five different weight methods is discussed for each test function separately in the global, local, and transect-based parts of the region. The parts where the differences between weight methods occur are highlighted and the advantages of using particular weight methods are discussed.

6.4.1 Doppler

Starting with the Doppler test function, a pairwise comparison of different weight methods is the first step. This can be checked by looking at the global mean squared error (MSE) between the estimates and the true function values based on different methods. The formula to calculate the global MSE for a data set is

$$\frac{1}{n_j}\sum_{i=1}^{n_j}(f_i - \hat{y}_i)^2 \tag{6.1}$$

where n_j is the size of the $j-$th data set and f_i and $\hat{y}_i$ are the true and estimated function values using the data locations in the $j-$th data set respectively. Even though global inference is important, it may not be very informative since the local details are hidden behind the global inference. What is more interesting and valuable is to check the MSE at the transects and at the locations where the test function show high activity, spikes, and discontinuities.

The same data sets are used for function estimation when we alter the weight method. Hence the function estimation is made for the same data locations but using different approaches. This allows us to see the function estimation at the same locations using different configurations of weight methods. Therefore, it is possible

to have a pairwise comparisons of the weight methods for each test function. We rely on the mean squared error when comparing the methods, which can be global MSE, or MSE at different parts such as the edge, or transects. If the global MSE is not being used, we use a version of equation in (6.1) by sampling the locations in the defined area or transect which we are interested in.

We first look at whether there are significant differences between the weight methods in terms of function estimation. If we look at Figure 6.2 again, there are numerous data locations in each bin. The pairwise comparison of the weight methods is going to be made for the data locations in each bin separately and the results are going to be transformed to plots and the tables.

We are interested in the comparison at local details in addition to the standard global results. The MSE results using different methods and transects are given in Figure 6.3. Some vertical and horizontal transects v_1, v_2, h_1, h_2, and diagonal transect are selected over the sampling region, and shown in the top-left image on original Doppler test function; different transects will be used for other test functions. These transects are mainly selected in the regions that we are interested in. The transects are denoted as the v_1, v_2, h_1 and h_2 in Figure 6.3. The $v_1, v_2, \ldots, v_{max}$, and $h_1, h_2, \ldots, h_{max}$ are always the vertical and horizontal edge transects and there are no other transects between these transects and the boundary for the other test functions we are going to consider.

The MSE is calculated based on the estimated function values for the data locations in each bin that the transects pass over. There are 50 bins in each transect, and the MSE is calculated for the data locations in each bin separately. Then the 50 MSE values are shown in the line plots in Figure 6.3 for each transect separately. The line colours signify the weight method which are labeled in the bottom right plot. In each line plot from top-centre to the bottom-right, we aim to check the differences between the MSE values obtained from each weight method based on different transects.

There are parts where the results for different methods have obvious differences and similarities. The differences mainly exist near the edges at v_1 and h_1 where the oscillation of the function has higher activity. At v_1 and h_1, the green line (augmented method) generally has the smallest MSE and the black line (convex hull boundary) has the highest. The pattern is similar for both transects since the Doppler has symmetric properties. On the other hand, the MSE is very small at the v_2 and h_2 transects where the function show lower activity. The performances of the different weight methods are both similar and satisfactory at v_2 and h_2. The bins

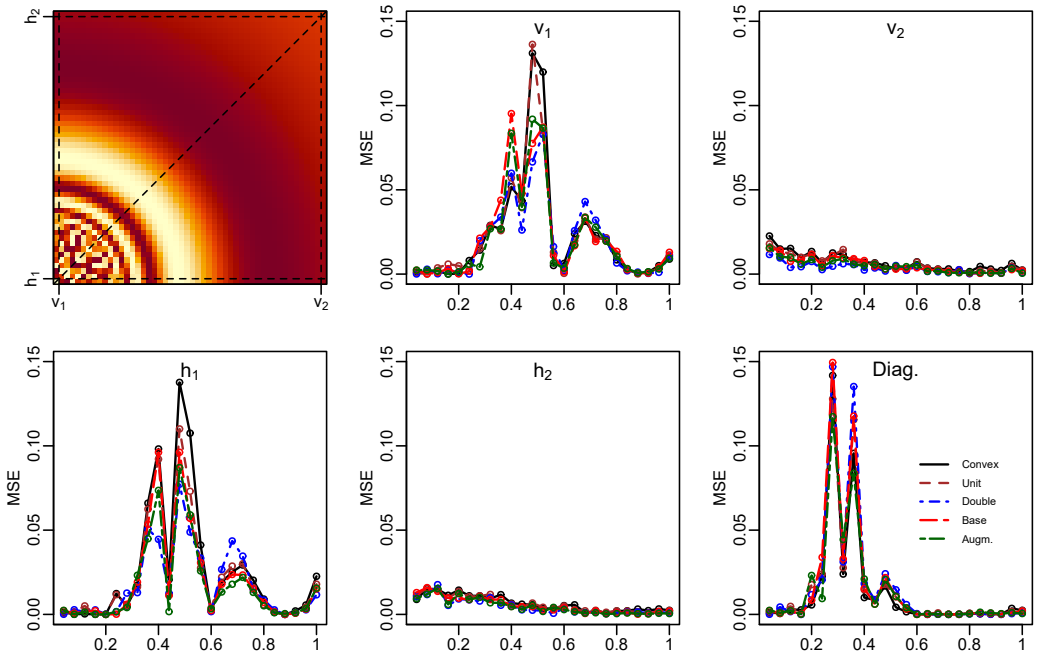


Figure 6.3: The original Doppler test function (top left). The lines show the mean squared error calculated at transect bins using different weight weight methods. Transects are shown with dashed lines on the test function.

located at the diagonal transect mostly take place far from boundaries hence the edge effect is minimal. Here, the methods show similar patterns and some variation at different parts which is due to the small sample size in individual bins.

The numerical results of the MSE in different spatial regions and transects are shown in Table 6.1. The results are separated into MSE calculated globally, interior and outer region, and the transects. The entire sampling region is $\Omega = [0, 1]^2$, and the interior region is defined as $\Omega_{in} = [0.15, 0.85]^2$ and the outer region (edge) is the $\Omega_{ed} = \Omega'_{in}$. The smallest MSE in each row is coloured in blue, and multiple values are highlighted if they are equal or very close. The augmented method outperforms the others in most cases especially close to the boundaries and where there is high activity in the function. The results in the table validates the conclusions from Figure 6.3.

6.4.2 Heavisine

Results for the Heavisine test function are presented and discussed in this section. We check the accuracy of the estimations based on the MSE in the line plots and summarize the numerical values in the table that gives a better understanding of the best and worst methods.

	Convex	Unit	Double	Base	Augm.
Ω	0.046	0.046	0.047	0.045	0.044
Ω_{in}	0.039	0.041	0.044	0.041	0.040
Ω_{ed}	0.054	0.051	0.050	0.050	0.049
v_1	0.111	0.102	0.093	0.094	0.087
v_2	0.017	0.014	0.013	0.013	0.013
h_1	0.100	0.089	0.080	0.088	0.083
h_2	0.016	0.014	0.014	0.013	0.013
D	0.070	0.069	0.070	0.067	0.062

Table 6.1: Results for the Doppler test function. Table shows the global Ω, interior Ω_{in} and edge Ω_{ed} MSE, and MSE at vertical v_1, v_2, horizontal h_1, h_2, and diagonal D transects. The smallest MSE is highlighted in blue.

The MSE values calculated along the transects are shown in Figure 6.4 using the transects v_1, v_2, h_1, h_2, and the diagonal transect. Since the v_1, h_1 and the v_2, h_2 are the transects where the function values are symmetric, the MSE calculated for these pairs of transects are very similar. The line plots show that the green line (augmented method) is having smaller MSE in general and the black line (convex hull method) is the highest hence we can conclude the overall performance of the augmented methods is satisfactory and the worst method is the convex hull boundary as in the Doppler case. In Table 6.2, the augmented model has smallest MSE globally and at the edges, and at edge transects v_1, h_1.

	Convex	Unit	Double	Base	Augm.
Ω	0.334	0.304	0.309	0.297	0.293
Ω_{in}	0.300	0.307	0.336	0.303	0.307
Ω_{ed}	0.367	0.301	0.282	0.291	0.279
v_1	0.438	0.365	0.317	0.336	0.307
v_2	0.639	0.433	0.345	0.399	0.351
h_1	0.437	0.379	0.332	0.342	0.306
h_2	0.628	0.434	0.323	0.390	0.332
D	0.362	0.282	0.308	0.285	0.294

Table 6.2: Results for the Heavisine test function. Table shows the global Ω, interior Ω_{in} and edge Ω_{ed} MSE, and MSE at vertical v_1, v_2, horizontal h_1, h_2, and diagonal D transects. The smallest MSE is highlighted in blue.

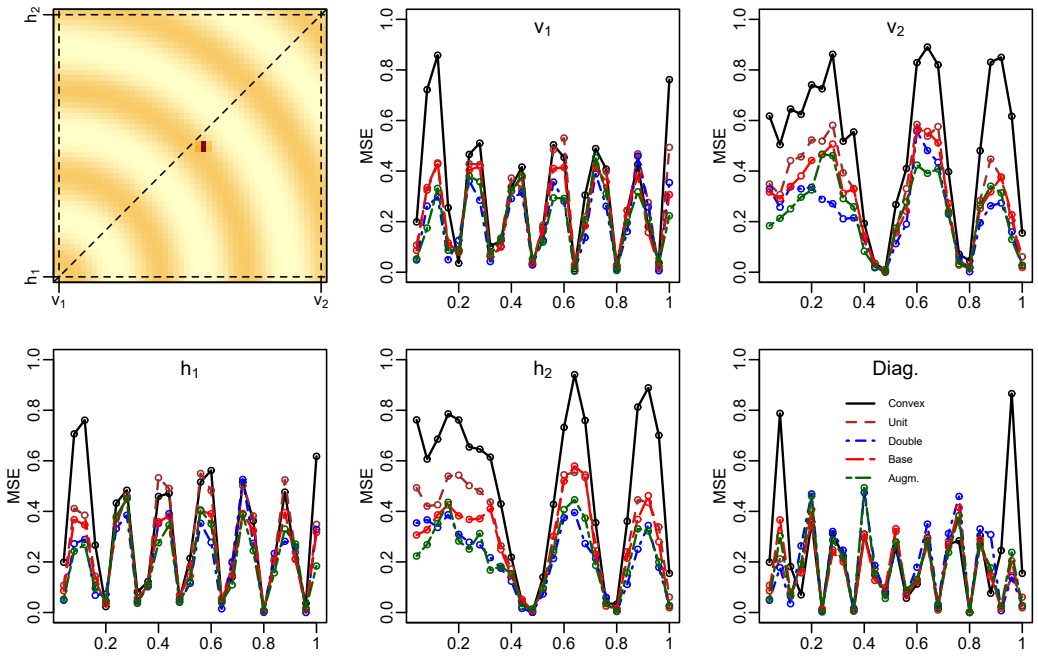


Figure 6.4: The original Heavisine test function (top left). The lines show the mean squared error calculated at transect bins using different weight weight methods. Transects are shown with dashed lines on the test function.

6.4.3 Blocks

The Blocks test function has different characteristics compared to the previous test functions with its block spikes in different shapes and heights. The spatial pattern of the MSE in Figure 6.5 is not very clear to visually distinguish, and a generalization of the overall quality of the best method is difficult. The MSE values have spikes where the function has sharp changes. Otherwise, the MSE for all methods are very close to zero at the parts where the function is flat. The numerical values are shown in Table 6.3 where the augmented method has the smallest MSE at most of transects. The base and augmented methods perform similarly in some transects, and it is important to highlight their comparability for the edge MSE.

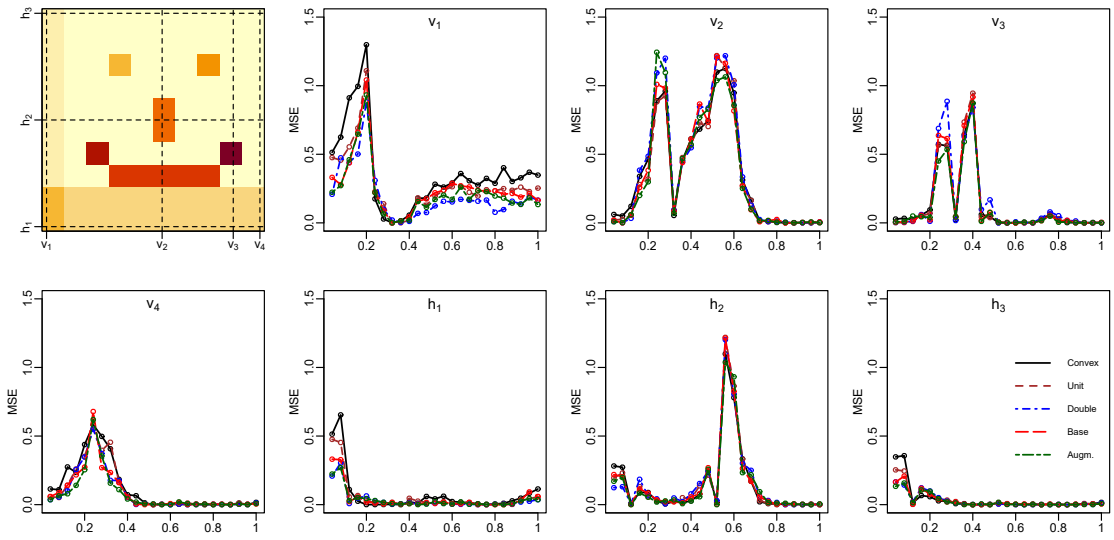


Figure 6.5: The original Blocks test function (top left). The lines show the mean squared error calculated at transect bins using different weight weight methods. Transects are shown with dashed lines on the test function.

	Convex	Unit	Double	Base	Augm.
Ω	0.531	0.513	0.538	0.511	0.518
Ω_{in}	0.759	0.764	0.818	0.767	0.780
Ω_{ed}	0.312	0.271	0.270	0.265	0.266
v_1	0.545	0.435	0.351	0.403	0.380
v_2	0.800	0.801	0.858	0.800	0.787
v_3	0.445	0.438	0.465	0.434	0.399
v_4	0.320	0.260	0.249	0.255	0.263
h_1	0.257	0.190	0.183	0.188	0.179
h_2	0.488	0.482	0.501	0.492	0.479
h_3	0.133	0.116	0.120	0.111	0.113

Table 6.3: Results for the Blocks test function. Table shows the global Ω, interior Ω_{in} and edge Ω_{ed} MSE, and MSE at vertical $v_1, \ldots, v_4$, and horizontal h_1, h_2, h_3 transects. The smallest MSE is highlighted in blue.

6.4.4 Bumps

The results for the Bumps test function show that the noticeable differences between the weight methods occur near the boundaries. The transects $v_1, v_2, v_3, v_4, v_5, h_1$, and h_2 are selected to see the patterns of MSE for different weight methods in Figure 6.6. The are edge transects, and transects where the function has spikes. We also selected v_3 where the function is flat as a control case where all methods are expected to work equally well.

Figure 6.6 shows the noticeable differences between weight methods on v_1, v_2, v_4, v_5 and the non flat parts of h_1 and h_2 that are all transects close to the edges. In the vertical transects v_1 and v_5, the green line is the closest one to zero in most bins hence the augmented method outperforms the other methods. The usage of unit square and the convex hull methods give the worst estimates on these transects. The differences are less obvious for v_2 and v_4 which are relatively close to the boundary, however, the green line of the augmented method seems to have the smallest MSE values which is confirmed in Table 6.4. The augmented method also performs well at the horizontal transects h_1 and h_2 where the differences are detected when the function is not flat. The high MSE values are calculated in regions where the function has spikes, which is a general issue in all test functions.

Table 6.4 clarifies the good performance of the augmented method in all cases. Even though the augmented method MSE is not the smallest in some cases such as the Ω_{in} and v_3, it is very close to the smallest value found from another weight method. The results for the Bumps test function clearly suggest the usage of augmented method. The exceptional scenarios are for the interior region and v_3 transect which the function is flat. The MSE values for these two cases are very close between weight methods but the augmented method has the best performance in all important cases.

6.4.5 Maartenfunc

The last test function used in the simulations is the Maartenfunc. The differences between the weight weight methods are not very obvious from the MSE line plots in Figure 6.7. The most visible differences between weight methods are in v_3 which is an edge transect and the function is close to be linear. The edge cell doubling method seem to give the smallest MSE values along the transect and the MSE between the best and worst method is larger near the corners as the convex hull

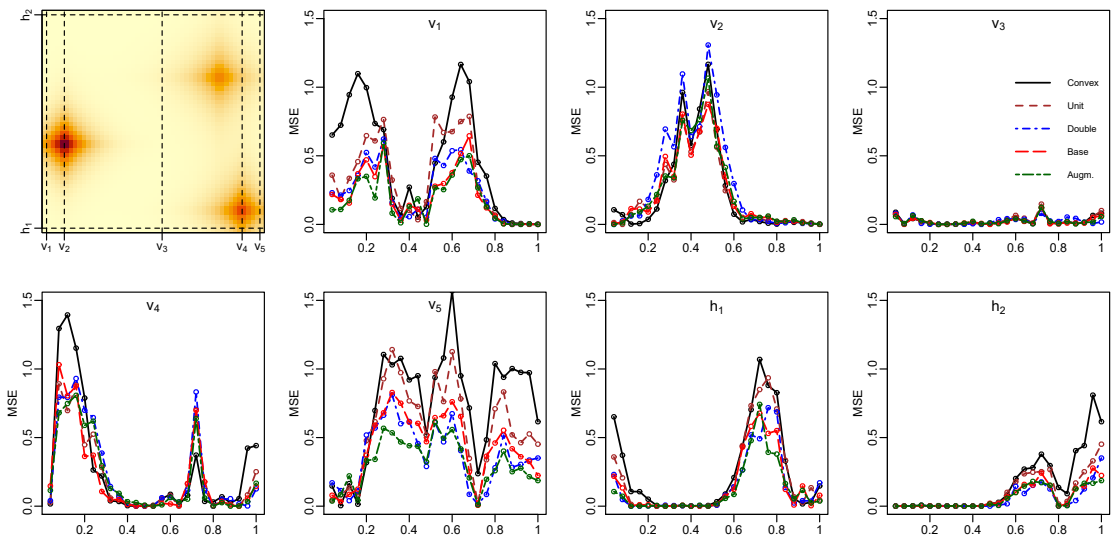


Figure 6.6: The original Bumps test function (top left). The lines show the mean squared error calculated at transect bins using different weight weight methods. Transects are shown with dashed lines on the test function.

	Convex	Unit	Double	Base	Augm.
Ω	0.427	0.376	0.377	0.358	0.349
Ω_{in}	0.295	0.299	0.328	0.293	0.295
Ω_{ed}	0.553	0.449	0.424	0.420	0.402
v_1	0.969	0.734	0.645	0.592	0.548
v_2	0.589	0.468	0.499	0.485	0.468
v_3	0.099	0.098	0.100	0.093	0.100
v_4	0.701	0.589	0.588	0.586	0.576
v_5	1.265	0.974	0.801	0.837	0.721
h_1	0.607	0.495	0.429	0.429	0.396
h_2	0.302	0.213	0.192	0.179	0.177

Table 6.4: Results for the Bumps test function. Table shows the global Ω, interior Ω_{in} and edge Ω_{ed} MSE, and MSE at vertical $v_1, \ldots, v_5$, and horizontal h_1, h_2 transects. The smallest MSE is highlighted in blue.

method is the worst. The differences between weight methods are more apparent at the other edge transects v_1, h_1, h_3 for the parts $[0, 0.2]$ and $[0.8, 1]$.

The MSE values in Table 6.5 are very close to each other especially when the double, base, and augmented methods are used. They have compatible performances in the edge region and it is appropriate to use the adjusted weights. However, since the doubling is a rigid process compared to the prediction of cell area using base and augmented models, the usage of augmented method would be more appropriate considering its overall performance in the other test functions.

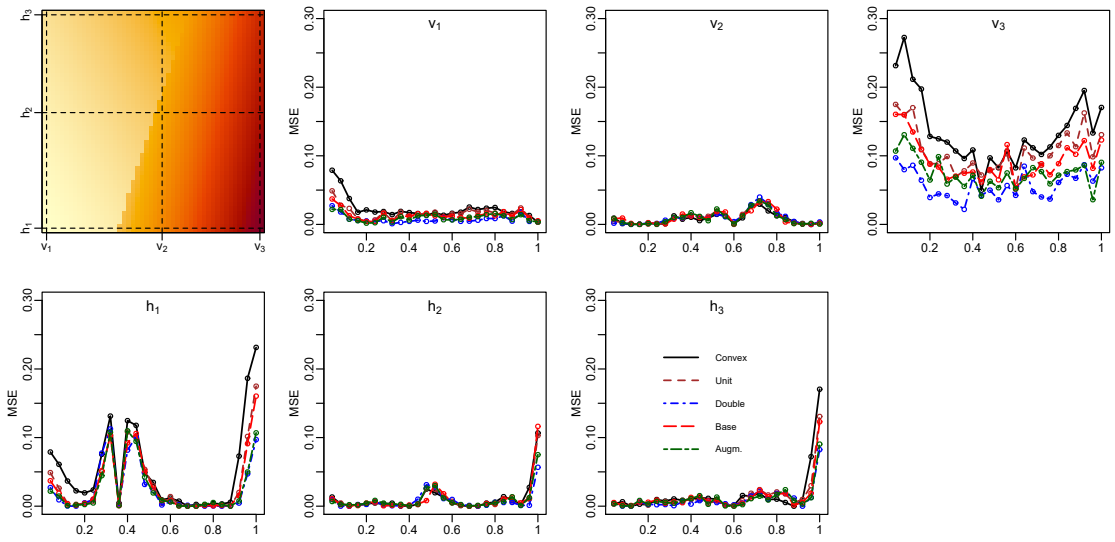


Figure 6.7: The original Maartenfunc test function (top left). The lines show the mean squared error calculated at transect bins using different weight weight methods. Transects are shown with dashed lines on the test function.

	Convex	Unit	Double	Base	Augm.
Ω	0.055	0.049	0.045	0.047	0.047
Ω_{in}	0.040	0.040	0.038	0.039	0.039
Ω_{ed}	0.070	0.058	0.053	0.055	0.054
v_1	0.057	0.047	0.044	0.041	0.040
v_2	0.053	0.050	0.050	0.049	0.052
v_3	0.206	0.162	0.116	0.146	0.134
h_1	0.106	0.079	0.077	0.077	0.074
h_2	0.046	0.042	0.043	0.043	0.044
h_3	0.054	0.046	0.049	0.044	0.043

Table 6.5: Results for the Maartenfunc test function. Table shows the global Ω, interior Ω_{in} and edge Ω_{ed} MSE, and MSE at vertical v_1, v_3, v_3, and horizontal h_1, h_2, h_3 transects. The smallest MSE is highlighted in blue.

6.5 Conclusions

This chapter presented and discussed the function estimation results using the lifting scheme for homogeneous data, giving emphasis on what happens when different weight methods are used in the lifting. The simulation setting considers various important configurations such as the usage of test functions that have different spatial characteristics, and weight methods to evaluate the performances of different approaches in function estimation. The MSE values attained at different parts and transects of the region highlight the differences between the usage of different weight methods.

The results in this chapter highlight two important aspects of function estimation using lifting. First, the significant differences between weight methods are demonstrated, then the best method to estimate the function is suggested. More importantly, we focused on local details such as the parts that are close to the boundaries, and functions having discontinuities, spikes, etc. Throughout the discussion of the results for each test function, the augmented method gave a better performance compared to the other weight methods. It achieved more accurate function estimation especially at the edge transects in which we aimed to improve the accuracy of the function estimation obtained from standard observed weights.

Earlier lifting research used traditional weight methods and emphasized the issues that may occur for the data locations near the boundaries. Our work in this thesis primarily suggests ways to eliminate, or at least reduce the boundary effects in function estimation. Simulation results show that our proposed weight method, which uses the predicted cell area from augmented models, is the favourable option. Therefore, a general use of the augmented weight method is suggested in lifting for homogeneous data.

Chapter 7

Lifting results for regular, clustered and real data examples

7.1 Lifting for regular and clustered data

In this chapter, the lifting study is extended to the case of regular and clustered data cases. An introduction to the regular and clustered points was given in Section 4.2, where we relied on the saturation process introduced by Geyer (1999) to create regular and clustered points with different types of irregularity such as clustering and inhibition. We rely on the same point process to simulate regular and clustered points in this chapter. The lifting scheme is a multiscale method used to analyze irregularly spaced data, and it is important to see its capability in dealing with different types of irregularity. In this chapter, we not only investigate the performance of the lifting scheme in function estimation for regular and clustered points, we also consider the extreme cases of highly regular and clustered point patterns and check how the lifting scheme performs.

This chapter presents and discusses the lifting results for regular and clustered data from simulations, and real data examples. The design of the simulation is similar to Section 6.3, however, there are various point pattern cases rather than a single homogeneous Poisson point process case. The point patterns are determined based on different values of the parameter γ of the process. Essentially, the same parameter values $\gamma = 0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 2, 3$ are used to generate the data sets as in Figure 4.1 in Section 4.2. For each value of γ, a set of n points, where $n \sim Po(200)$, are generated and function estimation is conducted for the test functions from Section 6.1. This process is carried out for 250 replicates of each

point pattern altering γ. Therefore we have 250×9 data sets each of which has a size $\{n_j\}_{j=1}^{250}$. Function estimation results attained from lifting scheme with various weight methods are compared to kriging estimates.

The main purpose of this chapter is to investigate how the regularity and clustering in the data affects the function estimation. Also, a comprehensive analysis is performed to see the effects of different weight methods. In addition to the five weight methods used in Chapter 6, we introduce another version of the base and augmented model prediction methods. Area prediction for regular and clustered points is done using the scaled covariates based on the estimated local intensities as explained in Chapter 4 that draw attention to the better performance of the $B^\star$ and $Ag^\star$ models over B and Ag models in the sense of area prediction. The weight methods with $\star$ superscript are the versions of B and Ag that are designed for regular and clustered data cases, and we use both versions to have a comparison in this chapter. Application of the lifting for the regular and clustered points does not have methodological differences to the homogeneous case, so the same lifting steps for forward transform, thresholding the detail coefficients, and the inverse transform are followed.

This chapter also considers the application of lifting to the real data sets; `spruces`, Barro Colorado Island (BCI), `waka`, `finpines`, and `longleaf` which are explained in Section 4.3.2. We examine how lifting works for real data locations and measurements at the locations that show examples of regular, clustered and homogeneous patterns. These data sets are especially chosen for the purpose of having examples of homogeneous and regular and clustered real data examples. The lifting results for simulated and real data sets are presented and discussed in Sections 7.2 and 7.4 respectively. We also compared results for function estimation using lifting and kriging in Section 7.3, using the weight method that gave the best results in lifting for simulated regular and clustered data.

7.2 Results for simulated data

The lifting results for the regular and clustered data cases from all the test functions are presented in Figures 7.1 - 7.5. The number of cases we investigate is enormous, hence discussion of the numerical results is not very practical. Tables C.1 - C.5 in Appendix C show the MSE values for the function estimation for different configurations of point patterns, weight methods, and test functions at different parts and

transects of the sampling region. Due to the complex structure of the tables, it is more effective to inspect the plots.

7.2.1 Doppler

We start presenting lifting results from the simulations based on the Doppler test function. We selected the same vertical, horizontal and diagonal transects as in the previous chapter to compare the weight methods at different point pattern cases. The coloured points in the plots in Figure 7.1 show the MSE value corresponding to the different weight methods, and the values $i = 1, 2, \ldots, 9$ in the $x-$axis correspond to the index value of parameter $\{\gamma_i\}_{i=1}^{9} = 0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 2, 3$. In each plot, the results are given for global, interior and edge of the region, and at different transects of the region. Note that the range of the $y-$axis is different for each plot. The MSE values shown with points in different shapes and colour are the overall MSE for the associated transect, global, interior, or edge points. Numerical results are presented in Tables C.1 - C.5 in Appendix C.

The top-left plot in Figure 7.1 shows the global MSE for the Doppler test function. Whilst the points are very close for the regular point pattern and even overlap, the differences are more apparent for the homogeneous and clustered points as γ increases to 3. For the interior points, the MSE is smaller than the global MSE values. It is more interesting to analyze the edge points and the points located on the edge transects as the differences between the methods are clearer and have some pattern. For the ease of interpretation, convex hull, unit square and doubling are shown in red, base and augmented methods in blue, and $\star$ models in black. The weight methods show differences in terms of the MSE values for edge points. The smallest MSE is achieved with the solid black triangle for most of the regular and clustered point patterns which is the augmented weight method $Ag^{\star}$. The solid circles of the $B^{\star}$ method are either very similar or better in occasional cases that shows the robustness of the usage of local intensity methods in the Doppler example. The weight methods using the observed cell area generally give the worst estimation results.

The edge transects v_1 and h_1 are the transects where the function has spherically symmetric properties, hence conclusions can be made jointly. A satisfactory performance of the $Ag^{\star}$ method exists in v_1 and h_1 for the regular and clustered points. It is important to note that the smallest MSE is found using the blue triangle at $\gamma_5 = 1$ which indicates the homogeneity of the points. It is also the same for the

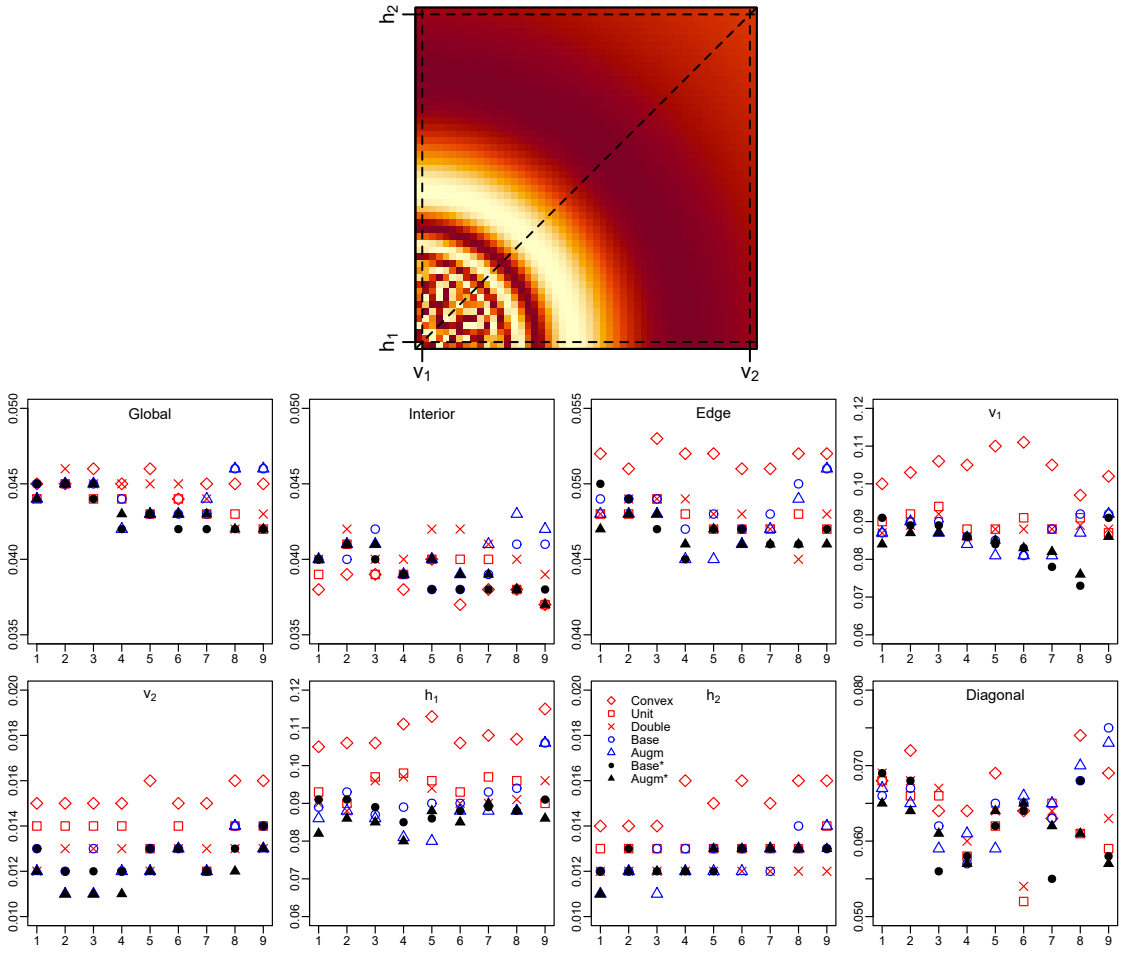


Figure 7.1: Lifting MSE results for Doppler at different parts and the transects. $x-$axis shows the index of γ_i for $i = 1, ..., 9$, and $y-$axis is the MSE which varies for each plot.

plot titled with 'edge'. This means the Ag method that is created for the homogeneous data has better performance than methods that uses the local intensities. It is pertinent because the B and Ag models are trained to predict the cell area for homogeneous points but the $B^\star$ and $Ag^\star$ models scale the covariates based on the estimated local intensity which are designed to be used for regular and clustered data cases.

Revisiting the 'global, edge, interior' plots, the pattern of the MSE values for adjusted weights from γ_7 to γ_9 deviate from each other. The solid points remain the smallest, however the non-solid points increase in MSE and perform almost as badly as the convex hull method. Although the performances of the weight methods shown with solid and open points were similar for regular point patterns, they are extremely different for the highly clustered points. Considering the regular point patterns, where the cells would have similar sizes, and the estimated local intensity

$\hat{\rho}_i$ at the points is not too different than the global point intensity ρ_0 as can be seen in Figure 4.2. Therefore, in these cases the scaling procedure of the $B^\star$ and $Ag^\star$ models will have minimal changes on the covariates since the scaling factor is $\hat{\rho}_i/\rho_0 \approx 1$. Hence the (B, Ag), and $(B^\star, Ag^\star)$ methods obtain similar function estimation results for the regular point patterns. However, the usage of estimated local intensity $\hat{\rho}_i$ become more important for the clustered points because the estimated local intensity $\hat{\rho}_i$ can have a larger deviation from the global intensity ρ_0. Therefore, the new methods, especially $Ag^\star$, has better performance for the clustered data cases than the unscaled methods (B, Ag).

7.2.2 Heavisine

In this section, lifting results for the Heavisine function are presented. Figure 7.2 shows the MSE values for different weight methods. The function estimates at the edge region give smaller MSE using the proposed weight methods $B, Ag, B^\star$, and $Ag^\star$ compared to the weight methods from observed cell area. The solid and open triangles and circles give very similar MSE for regular points that indicates the similarity of the $B, Ag, B^\star$, and $Ag^\star$ methods. However, the solid points of $B^\star$ and $Ag^\star$ methods persistently perform better for highly clustered points, and the open points of B and Ag start giving higher MSE after γ_7. For the interior points, the weight methods generally show compatible results except the highly regular and clustered cases at $\gamma = 1$ and $\gamma = 9$.

The results at the separate edge transects give similar conclusions. The augmented method $Ag^\star$ gives the smallest MSE for all point pattern types at transects v_1, v_2, h_1 and h_2. The MSE values for all methods decrease from γ_1 to γ_9 that points out the Heavisine function can be better estimated using the clustered points, and highly clustered point pattern types give the smallest MSE. The function has sinusoidal waves with a sharp spike around the centre. The neighbourhood structure has a major impact on the prediction of the function values in the prediction step of lifting. The function value at a data location is predicted as the weighted average of the function values of its neighbours. If the points have a regular pattern, then the interesting features of the test function may not be captured properly by the more distant neighbours, especially in functions like Heavisine. The function value at a selected point may be very different from its neighbours.

To make the example more specific, consider a point x_1 located on the spike, and it has several neighbours say x_2, x_3, x_4 and x_5 with a reasonable distance due to a regular point pattern. If the point x_1 on the spike is selected to be lifted, then

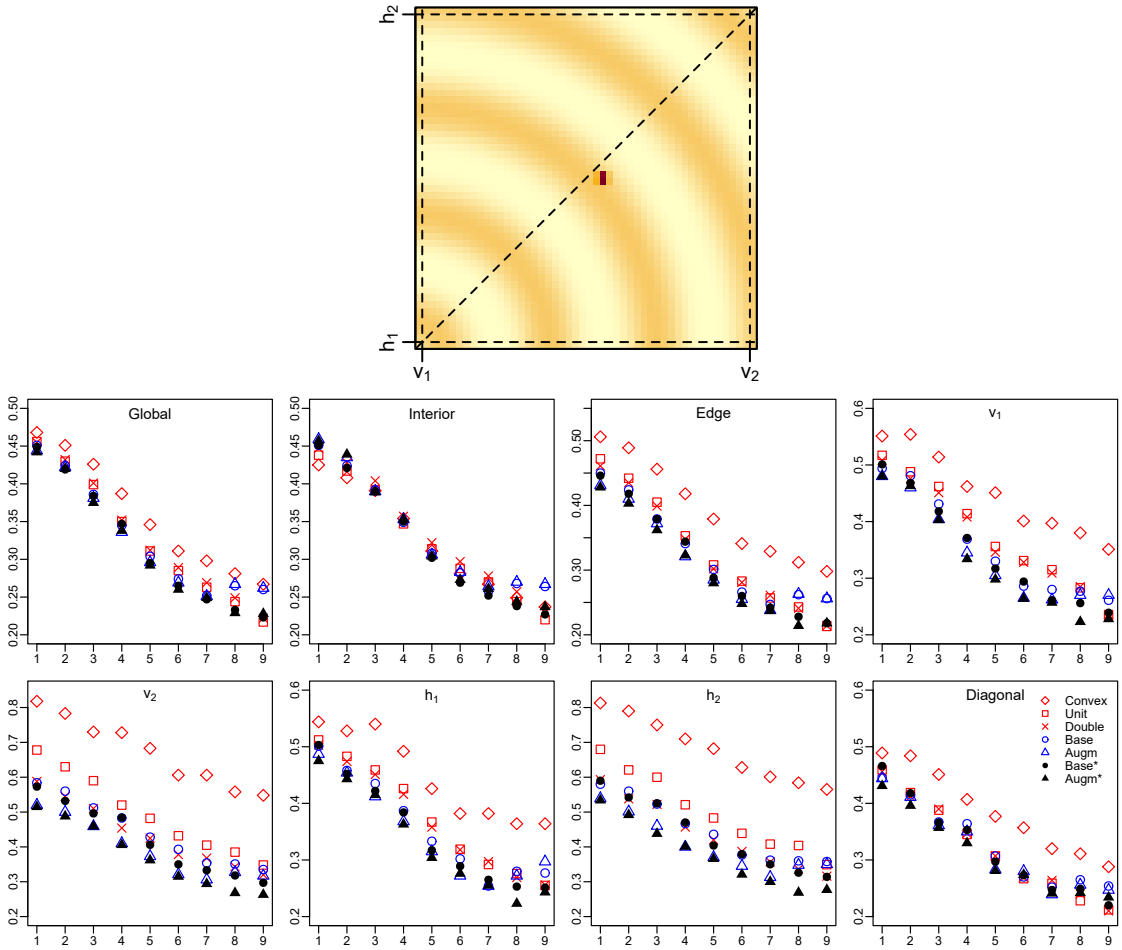


Figure 7.2: Lifting MSE results for Heavisine at different parts and the transects. $x-$axis shows the index of γ_i for $i = 1, ..., 9$, and $y-$axis is the MSE which varies for each plot.

its function value is predicted based on the values of its neighbours. Since its neighbours' function values range between the highest and lowest values of the sinusoidal waves, hence the predicted function value $\hat{f}(x_1)$ for x_1 will be affected by the discrepant values of the neighbours. However, if the points are clustered which means the neighbours are likely to be closer (at least most of them), then the neighbours have more relevant information about the function value to be predicted at x_1. This is likely to happen for the functions that short range irregularities like Heavisine.

7.2.3 Blocks

The Blocks test function has different types of sharp changes than those seen in the Doppler and Heavisine functions. The results are shown in Figure 7.3. The global results show little difference between methods, but differences between weight

methods are noticeable at the edge region and separate transects. The global MSE for all weight methods have a decreasing trend from γ_1 to γ_9 as observed for the Heavisine. The Blocks and Heavisine test functions have characteristics in common in terms of the spikes in Heavisine and the discontinuities in Blocks. However, the discontinuities of the Blocks function are rectangular prisms in different shapes and heights.

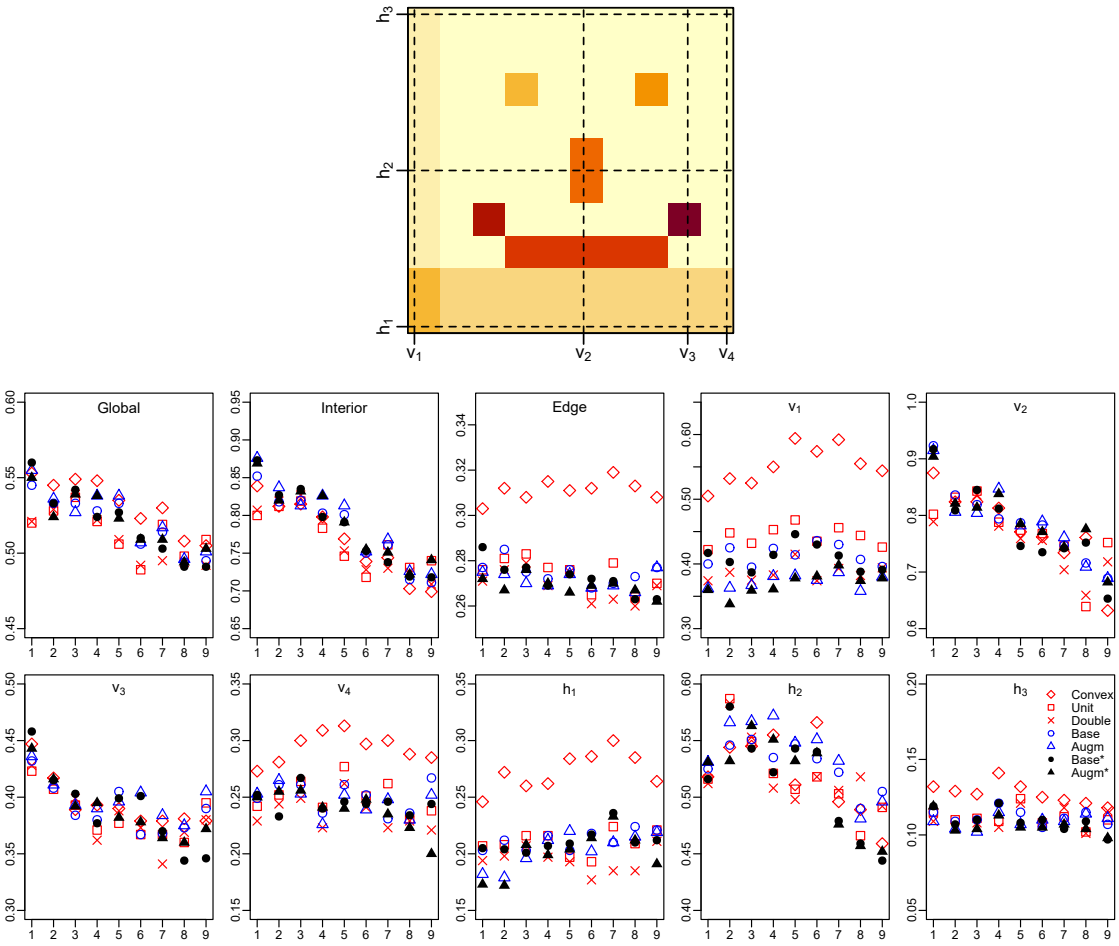


Figure 7.3: Lifting MSE results for Blocks at different parts and the transects. $x-$axis shows the index of γ_i for $i = 1, ..., 9$, and $y-$axis is the MSE which varies for each plot.

The number of transects we selected is higher for Blocks since this function also has interesting features at both the edge and intermediate transects. In the edge plot (top-centre) in Figure 7.3, the weight methods show similar results except for the convex hull. Even though the smallest MSE is achieved by the $Ag^\star$ method for only $\gamma_2, \gamma_4, \gamma_5$, and γ_9, it is very close to the best method in the other cases. The function has similar features at edge transects v_1 and h_1. They overlap at

the bottom-left corner, and there is a sharp discontinuity on v_1 compared to the moderate discontinuity on h_1.

The smallest MSE at v_1 ranges within the approximate interval of $(0.33, 0.4)$, but for h_1, it is $(0.17, 0.2)$. The two flat parts of the function in v_1 form a step that is higher compared to h_1 hence the MSE is larger at v_1. The $Ag^\star$ method (shown with solid triangles) has good performance in both transects. The point shown with ($\times$) symbol interestingly gives the smallest MSE for $\gamma_4 - \gamma_8$ at h_1. The other two edge transects, v_4 and h_3, also have similar features. A large part of both transects contain the same constant value of the function. A small part of h_3 at the top left corner of the function has slightly higher value than the remaining part. For v_4, the bottom right part of the function has higher values, hence the h_3 give smaller MSE.

If the intermediate transects are checked, for instance, the v_2, v_3, and h_2, the MSE values are the highest for v_2 that passes over three blocks of the function and it is hard to suggest a specific weight method due to the inconsistent patterns. However, the MSE has a decreasing trend from regular to clustered points which validates the previous conclusions regarding the better function estimations using clustered points for the functions that have discontinuities and spikes.

7.2.4 Bumps

The Bumps test function has three spikes at different parts of the region. However, the spikes are due to the finite exponential increases in the function value rather than rectangular prisms as in Blocks. The function is constant on the rest of the region. The global MSE in Figure 7.4 shows a decreasing trend from regular to clustered points as it was in the Heavisine and the Blocks. These three functions may be considered to belong to the same family based on having sharp increases such as discontinuities and spikes.

The edge region MSE shows that the smallest values are achieved using the $Ag^\star$ method in most γ_i cases, and the Ag method has a comparable performance for regular and homogeneous points. But the performance of Ag becomes worse for clustered points. The edge transects v_1, v_5, h_1 and h_2 give the same conclusions in terms of the best weight methods. The solid and non-solid points mostly have similar MSE values except for the highly clustered cases. For general use, $Ag^\star$ weight method would be the preferred method since it has a consistent performance for all

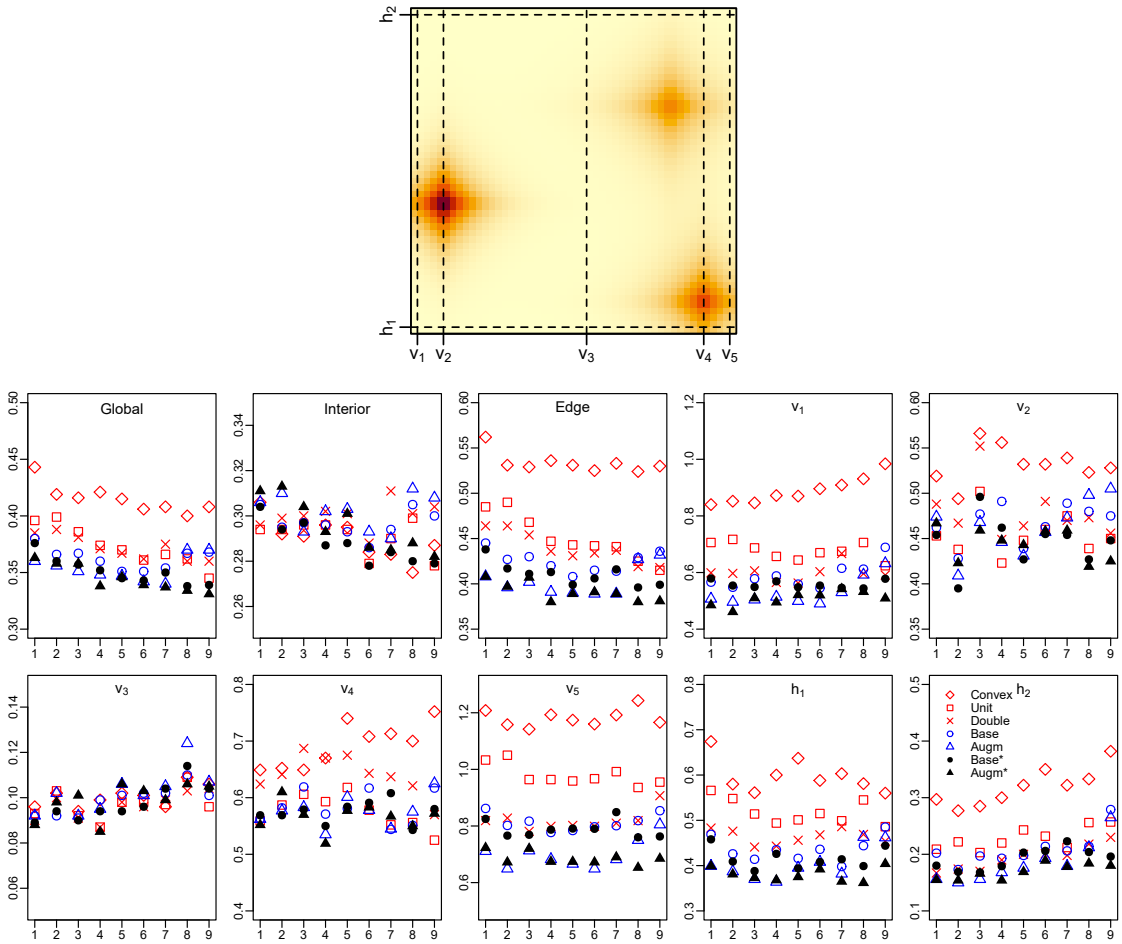


Figure 7.4: Lifting MSE results for Bumps at different parts and the transects. $x-$axis shows the index of γ_i for $i = 1, ..., 9$, and $y-$axis is the MSE which varies for each plot.

point patterns. The intermediate transect v_3 do not contain any variability of the function, hence the MSE is very similar and close to zero for all weight methods.

7.2.5 Maartenfunc

Maartenfunc is a completely different type of test function than the previously discussed ones. It has two separate intersecting planes where a discontinuity exists at the intersection line. The global MSE results for Maartenfunc in Figure 7.5 show an increasing trend from regular to clustered points. The regular point patterns give better function estimation in this case, since the function value of a selected point and its neighbours would be similar except near the discontinuity.

The smallest global and edge MSE values are achieved by the $Ag^{\star}$ method (with solid triangle) for all cases and this exists in most cases of the edge transect plots

v_1, v_3, h_1, and h_3, showing the $Ag^\star$ method to be more favourable. The minimum MSE in v_3 is always higher than 0.10 for all γ values, but it is always smaller than 0.05 in v_1 and h_3. Maartenfunc has its highest values at v_3 and the lifting has a weakness on estimating the maximum values of the function, hence the MSE at the transects where the function has local or global maximums are higher. Higher MSE was also observed in Chapter 6 when the transect bins coincided with the local or global maximums of the function. In this chapter, the information based on the transect bins are collapsed for the ease of interpretation.

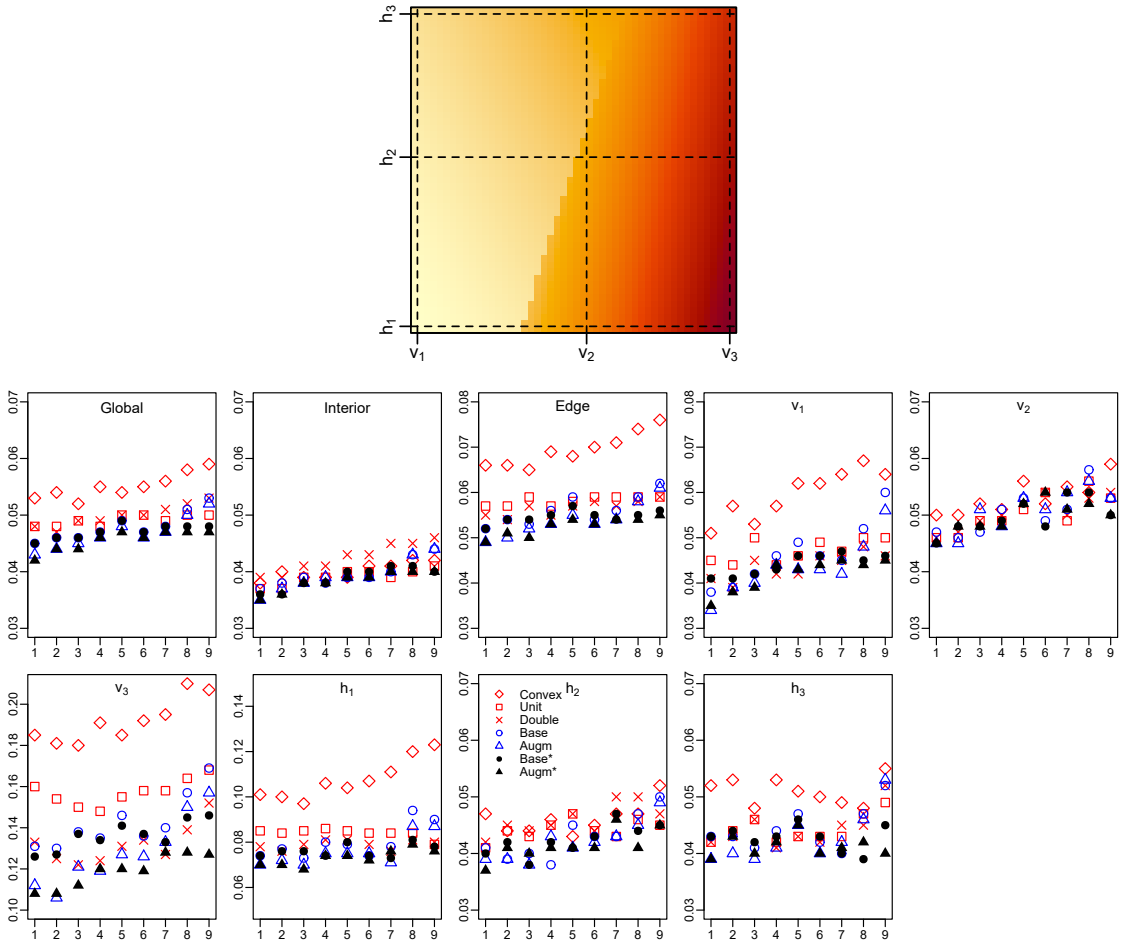


Figure 7.5: Lifting MSE results for Maartenfunc at different parts and the transects. $x-$axis shows the index of γ_i for $i = 1, ..., 9$, and $y-$axis is the MSE which varies for each plot.

7.3 Comparison of lifting estimates with kriging

In this section, we compare the lifting method using the suggested weight method $Ag^\star$ to kriging or Gaussian process regression which is a standard spatial interpola-

tion method introduced in Cressie (2015). Given the observations $\{z(x_1), \ldots, z(x_n)\}$ at locations $\{x_1, \ldots, x_n\}$, kriging aims to give a linear prediction of the value $Z(x_0)$ at a location x_0 that is the Best Linear Unbiased Estimator (BLUE) (Christensen, 1991). We use the ordinary kriging method which assumes spatial stationarity in $Z(x) = \mu + \epsilon(x)$ where the unknown constant μ is mean of $Z(x)$. The linear prediction at a location x_0 is made as,

$$\hat{Z}(x_0) = \sum_{i=1}^{n} \lambda_i Z(x_i) + \epsilon(x_0), \tag{7.1}$$

where λ_i is the kriging weight and estimated by minimizing the prediction variance as

$$\begin{aligned} \{\hat{\lambda}_i\}_{i=1}^n &= \operatorname{argmin} \mathrm{E}[\epsilon(x_0)^2] \\ &= \mathrm{Var}[\hat{Z}(x_0) - Z(x_0)] \\ &= \mathrm{E}[\{\hat{Z}(x_0) - Z(x_0)\}^2] - \{\mathrm{E}[\hat{Z}(x_0) - Z(x_0)]\}^2. \end{aligned} \tag{7.2}$$

Denote $\boldsymbol{X} = (x_1, \ldots, x_n)^\top, \Sigma = \mathrm{Cov}(\underline{\boldsymbol{X}})$ and $\boldsymbol{c} = \mathrm{Cov}(\underline{\boldsymbol{X}}, x_0)$ and $\boldsymbol{c} \in \mathbb{R}^n$, the linear prediction of $Z(x_0)$ minimizing (7.2) is

$$\hat{Z}(x_0) = \boldsymbol{c}^\top \Sigma^{-1} \boldsymbol{X}. \tag{7.3}$$

We used the functions in the `R` package `gstat` developed by Gräler *et al.* (2016); Pebesma (2004) to perform kriging. Although kriging is mainly used to estimate the value of a random variable over a continuous region using unknown locations such as the grid points, our study aims to attain estimations at the actual data locations by detecting and separating the measurement errors. In this case, the kriging estimates can be compared to the lifting estimates. We used a '*nugget effect*' to determine the short scale random variability in the data. The value of the nugget term can be obtained from the variogram as the intercept of the variogram function at a lag distance of almost zero. A large nugget effect value indicates high short-range variability in the random variable and would lead to smooth kriging estimates.

The same replicates of regular and clustered point patterns and the same noisy function values are used to perform the kriging as explained in Section 7.1. The kriging estimates are obtained for the data locations for 250 replicates, using the Doppler, Heavisine, Blocks, Bumps and Maartenfunc for various regular, homogeneous, and clustered point patterns. In Table 7.1 we compared the function estimation results

using $Ag^{\star}$ which was the most favourable weight method in lifting, and kriging results in terms of MSE values with its standard errors. We only selected $\gamma_1, \gamma_5, \gamma_9$ point pattern cases rather than using all alternatives, and summarized the results based on the global, interior and edge MSE. For each case, the method with the smallest MSE is highlighted in blue colour. The presented values are for MSE$\times$100, and $se \times 100$.

In the Doppler test function, $Ag^{\star}$ method performed better than kriging for all cases. The variability for the $Ag^{\star}$ is similar for interior and edge regions however the variability at the edges is always higher than interior in kriging. Although the Doppler test function generally has smooth features which is suitable for kriging estimation, there is a highly oscillated part located at the corner which cause very large MSE for kriging.

We make opposite conclusions for Heavisine, for which kriging performs better in all cases with less variation in the estimated function values. The periodic waves of Heavisine can be better estimated by kriging, and both $Ag^{\star}$ and kriging have higher MSE for interior region where the spike occurs.

The Blocks test function includes various discontinuities over the surface which is difficult for kriging to capture. Kriging over-smooths the blocks and hence causes very high MSE and standard error. It is important to note that the MSE is higher for the interior part where the blocks mostly take place and the edges are flat with minor discontinuities.

Kriging gives better performance in the Bumps test function for homogeneous and clustered points, and only for the interior of the regular points. The MSE at the edges where the bumps are located is higher than the interior. Since the bumps are not sharp discontinuities, the kriging method still works better than $Ag^{\star}$.

Finally, in the Maartenfunc, both methods perform similarly but kriging MSE and standard errors are slightly smaller especially for the clustered point patterns. Kriging would not find it difficult to estimate the flat parts of the piecewise linear functions, but to understand how well it estimates the parts with discontinuity, we may check the results for the transect h_1 which is both an edge transect and contain the sharpest discontinuity. Since we are interested in the part of the h_1 where the discontinuity happens, we cut the transect and take only the part when $x = [0.2, 0.4]$. The MSE results are as follows; Kriging: $(R : 17.52, H : 16.63, C : 15.14)$, and $Ag^{\star}$: $(R : 14.13, H : 11.69, C : 13.42)$ for regular, homogeneous and clustered points respectively. It is clear that the $Ag^{\star}$ method performs much better than kriging

where the discontinuity exists, and this is valid for the other transect sub-parts as well.

If entire transect, or the global, interior and edge parts are considered, the dominance of the good estimation results at the smooth parts of the function in that transect may overshadow the identification of the local performance where the discontinuity occur. Hence checking finer details where the discontinuity happen is more accurate as we uncovered. To sum up, the lifting scheme with $Ag^\star$ method has significantly better performance for the test functions or sub-regions with the discontinuities, which kriging over-smooths. For some cases in Bumps and in Maartenfunc, the $Ag^\star$ method compares favourably with kriging, and kriging perform better for the functions with smooth features.

		Regular		Homogeneous		Clustered	
	Method	$Ag^\star$	*Kriging*	$Ag^\star$	*Kriging*	$Ag^\star$	*Kriging*
DP	Global	4.41 ± 0.02	8.95 ± 0.04	4.33 ± 0.02	8.73 ± 0.04	4.20 ± 0.02	8.82 ± 0.04
	Interior	4.03 ± 0.03	6.80 ± 0.04	3.99 ± 0.03	7.17 ± 0.05	3.75 ± 0.02	7.26 ± 0.05
	Edge	4.75 ± 0.03	10.83 ± 0.07	4.65 ± 0.03	10.2 ± 0.06	4.65 ± 0.03	10.42 ± 0.07
HV	Global	44.19 ± 0.20	34.14 ± 0.15	29.17 ± 0.13	17.19 ± 0.08	22.77 ± 0.10	14.73 ± 0.07
	Interior	45.74 ± 0.30	37.30 ± 0.24	30.37 ± 0.19	19.15 ± 0.12	23.74 ± 0.15	16.68 ± 0.10
	Edge	42.84 ± 0.26	31.38 ± 0.19	28.02 ± 0.17	15.30 ± 0.10	21.79 ± 0.14	12.72 ± 0.08
BL	Global	55.03 ± 0.25	200.68 ± 0.90	52.27 ± 0.23	158.26 ± 0.71	50.30 ± 0.22	156.28 ± 0.70
	Interior	86.91 ± 0.57	361.71 ± 2.37	79.06 ± 0.50	271.79 ± 1.73	74.14 ± 0.47	259.02 ± 1.62
	Edge	27.15 ± 0.17	59.81 ± 0.37	26.60 ± 0.17	49.42 ± 0.31	26.25 ± 0.17	50.42 ± 0.32
BM	Global	36.26 ± 0.16	42.32 ± 0.19	34.61 ± 0.15	25.53 ± 0.11	33.12 ± 0.15	21.92 ± 0.10
	Interior	31.12 ± 0.20	22.79 ± 0.15	30.09 ± 0.19	16.06 ± 0.10	28.23 ± 0.18	14.22 ± 0.09
	Edge	40.75 ± 0.25	59.40 ± 0.36	38.93 ± 0.24	34.61 ± 0.22	38.06 ± 0.24	29.86 ± 0.19
MR	Global	4.22 ± 0.02	3.02 ± 0.01	4.66 ± 0.02	3.11 ± 0.01	4.73 ± 0.02	3.31 ± 0.01
	Interior	3.49 ± 0.02	3.02 ± 0.02	3.88 ± 0.02	3.03 ± 0.02	4.00 ± 0.03	3.25 ± 0.02
	Edge	4.86 ± 0.03	3.02 ± 0.02	5.42 ± 0.03	3.19 ± 0.02	5.47 ± 0.03	3.37 ± 0.02

Table 7.1: Mean squared errors with standard errors (both $\times 100$) for lifting estimates using $Ag^\star$ method and kriging. Only the results for $\gamma_1 = 0$: regular, $\gamma_5 = 1$: homogeneous, and $\gamma_9 = 3$: highly clustered points are shown. Row panels show the global, interior and edge MSE for Doppler, Heavisine, Blocks, Bumps, and Maartenfunc respectively.

7.4 Real data application of lifting

In this section, we present lifting estimation results for the real data sets `spruces`, Barro Colorado Island (BCI), `waka`, `finpines`, and `longleaf` described in Section 4.3.2. These data sets are particularly selected since the estimated $\hat{\gamma}$ for the data sets fall in the interval $[0, 3]$ we used in the simulations, and the real data sets also have examples of regular, clustered and homogeneous points. Also, the number of data locations in these data sets has a large range. The sizes of each

data, estimated parameter $\hat{\gamma}$, the sampling region Ω, and the data set descriptions are given in Table 4.5.

In the area prediction, only the locations of the points were necessary to tessellate the points in the sampling region and to calculate the cell properties. However, lifting requires both the data locations x_i for $i = 1, \ldots, n$ and the observations y_i at the locations y_i. The `spruces`, `waka`, `finpines`, and `longleaf` data sets from the `spatstat` package contain locations of the trees (x_i) and tree diameter observations (y_i). The Barro Colorado Island data set is based on the soil nutrient measurements (y_i) at the sampled locations (x_i) in a region. There are several chemicals measured but we only used the Aluminum level in the soil.

These real data examples have different structures; the BCI data set is a geo-referenced data that we measure chemical levels at sampled locations and these type of data are ideally analyzed using kriging methods. The remaining data sets are marked point patterns where the marks are the tree diameter and height. We initially used the locations obtained from these data sets in Chapter 4 for the prediction of Voronoi cell area. We also use these data sets for the application of the lifting scheme since it is our intention to see how the lifting method works in different data structures. Therefore, in this chapter, we are more interested in the illustration of lifting method for real data sets that have homogeneous, clustered, regular, and sampled points with different sizes and boundary windows, and measurements rather than solving a real life problem. Hence we aim to inspect the if lifting method has limitations in certain cases.

We concluded in Section 7.2 that the weight method $Ag^{\star}$ has the best overall performance for the test functions over varying point patterns in terms of the accuracy of the lifting estimates. Therefore, it is suggested as the best method and we apply it to the real data since its validity has been demonstrated based on the different configurations in the simulation study. We performed the lifting scheme for the real data sets based on the tree diameters at tree locations, and the Aluminum levels at the sampled locations. The forward transform, thresholding and the inverse transform procedures give the lifting estimations at the locations.

The observed values and the results of the estimated values are visualized in Figure 7.6 for all real data sets together. Rows correspond to `spruces`, BCI, `waka`, `finpines`, and `longleaf` data sets respectively. In each row, the left plot is the observed measurements, centre is the lifting estimates, or the denoised values, and the normal quantile-quantile plot of the residuals are on the right. Voronoi cells are coloured based on the observed or estimated values. In this thesis, we do not

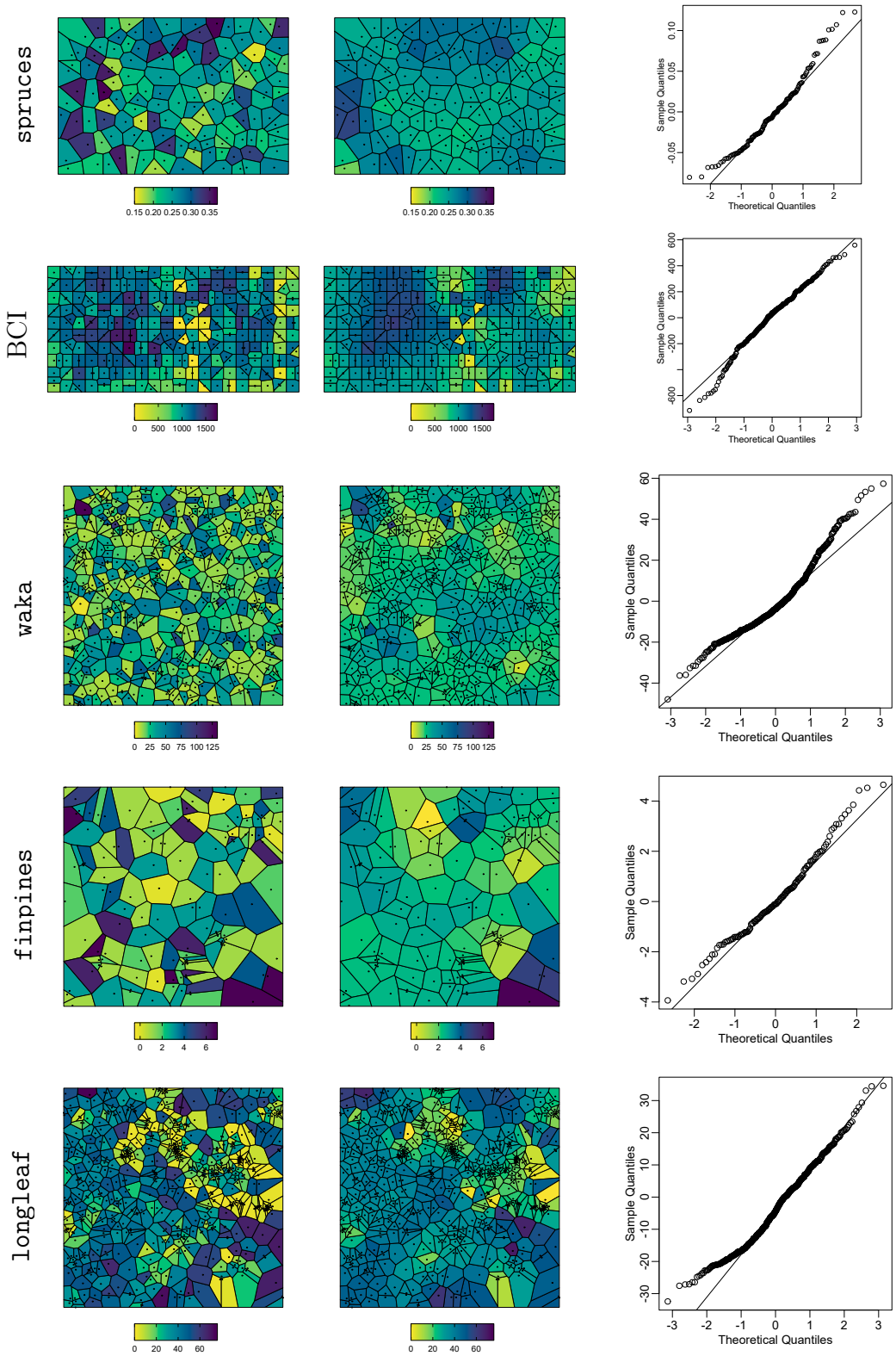


Figure 7.6: From top row panel to bottom, the lifting results are presented for `spruces`, BCI, `waka`, `finpines`, and `longleaf` data sets respectively. At each row panel, original tree diameter (left), lifting estimations (centre), the normal q-q plot of the residuals (right) are shown.

study the imputation feature of the lifting at unobserved locations such as the grid points. It is one of the challenging aspect of the lifting as discussed in Heaton & Silverman (2008) and Peck (2010). The estimated surface is created by disjoint polygons which are piecewise linear sub-surfaces.

The presentation order of the results is determined based on their estimated $\hat{\gamma}$ parameter values from Table 4.5. The `spruces` data set has the most regular pattern in the first row panel. The original tree diameter measurements have some irregularity with observed high diameter near the top and left boundary lines and small diameters at intermittent locations. The lifting estimates at the centre plot creates a quite smooth pattern. The estimated high values are located near the top and left boundary lines where the high diameter was observed at the relevant locations. The locations at the centre of the region are mainly similar and take approximately the mid value of the scale.

There is over-smoothing cases at some locations, for instance, two points at the top right corner originally have very small and high diameters that are coloured in green and dark blue, however, lifting estimated the very similar values for these points. Hence over-smoothing might be a concern here. The normal q-q plot of the residuals on the right show that the majority of the points are located on the reference line, but minor violations indicate a slightly right skewed distribution.

The Barro Colorado Island data in the second row panel has different nature due to the mixture of regular and irregular points. The sampling region is rectangular, hence the BCI data set is an example of lifting in a non-square region. The measurements (y_i) at the locations (x_i) is the soil Aluminum level. The smoothing in the lifting estimates at the centre is reasonable and not over-smoothed. We also see some of the discontinuities are preserved from the observed values. The soil Aluminum estimation is the lowest at the centre of the region and top right corner, and high values are observed at the centre-left and top-centre-right. If the overall surfaces for measured and estimated values are compared, the estimated values seem to be an appropriately denoised version. The residuals for this data has a slight left-skewed shape.

A homogeneous point pattern, `waka`, is given the third row panel. It is actually difficult to identify any pattern from the observed tree diameters on the left. One extreme observation is located around the top left corner for which the estimated value is smaller. The overall pattern of the lifting estimates do not show anomaly and occasional patterns are identified. Although irregular discontinuities occur near the boundary, the lifting estimations are smooth. The centre plot gives a sensible

underlying pattern using lifting. The right skew on the residuals is heavier than that seen for `spruces`.

Although the tree locations in `finpines` data set constitute some clustering, the clusters are not isolated from each other. The pattern of the tree locations generally looks homogeneous but contains clusters at slightly right of centre at the bottom, and the top-right corner. The original tree diameter measurements look irregular and it is hard to identify if any pattern exists. However, the lifting estimations on the centre plot show estimated values that clarifies the underlying pattern by smoothing the random variations and also preserving discontinuities. We do not see any anomalous boundary effect in the estimated values. The lifting estimates are the highest at the bottom right corner due to the high observed values, however, some of the cells with light yellow colour became darker due to the smoothing.

In the original tree diameter values, there are examples of two trees very close to each other, one with large and the other with very small diameter. It is possible for two adjacent trees to have small diameters, but it is not likely to have both trees to have large diameter. The lifting estimates for such adjacent trees are indistinguishable although they are very different in the reality. Other high diameter values are estimated at the top-left corner and top-centre, and there are three parts where the estimated diameter is small two of which are near the clusters, and one is closer to the top-centre. At the other locations, the lifting estimates are similar and close to the median value.

The last row panel in Figure 7.6 shows the results for the `longleaf` data which also show some degree of clustering. At the highly clustered locations, trees have smaller diameter coloured in yellow and the most high values are observed at the bottom-right and bottom-centre. The lifting estimates clarify the underlying pattern that is smooth but the parts where the tree diameter is small and high are still noticeable. The residuals have a skewed distribution as observed in the previous data sets.

7.5 Conclusions

This chapter extends the application of lifting for regular and clustered data scenarios, and real data sets. As concluded in Chapter 6 which suggested the usage of the adjusted weights rather than the observed weights, and $Ag^\star$ in particular. Since cell properties such as the area depend upon global point intensity ρ for homogeneous points, and local intensity ρ_i for regular and clustered points, we estimated the $\{\hat{\rho}\}_{i=1}^{n}$ and scaled the covariates with respect to $\hat{\rho}_i$ and highlighted the area

prediction with $\star$ superscript. The base and augmented model prediction of area using $\hat{\rho}_i$ is denoted as $B^\star$ and $Ag^\star$ which are used as the new weight methods in this chapter hence the total number of weight methods we consider is increased to seven.

The performances of the weight methods are investigated for the test functions using regular and clustered point patterns with different degrees of regularity and clustering. The process we used to generate regular and clustered points is the saturation process by Geyer (1999). The lifting estimates are examined globally, interior, and edge regions, and at different transects. The $Ag^\star$ method gave the smallest MSE for most cases, especially near the boundaries. The usage of the standard weight methods that are the observed cell area using the boundaries do not give satisfactory results. In fact, we demonstrated that the convex hull boundary has a poor performance. The area prediction model framework we proposed can be used in the situations where the spatial data has a boundary that is either known or unknown as we explained in Section 3.5.1 and 3.5.2.

The adjusted weight methods $B, Ag, B^\star$ and $Ag^\star$ generally have compatible performances, except for highly clustered points. When points are highly clustered, the range for $\{\hat{\rho}_i\}_{i=1}^n$ is expected to be higher than the regular points and hence the cell area is highly affected by the local intensities. We concluded in Chapter 4 that the area prediction is not robust for regular and clustered points and it is better to use the estimated local intensity $(\hat{\rho}_i)$ to scale the covariates which gave better predictions. Similarly, in lifting, the use of $B^\star$ and $Ag^\star$ methods gives better function estimations compared to B and Ag. For the regular points, the lifting is robust within the adjusted weight methods.

The area prediction results in Chapter 3 and 4 showed that the base model gave an overall smaller MSE and the augmented method gave larger MSE but reduced the maximum error. It is surprising that the usage of augmented model in the lifting generally outperformed the base model but the lifting estimations in some cases were very similar. It is our conclusion that the reduced maximum error on the area prediction yield better lifting estimates since the area is predicted more accurately, or the predicted area that has an extreme error gives unstable lifting estimates since the area is used as the weight. Hence, although it is difficult to suggest one of the base or augmented methods strictly, we recommend that if one would wish to reduce the maximum error, then augmented models may be used for both area prediction and lifting scheme.

There are cases where the lifting using the $Ag^{\star}$ weight method performs better against kriging for the test functions, or the parts of the test functions. The better performance of the lifting is usually near the edges, and on the sub-regions where discontinuities occur. We performed lifting using the $Ag^{\star}$ method for real data sets which are examples of regular, homogeneous, and clustered point patterns. It is useful to apply the method on such data sets that forces the previously considered settings of the lifting such as the highly clustered and regular point patterns, having large number of points, and rectangle boundaries. However, the lifting scheme works well for all real data sets and the results are satisfactory. The only downside of the algorithm is to become computationally expensive for large n since matrix calculations are involved which would also happen in many other spatial data analysis method.

Chapter 8

Discussion

This thesis investigated the statistical properties of Voronoi cells in bounded regions, proposed ways that consider the data in a finite region as if it is in the infinite plane, and implemented this method into the lifting scheme framework which is a denoising method for spatial data. We started the thesis with the investigation of the statistical properties of Voronoi cells in the bounded regions using various boundary types in Chapter 2. This part of the study was based on the homogeneous Poisson points with a specific point intensity and discovered the effects of imposed boundaries on the cell properties such as cell area, perimeter, and number of cell edges. The distributions of the cell properties differed for the cells that are close to the boundary compared to the cells in the infinite plane. Also, we found that the boundary type matters. The study was also carried out for various unit intensities of points and we found that the differences in the cell properties near the boundaries remained.

Our initial study in Chapter 2 raised an important concern in the analysis of spatial data that usually come within a finite region and depend on a neighbourhood structure. In the case of bounded or finite regions, the neighbourhood structure is disrupted by the boundary. Therefore a data point located at or near the edge or the corner of the region may only have neighbours occasionally. This may cause issues if the neighbourhood structure is used in the analysis of the spatial data such as the lifting scheme we used.

In the lifting scheme, a weighted sum of the values of the neighbours is calculated. For a data point located at the edge or corner of the region, there might be only a few neighbours which the weighted sum is calculated from, and no neighbours on the other side. Boundaries act as a cutoff point and hence no further observations are

available. In the context of Voronoi tessellations, boundaries restrict the Voronoi cells that affect the cell area. Since the lifting scheme uses both the neighbourhood structure and the cell area that are determined by the Voronoi cells, our findings in Chapter 2 have importance on understanding the boundary effects on the function estimation.

We devised a process in Chapter 3 that treats the areas of Voronoi cells in the bounded region as if they are in an infinite plane by area adjustment based on a regression-based method. We extended this method for regular and clustered data case in Chapter 4 and combined it with the lifting scheme later in Chapter 6 and 7 and compared the performances of the proposed and standard methods. However, the approaches we proposed in Chapter 3 and 4 would have a general potential use in the analysis of spatial data.

One application would be the case where data are represented as a marked point process, with the marks being related to the area surrounding each point. For instance, consider ecological or forestry data that contain plant or tree locations and the territories that the plants occupy. Voronoi cells can be considered as the territories that the plants occupy, and we would adjust the cells (territories) near the edges using the method we created. This would be useful when the areas of Voronoi cells are considered as the mark process that is associated with a point process, hence the correction of the cell areas would aim to reduce the bias near the edges. Such issues related to dependencies between the marks and locations are mentioned in Schlather *et al.* (2004).

An exploratory analysis is performed in Section 3.6 for the classification of the cells that are likely to be affected by a boundary. We used the simplest approach to classify the boundary effected cells which gave promising results. However, the classification of boundary effected cells can be a separate extensive study where more sophisticated binary classification methods such as decision trees or random forests.

The lifting scheme in two dimensions based on Voronoi tessellations is the mechanism we used throughout the thesis based on the specifications in Jansen *et al.* (2009). The lifting scheme requires data locations and observations, and has vital configurations which depend on cell area such as the decision of the lifting order, and the weights that are used in the calculations. Voronoi tessellations assist to handle these facets of the method. However, one should be careful about the boundary effects on the Voronoi cells. The standard way of performing the Voronoi-based lifting is to use the cell areas as the weights. If the data is given within a finite

region, the actual sampling region Ω or the convex hull can be taken as the boundary as standard options to calculate the cell areas. However, Ω may not always be available or we would like to use a more sophisticated way that deals with this issue. Here the area adjustment method introduced in Chapter 3 becomes a functional tool to assign new weights to the cells rather than using observed cell area.

The steps of the lifting scheme are described in Chapters 5 which we explained the role of the weights. The method in Chapter 3 and 4 works well in conjunction with the lifting scheme and gives promising results, as presented in Chapter 6 and 7. When such a method is suggested, it is important to validate its performances for various scenarios so we considered many different point patterns, test functions, and weight methods, and compared the results with methods such as kriging which our suggested method compares favourably to many cases. However, the settings of the configurations can be expanded and other situations may be taken into account. For instance, we considered homogeneous Poisson points, clustered points, and regular points, but there are many other point processes for which the lifting scheme and cell area adjustment method can be tested. Also, we focused on two essential boundary types: convex hull and unit square, and have not used other types of imposed or real boundaries to avoid moving beyond the scope of the thesis, but this is a potential avenue for future work.

We understood from this thesis that the geometrical properties of Voronoi cells change when the boundaries are imposed. This consequence should be contemplated in the usage of methods that rely on Voronoi tessellations. We used the cell area in the lifting scheme but there are many areas where the properties of Voronoi cells are used such as astronomy, geology, agriculture, physics, and wireless networks. It is important to consider the impact of the boundaries on the cell perimeter and the number of cell edges when these cell properties are used in a study solely or in conjunction with other methods. Furthermore, the study of Voronoi tessellations in two-dimensional bounded regions can be expanded to three-dimensional case. Although the three-dimensional Voronoi tessellation is investigated in Kumar *et al.* (1992); Lazar *et al.* (2013); Muche (1996); Tanemura (2003), the focus was not on the properties of the polyhedrons due to imposed boundaries. The lifting scheme in three dimensions using the Voronoi polyhedrons would be another important future study.

The lifting scheme we used aims to estimate underlying true patterns separated from noise by the inverse transform of the thresholded detail coefficients. These

estimations are made for the data locations itself. However, another interesting objective would be to estimate the value at an unobserved location. The lifting scheme is a recently developed method and most of its aspects are still being developed. Heaton & Silverman (2008) and Peck (2010) introduced lifting-based imputation methods which our weight method approaches might be combined with, and the performances of different weight methods could be tested. Finally, we have checked the q-q plots of the lifting estimates for real data cases and the results show that the residuals do not obey a particular parametric distribution. Hence, the future work may also consider the residual analysis of the lifting estimates.

Appendix A

Extra plots and tables

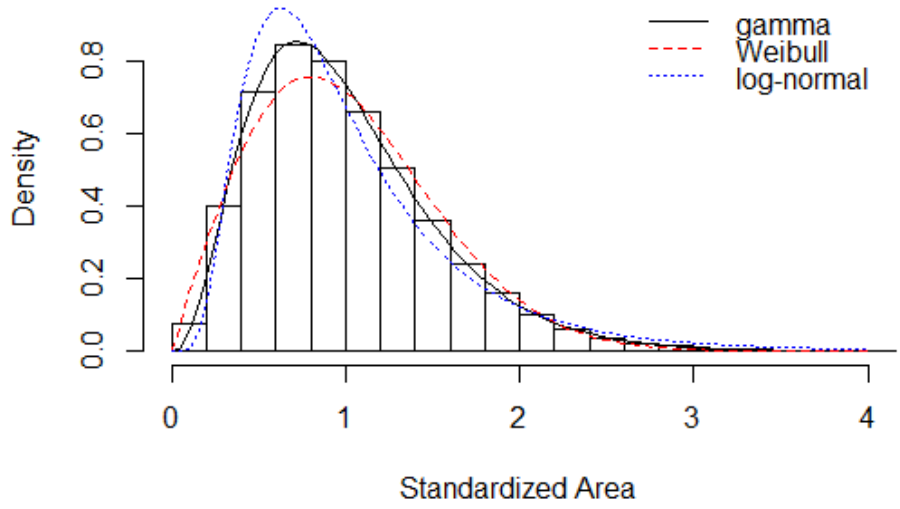


Figure A.1: Gamma, Weibull and log-normal distributions are fitted for the standardized cell area in the infinite plane.

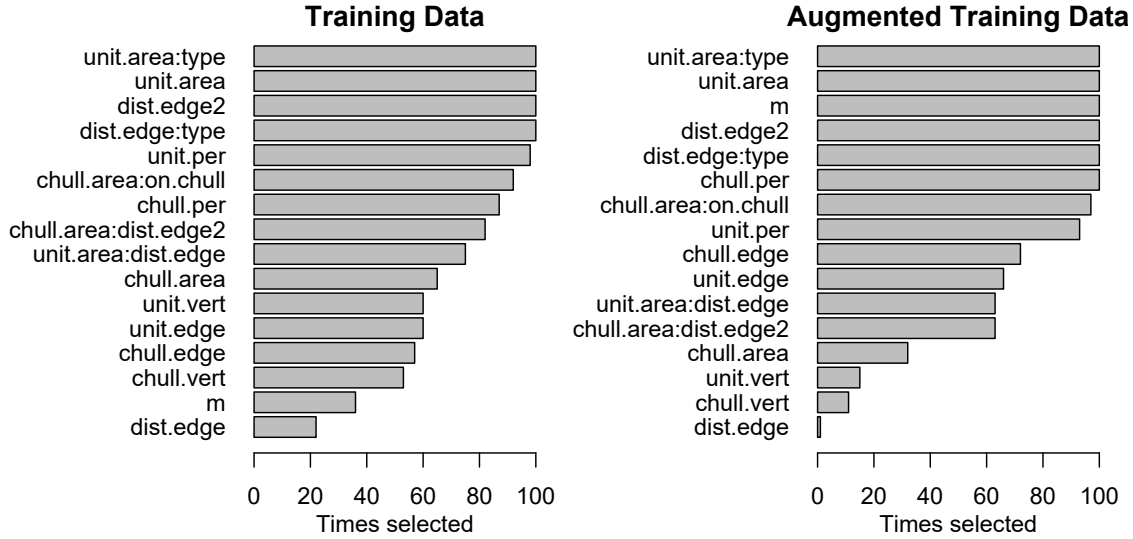


Figure A.2: Selected variables in the unit square boundary models when the related variables are removed. Results are given for the base models (left) and augmented models (right).

Appendix B

Test functions and R Codes

B.1 Test functions

Let us specify the theoretical definitions of the test functions.

Doppler

$$f(x,y) = \sin\left(\frac{1}{x^2+y^2}\right), \quad 0 < x, y \leq 1 \tag{B.1}$$

Heavisine

$$\begin{aligned} f_1(z;\mu,\sigma^2) &= \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{z-\mu}{\sigma}\right)^2} \\ f_2(x,y) &= \sin(a\sqrt{x^2+y^2}) \\ f_3(x,y) &= f_1(x;\mu_1,\sigma^2) f_1(y;\mu_2,\sigma^2) \\ f(x,y) &= f_2 + f_3 \quad 0 \leq x, y \leq 1 \end{aligned} \tag{B.2}$$

$$(a = 20,\ \sigma = 0.01,\ p = 0.005,\ \mu_1 = 0.55,\ \mu_2 = 0.50)$$

Blocks

$$f(x,y) = \begin{cases} 1 & \text{if} \quad 0 \le x < 0.1 \\ 2 & \text{if} \quad 0 \le y < 0.2 \\ 3 & \text{if} \quad 0.3 < x < 0.4,\ 0.7 < y < 0.8 \\ 4 & \text{if} \quad 0.7 < x < 0.8,\ 0.7 < y < 0.8 \\ 5 & \text{if} \quad 0.5 < x < 0.6,\ 0.4 < y < 0.6 \\ 6 & \text{if} \quad 0.3 < x < 0.8,\ 0.2 < y < 0.3 \\ 7 & \text{if} \quad 0.2 < x < 0.3,\ 0.3 < y < 0.4 \\ 8 & \text{if} \quad 0.8 < x < 0.9,\ 0.3 < y < 0.4 \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } 0 \le x, y \le 1 \tag{B.3}$$

Bumps

$$f_1(z;\mu,b) = \frac{1}{2b}\exp\left\{-\frac{|z-\mu|}{b}\right\}$$

$$f(x,y) = \sum_{j=1}^{3} f_1\left(x;\mu_j^x,\sqrt{b_j}\right) f_1\left(y;\mu_j^y,\sqrt{b_j}\right) \quad 0 \le x, y \le 1 \tag{B.4}$$

$$(\mu^x = (0.1, 0.8, 0.9),\ \mu^y = (0.4, 0.7, 0.1),\ b = (0.01, 0.02, 0.015))$$

Maartenfunc

$$f(x,y) = \begin{cases} 2x + y & \text{if} \quad 3x - y < 1 \\ 5x - y & \text{if} \quad 3x - y \ge 1 \end{cases} \qquad \text{for all } 0 \le x, y \le 1 \tag{B.5}$$

```
## Doppler test function
dopplerfunc <- function (x,y) {
  r = sqrt(x^2 + y^2)
  f = sin(1/(r^2))
  f }

## Blocks test function
blockfunc <- function(x,y){
    f <- rep(0, length(x))
    sv <- x < 0.1; f[sv] <- f[sv] + 1
    sv <- y < 0.2; f[sv] <- f[sv] + 2
    sv <- (x>0.3)&(x < 0.4)&(y<0.8)&(y>0.7); f[sv] <- f[sv] + 3
    sv <- (x>0.7)&(x < 0.8)&(y<0.8)&(y>0.7); f[sv] <- f[sv] + 4
    sv <- (x>0.5)&(x < 0.6)&(y<0.6)&(y>0.4); f[sv] <- f[sv] + 5
    sv <- (x>0.3)&(x < 0.8)&(y<0.3)&(y>0.2); f[sv] <- f[sv] + 6
    sv <- (x>0.2)&(x < 0.3)&(y<0.4)&(y>0.3); f[sv] <- f[sv] + 7
    sv <- (x>0.8)&(x < 0.9)&(y<0.4)&(y>0.3); f[sv] <- f[sv] + 8
    f }

## Heavisine test function
heavisinefunc = function(x,y, pp=0.005, sd=0.01, freq=20){
  r = sqrt(x^2 + y^2)
  f1 = sin(freq*r)
  f2 = pp*dnorm(x,0.55,sd=sd)*dnorm(y,0.5, sd=sd)
  f1+f2
  }

## Bumps test function
bumpsfunc <- function (x,y) {
xc = c(0.1, 0.8, 0.9); yc = c(0.4,0.7, 0.1); vc = c(0.01,0.02, 0.015)
nc = length(xc)
ans = rep(0,length(y))
for(i in 1:nc) {ans = ans + doubexp(x, mean=xc[i], rate=sqrt(vc[i]))*
                            doubexp(y, mean=yc[i], rate=sqrt(vc[i]))}
ans}
doubexp = function(x, mean=0, rate=1){exp(-abs(x-mean)/rate)/(2*rate)}

## Maartenfunc
maartenfunc <- function(x,y){
  fun = numeric()
  for (i in 1:length(x)) {
    if((3*x[i] - y[i]) < 1) {fun[i] = 2*x[i] + y[i]}
    if((3*x[i] - y[i]) >= 1){fun[i] = 5*x[i] - y[i]}
  }
  fun
}
```

B.2 Example code for statistical properties of Poisson Voronoi cells

```
require(deldir)
require(tripack)
require(rgeos)

## function to calculate distance from a point to a line
pt.ln = function(x0, y0, x1, y1, x2, y2){
  distance = abs((y2-y1)*x0 - (x2-x1)*y0 + x2*y1 - y2*x1)/sqrt((y2-y1)^2 + (x2-x1)^2)
  distance
}

## Generate points
set.seed(22)
rho = 200
n = rpois(1, rho)
x = runif(n, 0, 1);y = runif(n, 0, 1)

## Voronoi tessellation
tes = deldir(x,y, rw = c(0,1,0,1))

## Convex hull of the points
chull1 = convex.hull(tri.mesh(x, y))
poly1 = Polygon(cbind(chull1$x,chull1$y))
p1 = SpatialPolygons(list(Polygons(list(poly1), "p1")))

## Define variables
unit.area = unit.per = unit.edge = chull.area = chull.per = chull.edge = numeric()
dist.edge = dist.edge2 = dist.cent = unit.vert = chull.vert = numeric()
on.chull = type = logical()

for (k in 1:n) {# Loop to calculate cell properties for n points

  ## Cell vertex coordinates
  ss = rbind(as.matrix(tes$dirsgs[(tes$dirsgs[,5] == k)|(tes$dirsgs[,6] == k),]))
  v = matrix(unlist(ss[, 1:4]), ncol = 4)

  bp1 = ss[, 7]
  bp2 = ss[, 8]

  v1 = cbind(v[, 1:2, drop = FALSE], 0 + bp1)
  v2 = cbind(v[, 3:4, drop = FALSE], 0 + bp2)

  vv = rbind(v1,v2)
```

```
angle = atan2(vv[, 2] - y[k], vv[, 1] - x[k])
angle.0 = sort(unique(angle))

vert = vv[match(angle.0, angle), ]
vv1 = vert[, 1]
vv2 = vert[, 2]
bp = as.logical(vert[, 3])

rw = tes$rw
i.crnr = get.cnrind(x, y, rw)
ii = i.crnr %in% k
x.crnrs = rw[c(1, 2, 2, 1)]
y.crnrs = rw[c(3, 3, 4, 4)]

vert.x = c(vv1, x.crnrs[ii])
vert.y = c(vv2, y.crnrs[ii])

## Cell vertices ordered
vert.coord = cbind(vert.x,vert.y)
f.bp = c(bp, rep(TRUE, sum(ii)))

f.sort = atan2(vert.coord[, 2] - y[k], vert.coord[, 1] - x[k])
f.sort.0 = sort(f.sort)

f.vert = vert.coord[match(f.sort.0, f.sort), ]

## Cell edge segments
lgth = numeric()
sgm = dim(f.vert)[1]

for (kk in 1:sgm) {
  lgth[kk] = if(kk+1 <= sgm){
       sqrt((f.vert[kk,][1]-f.vert[kk+1,][1])^2 +
            (f.vert[kk,][2]-f.vert[kk+1,][2])^2)
    }
  else{sqrt((f.vert[1, ][1]-f.vert[kk,][1])^2 +
            (f.vert[1,][2]-f.vert[kk,][2])^2) }
}

# ------------------------------------------------
unit.per[k] = sum(lgth) # <-- unit perimeter
# ------------------------------------------------
# ------------------------------------------------
unit.edge[k] = sgm # <-- unit edges
# ------------------------------------------------
# ------------------------------------------------
```

```
type[k] = sum(f.bp) # <-- cell type
# -----------------------------------------------

# Cell vertices as SP class
chull2 = convex.hull(tri.mesh(vert.coord[,1], vert.coord[,2]))
poly2 = Polygon(cbind(chull2$x, chull2$y))
p2 = SpatialPolygons(list(Polygons(list(poly2), "p2")))

# Intersect the cell with convex hull
res = gIntersection(p1, p2)

# ------------------------------------------------------------------------------
chull.area[k] = unlist(sapply(slot(res, "polygons"), function(p) sapply(
                              slot(p, "Polygons"), slot, "area")))
# ------------------------------------------------------------------------------

# vertices of the intersection
pts = matrix(unlist(sapply(slot(res, "polygons"), function(p) sapply(
                           slot(p, "Polygons"), slot, "coords"))), ncol=2)

# chull line segments
chull.line = numeric()
chull.lgth = dim(pts)[1]

for (jj in 1:(chull.lgth-1)) {

  chull.line[jj] =
    sqrt((pts[jj,][1]-pts[jj+1,][1])^2 +
         (pts[jj,][2]-pts[jj+1,][2])^2)
}

# ---------------------------------------------------------------
chull.per[k] = sum(chull.line)    # <-- Convex hull perimeter
# ---------------------------------------------------------------

# ---------------------------------------------------------------
chull.edge[k] = (chull.lgth - 1)  # <-- Convex hull edges
# ---------------------------------------------------------------

# ------------------------------------------------------------------------------
dist.edge[k] =  min(abs(x[k]-1), abs(x[k]-0), # <-- distance from the point to
                    abs(y[k]-1), abs(y[k]-0)) #     unit boundary
# ------------------------------------------------------------------------------

# ------------------------------------------------------------------------------
on.chull[k] = sum(on.convex.hull(tri.mesh(x,y),x[k], y[k])) # <-- on convex hull
```

```
# ------------------------------------------------------------------------

## unit vertex dist
rwin = matrix(c(0,0,0,1,
                0,1,1,1,
                1,1,1,0,
                1,0,0,0), 4,4)

d = dd = numeric()
for (zz in 1:(nrow(vert.coord))) {
  for (ll in 1:4) {
    d[ll] = pt.ln(vert.coord[zz, 1], vert.coord[zz, 2], rwin[1,ll],rwin[2,ll],
                                                        rwin[3,ll],rwin[4,ll])}
  dd[zz] = min(d)}

# ------------------------------------------------------------------------
unit.vert[k] = min(dd) # <-- min distance from the vertices to the
       to the unit square boundary
# ------------------------------------------------------------------------

## chull vertex distance
cwin = slot(poly1, 'coords')
cver = pts

d = dd = numeric()
for (zz in 1:(nrow(cver))) {
  for (ll in 1:(nrow(cwin) -1)) {
    d[ll] = pt.ln(cver[zz, 1], cver[zz, 2], cwin[ll,1],cwin[ll,2],
                                            cwin[ll+1,1],cwin[ll+1,2])}
    dd[zz] = min(d)}

# ------------------------------------------------------------------------
chull.vert[k] = min(na.omit(dd)) # <-- min distance from the vertices
                                       to the chull boundary
# ------------------------------------------------------------------------

## min distance from chull boundary
dc = numeric()
for (ll in 1:(dim(cwin)[1] -1)) {
  dc[ll] = pt.ln(x[k], y[k], cwin[ll,1],cwin[ll,2],
                             cwin[ll+1,1],cwin[ll+1,2])}

# ------------------------------------------------------------------------
dist.edge2[k] = min(dc) # <-- distance from the point to chull boundary
# ------------------------------------------------------------------------
# ------------------------------------------------------------------------
```

```
  dist.cent[k] = (0.5-x[k])^2 + (0.5-y[k])^2 # <-- distace from the centre
  # ------------------------------------------------------------------------------

}# end of loop

# ------------------------------------------------------------------------------
unit.area = tes$summary$dir.area # <-- unit area
# ------------------------------------------------------------------------------

sim.df = data.frame(x, y, unit.area, unit.per, unit.edge, chull.area,
   chull.per, chull.edge, dist.edge, dist.edge2, dist.cent,
   unit.vert, chull.vert,on.chull, type)
```

Appendix C

Tables of MSE values for regular and clustered data

	$\gamma = 0$							$\gamma = 0.25$						
	Conv.	Unit	Doub.	B_0	A_0	$B^\star$	$A^\star$	Conv.	Unit	Doub.	B_0	A_0	$B^\star$	$A^\star$
G	0.045	0.044	0.044	0.045	0.044	0.045	0.044	0.045	0.045	0.046	0.045	0.045	0.045	0.045
I	0.038	0.039	0.040	0.040	0.040	0.040	0.040	0.039	0.041	0.042	0.040	0.041	0.041	0.041
E	0.052	0.048	0.048	0.049	0.048	0.050	0.047	0.051	0.048	0.049	0.049	0.048	0.049	0.048
v_1	0.100	0.090	0.087	0.087	0.087	0.091	0.084	0.103	0.092	0.089	0.090	0.090	0.089	0.087
v_2	0.015	0.014	0.012	0.013	0.012	0.013	0.012	0.015	0.014	0.013	0.012	0.011	0.012	0.011
h_1	0.105	0.093	0.090	0.089	0.086	0.091	0.082	0.106	0.090	0.088	0.093	0.088	0.091	0.086
h_2	0.014	0.013	0.012	0.012	0.011	0.012	0.011	0.014	0.013	0.012	0.012	0.012	0.013	0.012
D	0.068	0.068	0.069	0.066	0.067	0.069	0.065	0.072	0.066	0.068	0.067	0.065	0.068	0.064
G	0.468	0.456	0.455	0.451	0.444	0.449	0.442	0.451	0.430	0.432	0.424	0.422	0.419	0.420
I	0.425	0.438	0.447	0.451	0.459	0.451	0.457	0.408	0.417	0.428	0.424	0.435	0.421	0.439
E	0.506	0.472	0.461	0.450	0.431	0.446	0.428	0.489	0.442	0.435	0.424	0.410	0.418	0.403
v_1	0.551	0.517	0.508	0.494	0.480	0.501	0.480	0.554	0.488	0.476	0.481	0.460	0.468	0.463
v_2	0.818	0.678	0.588	0.584	0.520	0.573	0.515	0.783	0.630	0.545	0.560	0.500	0.532	0.488
h_1	0.544	0.512	0.503	0.501	0.487	0.503	0.475	0.528	0.483	0.474	0.458	0.454	0.452	0.443
h_2	0.813	0.680	0.593	0.580	0.539	0.590	0.534	0.790	0.621	0.538	0.560	0.501	0.542	0.492
D	0.489	0.459	0.456	0.445	0.444	0.466	0.431	0.484	0.419	0.412	0.415	0.411	0.418	0.396
G	0.553	0.520	0.521	0.545	0.555	0.560	0.550	0.545	0.528	0.529	0.533	0.536	0.533	0.524
I	0.839	0.800	0.807	0.852	0.876	0.873	0.869	0.812	0.812	0.819	0.818	0.837	0.827	0.820
E	0.303	0.276	0.271	0.277	0.275	0.286	0.272	0.312	0.281	0.276	0.285	0.274	0.276	0.267
v_1	0.505	0.422	0.374	0.400	0.362	0.417	0.360	0.532	0.448	0.387	0.425	0.363	0.403	0.338
v_2	0.875	0.802	0.789	0.923	0.915	0.917	0.904	0.824	0.832	0.827	0.836	0.806	0.809	0.821
v_3	0.447	0.423	0.428	0.432	0.436	0.458	0.443	0.417	0.407	0.413	0.408	0.411	0.415	0.416
v_4	0.273	0.242	0.229	0.249	0.253	0.252	0.250	0.281	0.252	0.244	0.261	0.265	0.233	0.255
h_1	0.246	0.207	0.194	0.203	0.182	0.205	0.173	0.272	0.208	0.198	0.212	0.179	0.204	0.172
h_2	0.518	0.516	0.512	0.525	0.531	0.516	0.531	0.544	0.587	0.583	0.546	0.566	0.580	0.532
h_3	0.132	0.115	0.109	0.115	0.109	0.119	0.119	0.129	0.110	0.105	0.109	0.104	0.107	0.103
G	0.443	0.396	0.385	0.380	0.360	0.376	0.363	0.419	0.399	0.388	0.366	0.356	0.360	0.358
I	0.306	0.294	0.296	0.305	0.306	0.304	0.311	0.292	0.294	0.299	0.295	0.310	0.294	0.313
E	0.562	0.485	0.464	0.445	0.408	0.438	0.408	0.531	0.490	0.464	0.427	0.396	0.417	0.397
v_1	0.840	0.706	0.599	0.566	0.507	0.579	0.485	0.852	0.717	0.597	0.548	0.496	0.554	0.461
v_2	0.519	0.453	0.488	0.462	0.474	0.454	0.467	0.494	0.438	0.467	0.428	0.409	0.395	0.423
v_3	0.096	0.093	0.090	0.092	0.092	0.089	0.088	0.102	0.103	0.100	0.092	0.102	0.094	0.098
v_4	0.649	0.559	0.624	0.563	0.562	0.569	0.552	0.652	0.587	0.641	0.582	0.577	0.569	0.610
v_5	1.208	1.033	0.819	0.862	0.711	0.825	0.723	1.158	1.050	0.828	0.802	0.649	0.766	0.672
h_1	0.674	0.566	0.483	0.469	0.399	0.458	0.399	0.580	0.548	0.476	0.426	0.387	0.409	0.381
h_2	0.297	0.209	0.167	0.202	0.156	0.180	0.155	0.277	0.222	0.174	0.173	0.150	0.169	0.154
G	0.053	0.048	0.048	0.045	0.043	0.045	0.042	0.054	0.048	0.047	0.046	0.044	0.046	0.044
I	0.038	0.037	0.039	0.037	0.035	0.036	0.035	0.040	0.037	0.039	0.038	0.037	0.036	0.036
E	0.066	0.057	0.055	0.052	0.049	0.052	0.049	0.066	0.057	0.054	0.054	0.050	0.054	0.051
v_1	0.051	0.045	0.041	0.038	0.034	0.041	0.035	0.057	0.044	0.039	0.039	0.039	0.041	0.038
v_2	0.050	0.046	0.046	0.047	0.045	0.045	0.045	0.050	0.046	0.047	0.046	0.045	0.048	0.048
v_3	0.185	0.160	0.133	0.131	0.112	0.126	0.108	0.181	0.154	0.125	0.130	0.106	0.127	0.108
h_1	0.101	0.085	0.078	0.074	0.070	0.074	0.070	0.100	0.084	0.076	0.077	0.072	0.076	0.070
h_2	0.047	0.041	0.042	0.041	0.039	0.040	0.037	0.044	0.044	0.045	0.039	0.039	0.042	0.041
h_3	0.052	0.042	0.042	0.043	0.039	0.043	0.039	0.053	0.044	0.043	0.043	0.040	0.044	0.043

Table C.1: Mean squared error for the lifting estimations for regular and clustered points when $\gamma = 0, 0.25$. The rows denote the different spatial parts of the region, and the columns give the results for different weight methods. If the parameter $\gamma < 1$ it indicates the incremental magnitudes of inhibition or repulsion, and clustering if $\gamma > 1$. Results are given for Dopper, Heavisine, Blocks, Bumps and Maartenfunc test functions from top to bottom panel respectively. MSE calculated globally is denoted as (G), interior (I), edge (E) of the region, and the vertical (v), horizontal (h), and diagonal (D) transects.

	$\gamma = 0.5$							$\gamma = 0.75$						
	Conv.	Unit	Doub.	B_0	A_0	$B^\star$	$A^\star$	Conv.	Unit	Doub.	B_0	A_0	$B^\star$	$A^\star$
G	0.046	0.044	0.045	0.045	0.045	0.044	0.045	0.045	0.044	0.045	0.044	0.042	0.042	0.043
I	0.039	0.039	0.040	0.042	0.041	0.040	0.041	0.038	0.039	0.040	0.039	0.039	0.039	0.039
E	0.053	0.049	0.049	0.049	0.048	0.047	0.048	0.052	0.048	0.049	0.047	0.045	0.045	0.046
v_1	0.106	0.094	0.092	0.090	0.087	0.089	0.087	0.105	0.088	0.086	0.086	0.084	0.086	0.086
v_2	0.015	0.014	0.013	0.013	0.011	0.012	0.011	0.015	0.014	0.013	0.012	0.012	0.012	0.011
h_1	0.106	0.097	0.096	0.087	0.086	0.089	0.085	0.111	0.098	0.097	0.089	0.081	0.085	0.080
h_2	0.014	0.013	0.012	0.013	0.011	0.012	0.012	0.016	0.013	0.012	0.013	0.012	0.012	0.012
D	0.064	0.066	0.067	0.062	0.059	0.056	0.061	0.064	0.058	0.060	0.057	0.061	0.058	0.057
G	0.426	0.399	0.401	0.386	0.381	0.384	0.375	0.387	0.350	0.352	0.345	0.336	0.347	0.338
I	0.391	0.392	0.404	0.392	0.390	0.390	0.389	0.354	0.347	0.357	0.349	0.353	0.350	0.353
E	0.456	0.405	0.399	0.379	0.372	0.379	0.362	0.418	0.353	0.348	0.341	0.321	0.344	0.323
v_1	0.514	0.462	0.451	0.431	0.404	0.418	0.403	0.462	0.414	0.408	0.369	0.345	0.371	0.334
v_2	0.730	0.590	0.510	0.512	0.459	0.496	0.459	0.728	0.520	0.454	0.483	0.410	0.484	0.406
h_1	0.540	0.459	0.451	0.435	0.412	0.422	0.415	0.492	0.426	0.416	0.387	0.368	0.384	0.363
h_2	0.750	0.600	0.522	0.523	0.460	0.525	0.438	0.710	0.521	0.457	0.467	0.400	0.470	0.401
D	0.451	0.388	0.390	0.367	0.361	0.366	0.357	0.407	0.345	0.347	0.364	0.350	0.353	0.330
G	0.549	0.537	0.539	0.533	0.527	0.542	0.539	0.548	0.521	0.522	0.528	0.538	0.524	0.538
I	0.815	0.819	0.827	0.820	0.813	0.835	0.832	0.798	0.783	0.794	0.803	0.826	0.798	0.826
E	0.308	0.283	0.279	0.275	0.270	0.277	0.276	0.315	0.277	0.269	0.272	0.269	0.269	0.270
v_1	0.525	0.432	0.375	0.395	0.367	0.387	0.359	0.550	0.453	0.383	0.424	0.381	0.414	0.361
v_2	0.824	0.843	0.836	0.819	0.804	0.845	0.814	0.813	0.788	0.781	0.794	0.847	0.812	0.838
v_3	0.389	0.393	0.399	0.384	0.393	0.403	0.392	0.393	0.371	0.362	0.380	0.390	0.377	0.395
v_4	0.300	0.264	0.249	0.261	0.253	0.267	0.256	0.309	0.241	0.223	0.236	0.226	0.240	0.241
h_1	0.260	0.216	0.207	0.203	0.196	0.201	0.208	0.262	0.216	0.197	0.216	0.212	0.207	0.199
h_2	0.545	0.549	0.553	0.551	0.567	0.543	0.563	0.555	0.521	0.508	0.535	0.572	0.522	0.551
h_3	0.127	0.111	0.108	0.110	0.102	0.110	0.104	0.141	0.109	0.105	0.121	0.115	0.121	0.113
G	0.416	0.386	0.381	0.367	0.351	0.357	0.358	0.421	0.374	0.371	0.360	0.348	0.352	0.338
I	0.291	0.296	0.300	0.297	0.293	0.297	0.304	0.296	0.296	0.302	0.296	0.302	0.287	0.293
E	0.529	0.468	0.454	0.430	0.402	0.411	0.407	0.536	0.447	0.436	0.420	0.391	0.413	0.380
v_1	0.846	0.687	0.606	0.578	0.504	0.549	0.510	0.872	0.657	0.565	0.588	0.514	0.569	0.495
v_2	0.566	0.502	0.552	0.477	0.468	0.496	0.459	0.556	0.423	0.449	0.491	0.446	0.462	0.448
v_3	0.094	0.092	0.091	0.091	0.092	0.090	0.101	0.099	0.087	0.087	0.099	0.095	0.094	0.085
v_4	0.649	0.606	0.687	0.619	0.583	0.579	0.570	0.670	0.593	0.670	0.571	0.535	0.550	0.519
v_5	1.142	0.964	0.781	0.817	0.713	0.769	0.720	1.193	0.964	0.799	0.777	0.682	0.787	0.674
h_1	0.561	0.514	0.441	0.414	0.370	0.388	0.373	0.600	0.494	0.443	0.435	0.364	0.426	0.368
h_2	0.285	0.203	0.170	0.197	0.156	0.168	0.166	0.300	0.220	0.190	0.193	0.168	0.179	0.154
G	0.052	0.049	0.049	0.046	0.045	0.046	0.044	0.055	0.048	0.049	0.047	0.046	0.047	0.046
I	0.039	0.039	0.041	0.039	0.038	0.038	0.038	0.039	0.039	0.041	0.038	0.039	0.038	0.038
E	0.065	0.059	0.057	0.053	0.052	0.054	0.050	0.069	0.057	0.055	0.056	0.053	0.055	0.053
v_1	0.053	0.050	0.045	0.042	0.040	0.042	0.039	0.057	0.044	0.042	0.046	0.044	0.043	0.044
v_2	0.052	0.049	0.049	0.047	0.051	0.048	0.048	0.051	0.049	0.050	0.051	0.048	0.049	0.048
v_3	0.180	0.150	0.122	0.138	0.121	0.137	0.112	0.191	0.148	0.124	0.135	0.119	0.134	0.120
h_1	0.097	0.085	0.079	0.073	0.070	0.076	0.068	0.106	0.086	0.081	0.080	0.075	0.074	0.074
h_2	0.044	0.043	0.044	0.040	0.038	0.038	0.040	0.046	0.045	0.045	0.038	0.043	0.042	0.041
h_3	0.048	0.046	0.046	0.041	0.039	0.042	0.040	0.053	0.042	0.041	0.044	0.041	0.043	0.042

Table C.2: Mean squared error for the lifting estimations for regular and clustered points when $\gamma = 0.5, 0.75$. The rows denote the different spatial parts of the region, and the columns give the results for different weight methods. If the parameter $\gamma < 1$ it indicates the incremental magnitudes of inhibition or repulsion, and clustering if $\gamma > 1$. Results are given for Dopper, Heavisine, Blocks, Bumps and Maartenfunc test functions from top to bottom panel respectively. MSE calculated globally is denoted as (G), interior (I), edge (E) of the region, and the vertical (v), horizontal (h), and diagonal (D) transects.

	$\gamma = 1$							$\gamma = 1.25$						
	Conv.	Unit	Doub.	B_0	A_0	$B^\star$	$A^\star$	Conv.	Unit	Doub.	B_0	A_0	$B^\star$	$A^\star$
G	0.046	0.043	0.045	0.043	0.043	0.043	0.043	0.044	0.044	0.045	0.043	0.043	0.042	0.043
I	0.040	0.040	0.042	0.038	0.040	0.038	0.040	0.037	0.040	0.042	0.038	0.039	0.038	0.039
E	0.052	0.047	0.048	0.048	0.045	0.047	0.047	0.051	0.047	0.048	0.047	0.046	0.047	0.046
v_1	0.110	0.088	0.088	0.085	0.081	0.084	0.085	0.111	0.091	0.088	0.081	0.081	0.083	0.083
v_2	0.016	0.013	0.012	0.013	0.012	0.013	0.012	0.015	0.014	0.013	0.013	0.013	0.013	0.013
h_1	0.113	0.096	0.094	0.090	0.080	0.086	0.088	0.106	0.093	0.090	0.090	0.088	0.088	0.085
h_2	0.015	0.013	0.012	0.013	0.012	0.012	0.013	0.016	0.013	0.012	0.013	0.012	0.013	0.013
D	0.069	0.062	0.064	0.065	0.059	0.062	0.064	0.064	0.052	0.054	0.065	0.066	0.064	0.065
G	0.346	0.311	0.312	0.304	0.296	0.295	0.292	0.311	0.285	0.289	0.274	0.269	0.265	0.260
I	0.311	0.314	0.322	0.308	0.306	0.302	0.304	0.281	0.288	0.297	0.282	0.283	0.269	0.273
E	0.379	0.308	0.302	0.301	0.285	0.288	0.280	0.341	0.283	0.281	0.266	0.255	0.260	0.248
v_1	0.451	0.356	0.346	0.330	0.305	0.317	0.298	0.401	0.331	0.328	0.286	0.265	0.294	0.264
v_2	0.683	0.482	0.425	0.428	0.373	0.406	0.362	0.606	0.432	0.378	0.393	0.321	0.350	0.315
h_1	0.426	0.367	0.358	0.333	0.315	0.317	0.304	0.382	0.319	0.318	0.302	0.272	0.289	0.276
h_2	0.682	0.483	0.420	0.436	0.370	0.404	0.366	0.628	0.439	0.387	0.377	0.345	0.379	0.321
D	0.377	0.307	0.307	0.307	0.283	0.297	0.281	0.357	0.267	0.269	0.270	0.280	0.273	0.274
G	0.535	0.506	0.509	0.533	0.538	0.527	0.523	0.523	0.489	0.492	0.506	0.507	0.510	0.509
I	0.769	0.746	0.753	0.801	0.813	0.791	0.791	0.739	0.718	0.728	0.750	0.752	0.753	0.755
E	0.311	0.276	0.275	0.276	0.274	0.274	0.266	0.312	0.265	0.261	0.268	0.268	0.272	0.269
v_1	0.594	0.468	0.415	0.414	0.382	0.446	0.378	0.574	0.435	0.374	0.436	0.375	0.430	0.381
v_2	0.771	0.774	0.759	0.787	0.782	0.746	0.785	0.762	0.768	0.756	0.783	0.789	0.735	0.771
v_3	0.390	0.377	0.388	0.405	0.396	0.399	0.382	0.379	0.367	0.373	0.367	0.404	0.401	0.378
v_4	0.313	0.277	0.262	0.261	0.252	0.246	0.240	0.297	0.251	0.241	0.252	0.239	0.244	0.248
h_1	0.284	0.197	0.193	0.203	0.220	0.209	0.204	0.286	0.193	0.177	0.218	0.202	0.217	0.214
h_2	0.511	0.507	0.498	0.547	0.548	0.543	0.532	0.566	0.518	0.518	0.534	0.551	0.540	0.539
h_3	0.132	0.124	0.121	0.115	0.107	0.108	0.105	0.125	0.108	0.107	0.105	0.110	0.110	0.105
G	0.415	0.370	0.367	0.351	0.347	0.345	0.346	0.406	0.361	0.362	0.351	0.342	0.343	0.339
I	0.295	0.294	0.301	0.293	0.303	0.288	0.301	0.284	0.279	0.288	0.286	0.293	0.278	0.286
E	0.531	0.443	0.431	0.408	0.390	0.399	0.389	0.525	0.442	0.434	0.415	0.389	0.406	0.391
v_1	0.870	0.644	0.563	0.559	0.499	0.548	0.521	0.896	0.670	0.603	0.545	0.490	0.554	0.520
v_2	0.532	0.448	0.464	0.439	0.431	0.427	0.443	0.532	0.460	0.491	0.463	0.457	0.455	0.458
v_3	0.102	0.098	0.098	0.101	0.106	0.094	0.106	0.101	0.098	0.096	0.102	0.101	0.096	0.103
v_4	0.740	0.618	0.675	0.581	0.601	0.584	0.577	0.708	0.578	0.643	0.617	0.578	0.591	0.583
v_5	1.174	0.959	0.802	0.784	0.666	0.791	0.674	1.160	0.967	0.799	0.797	0.649	0.790	0.672
h_1	0.637	0.501	0.456	0.416	0.395	0.394	0.375	0.588	0.515	0.468	0.436	0.408	0.408	0.392
h_2	0.322	0.243	0.200	0.198	0.176	0.203	0.169	0.350	0.232	0.201	0.214	0.193	0.206	0.189
G	0.054	0.050	0.050	0.049	0.048	0.049	0.047	0.055	0.050	0.050	0.047	0.046	0.047	0.046
I	0.039	0.040	0.043	0.039	0.039	0.040	0.039	0.041	0.040	0.043	0.039	0.039	0.040	0.039
E	0.068	0.058	0.057	0.059	0.055	0.057	0.054	0.070	0.059	0.058	0.054	0.053	0.055	0.053
v_1	0.062	0.046	0.042	0.049	0.043	0.046	0.043	0.062	0.049	0.046	0.046	0.043	0.046	0.044
v_2	0.056	0.051	0.052	0.053	0.053	0.052	0.052	0.052	0.054	0.054	0.049	0.051	0.048	0.054
v_3	0.185	0.155	0.131	0.146	0.127	0.141	0.120	0.192	0.158	0.134	0.136	0.126	0.137	0.119
h_1	0.104	0.085	0.081	0.079	0.075	0.080	0.074	0.107	0.084	0.079	0.074	0.075	0.074	0.072
h_2	0.043	0.047	0.047	0.045	0.041	0.041	0.041	0.045	0.044	0.044	0.043	0.042	0.043	0.041
h_3	0.051	0.043	0.043	0.047	0.045	0.046	0.045	0.050	0.043	0.042	0.042	0.040	0.043	0.040

Table C.3: Mean squared error for the lifting estimations for regular and clustered points when $\gamma = 1, 1.25$. The rows denote the different spatial parts of the region, and the columns give the results for different weight methods. If the parameter $\gamma < 1$ it indicates the incremental magnitudes of inhibition or repulsion, and clustering if $\gamma > 1$. Results are given for Dopper, Heavisine, Blocks, Bumps and Maartenfunc test functions from top to bottom panel respectively. MSE calculated globally is denoted as (G), interior (I), edge (E) of the region, and the vertical (v), horizontal (h), and diagonal (D) transects.

	$\gamma = 1.5$							$\gamma = 2$						
	Conv.	Unit	Doub.	B_0	A_0	$B^\star$	$A^\star$	Conv.	Unit	Doub.	B_0	A_0	$B^\star$	$A^\star$
G	0.045	0.043	0.044	0.043	0.044	0.042	0.043	0.045	0.043	0.042	0.046	0.046	0.042	0.042
I	0.038	0.040	0.041	0.039	0.041	0.038	0.039	0.038	0.038	0.040	0.041	0.043	0.038	0.038
E	0.051	0.047	0.047	0.048	0.047	0.046	0.046	0.052	0.048	0.045	0.050	0.049	0.046	0.046
v_1	0.105	0.088	0.088	0.088	0.081	0.078	0.082	0.097	0.091	0.089	0.092	0.087	0.073	0.076
v_2	0.015	0.012	0.013	0.012	0.012	0.012	0.012	0.016	0.014	0.013	0.014	0.014	0.013	0.012
h_1	0.108	0.097	0.090	0.093	0.088	0.089	0.090	0.107	0.096	0.091	0.094	0.088	0.088	0.088
h_2	0.015	0.013	0.012	0.012	0.013	0.013	0.013	0.016	0.013	0.012	0.014	0.013	0.013	0.013
D	0.063	0.065	0.064	0.063	0.065	0.055	0.062	0.074	0.061	0.061	0.068	0.070	0.068	0.061
G	0.298	0.263	0.269	0.252	0.251	0.247	0.249	0.281	0.244	0.249	0.265	0.267	0.233	0.229
I	0.267	0.270	0.278	0.258	0.264	0.252	0.261	0.249	0.244	0.257	0.268	0.270	0.238	0.244
E	0.329	0.257	0.261	0.247	0.238	0.242	0.237	0.312	0.243	0.241	0.262	0.263	0.228	0.214
v_1	0.397	0.315	0.309	0.280	0.262	0.261	0.257	0.380	0.283	0.285	0.277	0.270	0.256	0.223
v_2	0.606	0.405	0.368	0.354	0.306	0.333	0.294	0.558	0.385	0.343	0.351	0.327	0.318	0.268
h_1	0.382	0.292	0.297	0.254	0.254	0.265	0.255	0.364	0.276	0.269	0.280	0.272	0.253	0.223
h_2	0.601	0.408	0.357	0.362	0.313	0.350	0.300	0.584	0.404	0.336	0.360	0.348	0.326	0.269
D	0.320	0.259	0.264	0.252	0.239	0.247	0.241	0.311	0.228	0.240	0.265	0.256	0.249	0.242
G	0.530	0.519	0.495	0.514	0.517	0.503	0.509	0.508	0.498	0.494	0.494	0.496	0.491	0.494
I	0.744	0.761	0.730	0.761	0.768	0.738	0.751	0.703	0.731	0.725	0.715	0.726	0.719	0.722
E	0.319	0.279	0.263	0.270	0.269	0.271	0.270	0.313	0.264	0.260	0.273	0.266	0.263	0.267
v_1	0.592	0.456	0.393	0.430	0.387	0.413	0.398	0.555	0.444	0.382	0.407	0.358	0.388	0.374
v_2	0.734	0.746	0.704	0.747	0.761	0.742	0.744	0.761	0.639	0.659	0.716	0.709	0.752	0.776
v_3	0.379	0.367	0.341	0.368	0.384	0.370	0.364	0.381	0.360	0.369	0.373	0.375	0.344	0.360
v_4	0.300	0.262	0.223	0.231	0.248	0.246	0.235	0.288	0.231	0.229	0.236	0.230	0.234	0.223
h_1	0.300	0.224	0.185	0.210	0.210	0.236	0.233	0.285	0.209	0.185	0.224	0.215	0.210	0.212
h_2	0.496	0.503	0.506	0.522	0.532	0.479	0.476	0.489	0.466	0.518	0.490	0.481	0.459	0.457
h_3	0.123	0.112	0.118	0.111	0.109	0.104	0.107	0.121	0.102	0.101	0.115	0.114	0.109	0.104
G	0.408	0.366	0.375	0.354	0.340	0.350	0.337	0.400	0.362	0.360	0.367	0.370	0.338	0.334
I	0.283	0.290	0.311	0.294	0.290	0.284	0.285	0.275	0.299	0.301	0.305	0.312	0.280	0.288
E	0.533	0.441	0.437	0.414	0.389	0.416	0.389	0.524	0.425	0.419	0.429	0.427	0.396	0.380
v_1	0.909	0.675	0.667	0.615	0.530	0.546	0.542	0.931	0.706	0.596	0.612	0.592	0.544	0.532
v_2	0.539	0.475	0.470	0.489	0.473	0.454	0.459	0.523	0.439	0.473	0.480	0.498	0.427	0.419
v_3	0.096	0.097	0.099	0.102	0.105	0.104	0.099	0.109	0.109	0.103	0.110	0.124	0.114	0.106
v_4	0.713	0.552	0.637	0.544	0.545	0.608	0.567	0.700	0.556	0.621	0.551	0.575	0.543	0.550
v_5	1.192	0.992	0.810	0.801	0.682	0.849	0.690	1.243	0.936	0.819	0.819	0.750	0.760	0.653
h_1	0.603	0.499	0.486	0.398	0.382	0.414	0.365	0.581	0.545	0.469	0.444	0.463	0.399	0.362
h_2	0.322	0.212	0.198	0.206	0.179	0.223	0.178	0.333	0.256	0.218	0.214	0.212	0.204	0.184
G	0.056	0.049	0.051	0.048	0.047	0.048	0.047	0.058	0.050	0.052	0.051	0.050	0.048	0.047
I	0.041	0.039	0.045	0.040	0.040	0.041	0.040	0.042	0.040	0.045	0.043	0.043	0.041	0.040
E	0.071	0.059	0.057	0.056	0.054	0.054	0.054	0.074	0.059	0.058	0.059	0.058	0.055	0.054
v_1	0.064	0.047	0.045	0.045	0.042	0.047	0.045	0.067	0.050	0.048	0.052	0.048	0.045	0.044
v_2	0.055	0.049	0.050	0.051	0.054	0.054	0.051	0.054	0.056	0.053	0.058	0.056	0.054	0.052
v_3	0.195	0.158	0.127	0.140	0.133	0.133	0.128	0.210	0.164	0.139	0.157	0.150	0.145	0.128
h_1	0.111	0.084	0.076	0.078	0.071	0.073	0.076	0.120	0.084	0.081	0.094	0.087	0.081	0.079
h_2	0.047	0.043	0.050	0.043	0.043	0.047	0.046	0.047	0.046	0.050	0.047	0.045	0.044	0.041
h_3	0.049	0.043	0.045	0.040	0.042	0.040	0.041	0.048	0.047	0.045	0.047	0.046	0.039	0.042

Table C.4: Mean squared error for the lifting estimations for regular and clustered points when $\gamma = 1.5, 2$. The rows denote the different spatial parts of the region, and the columns give the results for different weight methods. If the parameter $\gamma < 1$ it indicates the incremental magnitudes of inhibition or repulsion, and clustering if $\gamma > 1$. Results are given for Dopper, Heavisine, Blocks, Bumps and Maartenfunc test functions from top to bottom panel respectively. MSE calculated globally is denoted as (G), interior (I), edge (E) of the region, and the vertical (v), horizontal (h), and diagonal (D) transects.

	$\gamma = 3$						
	Conv.	Unit	Doub.	B_0	A_0	$B^\star$	$A^\star$
G	0.045	0.042	0.043	0.046	0.046	0.042	0.042
I	0.037	0.037	0.039	0.041	0.042	0.038	0.037
E	0.052	0.047	0.048	0.051	0.051	0.047	0.046
v_1	0.102	0.087	0.088	0.092	0.092	0.091	0.086
v_2	0.016	0.014	0.013	0.014	0.013	0.014	0.013
h_1	0.115	0.090	0.096	0.106	0.106	0.091	0.086
h_2	0.016	0.014	0.012	0.013	0.014	0.013	0.013
D	0.069	0.059	0.063	0.075	0.073	0.058	0.057
G	0.267	0.217	0.222	0.260	0.262	0.223	0.228
I	0.237	0.220	0.230	0.264	0.267	0.227	0.237
E	0.298	0.213	0.214	0.256	0.256	0.218	0.218
v_1	0.351	0.234	0.236	0.261	0.270	0.239	0.228
v_2	0.548	0.348	0.320	0.335	0.317	0.297	0.263
h_1	0.364	0.255	0.256	0.277	0.297	0.251	0.243
h_2	0.565	0.349	0.316	0.358	0.350	0.314	0.277
D	0.288	0.212	0.210	0.254	0.247	0.220	0.234
G	0.505	0.509	0.492	0.495	0.501	0.491	0.503
I	0.699	0.740	0.709	0.711	0.722	0.718	0.741
E	0.308	0.270	0.269	0.277	0.277	0.263	0.262
v_1	0.544	0.426	0.394	0.396	0.380	0.391	0.378
v_2	0.632	0.752	0.718	0.689	0.689	0.653	0.682
v_3	0.379	0.395	0.380	0.390	0.405	0.346	0.372
v_4	0.285	0.238	0.221	0.267	0.252	0.244	0.200
h_1	0.264	0.221	0.211	0.221	0.219	0.212	0.191
h_2	0.459	0.491	0.492	0.505	0.496	0.444	0.452
h_3	0.118	0.110	0.116	0.107	0.111	0.097	0.098
G	0.408	0.345	0.360	0.368	0.370	0.339	0.331
I	0.287	0.278	0.304	0.300	0.308	0.279	0.282
E	0.530	0.415	0.418	0.436	0.432	0.399	0.381
v_1	0.984	0.624	0.591	0.689	0.632	0.578	0.509
v_2	0.528	0.450	0.456	0.475	0.505	0.448	0.425
v_3	0.106	0.096	0.104	0.101	0.107	0.104	0.105
v_4	0.752	0.525	0.570	0.617	0.625	0.580	0.572
v_5	1.166	0.955	0.907	0.854	0.805	0.763	0.685
h_1	0.560	0.486	0.454	0.485	0.462	0.444	0.404
h_2	0.382	0.257	0.230	0.279	0.265	0.196	0.180
G	0.059	0.050	0.053	0.053	0.052	0.048	0.047
I	0.042	0.041	0.046	0.044	0.044	0.040	0.040
E	0.076	0.059	0.059	0.062	0.061	0.056	0.055
v_1	0.064	0.050	0.046	0.060	0.056	0.046	0.045
v_2	0.059	0.053	0.054	0.053	0.053	0.050	0.050
v_3	0.207	0.168	0.152	0.169	0.157	0.146	0.127
h_1	0.123	0.079	0.080	0.090	0.087	0.078	0.076
h_2	0.052	0.045	0.047	0.050	0.049	0.045	0.045
h_3	0.055	0.049	0.052	0.052	0.053	0.045	0.040

Table C.5: Mean squared error for the lifting estimations for regular and clustered points when $\gamma = 3$. The rows denote the different spatial parts of the region, and the columns give the results for different weight methods. If the parameter $\gamma < 1$ it indicates the incremental magnitudes of inhibition or repulsion, and clustering if $\gamma > 1$. Results are given for Dopper, Heavisine, Blocks, Bumps and Maartenfunc test functions from top to bottom panel respectively. MSE calculated globally is denoted as (G), interior (I), edge (E) of the region, and the vertical (v), horizontal (h), and diagonal (D) transects.

References

Akaike, H. (1987). Factor analysis and AIC. In *Selected Papers of Hirotugu Akaike*, 371–386, Springer. 53

Antoniadis, A., Bigot, J. & Sapatinas, T. (2001). Wavelet estimators in nonparametric regression: a comparative simulation study. *Journal of Statistical Software*, **6**, 1–83. 116

Arvanitakis, G. (2014). Distribution of the number of Poisson points in Poisson Voronoi tessellation. *Tech. Rep. RR-15-304*. 17

Baccelli, F. & Błaszczyszyn, B. (2001). On a coverage process ranging from the Boolean model to the Poisson–Voronoi tessellation with applications to wireless communications. *Advances in Applied Probability*, **33**, 293–323. 14

Baddeley, A., Rubak, E. & Turner, R. (2015). *Spatial point patterns: methodology and applications with R*. CRC press, New York. 85, 86

Besag, J. (1977). Discussion on Dr Ripley's Paper. *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**, 192–212. 94

Bivand, R. & Rundel, C. (2020). *rgeos: Interface to Geometry Engine - Open Source ('GEOS')*. R package version 0.5-5. 111

Brakke, K.A. (1987). Statistics of random plane Voronoi tessellations. *Department of Mathematical Sciences, Susquehanna University (Manuscript 1987a)*. 18

Cai, T.T. (1999). Adaptive wavelet estimation: a block thresholding and oracle inequality approach. *Annals of Statistics*, **27**, 898–924. 115

Cai, T.T. (2002). On block thresholding in wavelet regression: Adaptivity, block size, and threshold level. *Statistica Sinica*, **2**, 1241–1273. 115

Cai, T.T. & Silverman, B.W. (2001). Incorporating information on neighbouring coefficients into wavelet estimation. *Sankhyā: The Indian Journal of Statistics, Series B*, **63**, 127–148. 115

Christensen, R. (1991). *Linear models for multivariate, time series, and spatial data*, vol. 1. Springer, New York. 147

Claire, M. & Neocleous, T. (2019). Flexible regression lecture notes. *Academy for PhD training in Statistics, Oxford University*. 52

Claypoole, R.L., Baraniuk, R.G. & Nowak, R.D. (1998). Adaptive wavelet transforms via lifting. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, vol. 3, 1513–1516, IEEE. 102

Cox, D.R. & Isham, V. (1980). *Point processes*, vol. 12. CRC Press, Boca Raton. 85

Crain, I.K. (1978). The Monte-Carlo generation of random polygons. *Computers & Geosciences*, **4**, 131–141. 15

Cressie, N. (2015). *Statistics for Spatial Data*. John Wiley & Sons, New York. 85, 147

Cressie, N. & Wikle, C.K. (2015). *Statistics for spatio-temporal data*. John Wiley & Sons. 85

Dalling, J., John, R., Harms, K., Stallard, R. & Yavitt, J. (2021). Project: Effects of soil-borne resources on the structure and dynamics of lowland tropical forests. Funding: NSF DEB021104,021115, 0212284,0212818, OISE 0314581, STRI Soils Initiative and CTFS. Note: Thanks to Paolo Segre and Juan Di Trani for assistance in the field. 92

Daubechies, I. (1992). *Ten lectures on wavelets*. SIAM, Philadelphia. 100

Diggle, P. (1983). *Statistical analysis of spatial point patterns*. Mathematics in biology, Academic Press, London. 85

Diggle, P. (1985). A kernel method for smoothing point process data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **34**, 138–147. 87

Donoho, D.L. & Johnstone, I.M. (1995). Adapting to unknown smoothness via wavelet shrinkage. *Journal of The American Statistical Association*, **90**, 1200–1224. 115

Donoho, D.L. & Johnstone, J.M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, **81**, 425–455. 10, 112, 113, 114, 122

Donoho, D.L., Johnstone, I.M., Kerkyacharian, G. & Picard, D. (1995). Wavelet shrinkage: asymptopia? *Journal of the Royal Statistical Society: Series B (Methodological)*, **57**, 301–337. 10, 112, 113

Finney, J. (1970). Random packings and the structure of simple liquids. I. The geometry of random close packing. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 479–493. 14

Fischer, R. & Miles, R. (1973). The role of spatial pattern in the competition between crop plants and weeds. A theoretical analysis. *Mathematical Biosciences*, **18**, 335–350. 14

Garrett, R.C., Nar, A. & Fisher, T.J. (2021). *ggvoronoi: Voronoi Diagrams and Heatmaps with 'ggplot2'*. R package version 0.8.4. 116

Gebhardt, A., Eglen, S., Zuyev, S. & White, D. (2020). *tripack: Triangulation of Irregularly Spaced Data*. R package version 1.3-9.1. 111

Gelfand, A.E., Diggle, P., Guttorp, P. & Fuentes, M. (2010). *Handbook of spatial statistics*. CRC press. 85

Geyer, C. (1999). Likelihood inference for spatial point processes. In *Stochastic Geometry*, 79–140, Routledge. 6, 85, 87, 137, 154

Gezer, F., Aykroyd, R.G. & Barber, S. (2021). Statistical properties of Poisson-Voronoi tessellation cells in bounded regions. *Journal of Statistical Computation and Simulation*, **91**, 915–933. 13, 44

Gilbert, E. (1962). Random subdivisions of space into crystals. *The Annals of Mathematical Statistics*, **33**, 958–972. 15

Goldstein, J., Haran, M., Simeonov, I., Fricks, J. & Chiaromonte, F. (2015). An attraction–repulsion point process model for respiratory syncytial virus infections. *Biometrics*, **71**, 376–385. 6

Gräler, B., Pebesma, E. & Heuvelink, G. (2016). Spatio-temporal interpolation using gstat. *The R Journal*, **8**, 204–218. 147

Hall, P., Kerkyacharian, G. & Picard, D. (1999). On the minimax optimality of block thresholded wavelet estimators. *Statistica Sinica*, **9**, 33–49. 115

Hastie, T. (2020). *gam: Generalized Additive Models*. R package version 1.20. 53

Hastie, T.J. & Tibshirani, R.J. (1990). *Generalized additive models*. CRC press, Boca Raton. 50, 52

Heaton, T. & Silverman, B. (2008). A wavelet-or lifting-scheme-based imputation method. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **70**, 567–587. 102, 152, 159

Hinde, A. & Miles, R. (1980). Monte Carlo estimates of the distributions of the random polygons of the Voronoi tessellation with respect to a Poisson process. *Journal of Statistical Computation and Simulation*, **10**, 205–223. 16, 29, 32

Ho, T.K. (1995). Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, 278–282, IEEE. 56

Hofner, B., Mayr, A., Robinzonov, N. & Schmid, M. (2014). Model-based boosting in R: A hands-on tutorial using the R package mboost. *Computational statistics*, **29**, 3–35. 56

Icke, V. & Weygaert, R. (1987). Fragmenting the universe. *Astronomy and Astrophysics*, **184**, 16. 14

Illian, J., Penttinen, A., Stoyan, H. & Stoyan, D. (2008). *Statistical analysis and modelling of spatial point patterns*, vol. 70. John Wiley & Sons. 5, 12, 14, 85

James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013). *An Introduction to Statistical Learning*, vol. 112. Springer, New York. 52

Jansen, M. & Bultheel, A. (1999). Smooting irregularly sampled signals using wavelets and cross validation. *Technical Report TW289*. 116

Jansen, M., Nason, G.P. & Silverman, B.W. (2001). Scattered data smoothing by empirical Bayesian shrinkage of second-generation wavelet coefficients. In *Wavelets: Applications in Signal and Image Processing IX*, vol. 4478, 87–98, International Society for Optics and Photonics. 102

Jansen, M., Nason, G.P. & Silverman, B.W. (2009). Multiscale methods for data on graphs and irregular multidimensional situations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **71**, 97–125. 1, 7, 10, 12, 102, 106, 107, 122, 157

Jansen, M.H. & Ooninckx, P.J. (2005). *Second generation wavelets and applications*. Springer Science & Business Media, London. 101

Johnstone, I.M. & Silverman, B.W. (2004). Needles and straw in haystacks: Empirical Bayes estimates of possibly sparse sequences. *The Annals of Statistics*, **32**, 1594–1649. 114

Johnstone, I.M. & Silverman, B.W. (2005a). Empirical bayes selection of wavelet thresholds. *The Annals of Statistics*, **33**, 1700–1752. 114

Johnstone, I.M. & Silverman, B.W. (2005b). Ebayesthresh: R programs for empirical Bayes thresholding. *Journal of Statistical Software*, **12**. 115

Kiang, T. (1966). Random fragmentation in two and three dimensions. *Zeitschrift fur Astrophysik*, **64**, 433. 15, 16, 31

Kiskowski, M.A., Hancock, J.F. & Kenworthy, A.K. (2009). On the use of Ripley's K-function and its derivatives to analyze domain size. *Biophysical Journal*, **97**, 1095–1103. 93

Knight, M. & Nunes, M. (2018). *nlt: A Nondecimated Lifting Transform for Signal Denoising*. R package version 2.2-1. 111

Knight, M.I. & Nason, G.P. (2009). A 'nondecimated' lifting transform. *Statistics and Computing*, **19**, 1–16. 102

Koenker, R. & Machado, J.A. (1999). Goodness of fit and related inference processes for quantile regression. *Journal of the American Statistical Association*, **94**, 1296–1310. 34

Koufos, K. & Dettmann, C.P. (2019). Distribution of cell area in bounded Poisson Voronoi tessellations with application to secure local connectivity. *Journal of Statistical Physics*, **176**, 1–20. xii, 14, 18, 19, 31

Kumar, S. & Kurtz, S.K. (1993). Properties of a two-dimensional Poisson-Voronoi tesselation: a Monte-Carlo study. *Materials Characterization*, **31**, 55–68. 16, 31

Kumar, S., Kurtz, S.K., Banavar, J.R. & Sharma, M. (1992). Properties of a three-dimensional Poisson-Voronoi tesselation: A Monte Carlo study. *Journal of Statistical Physics*, **67**, 523–551. 158

Lazar, E.A., Mason, J.K., MacPherson, R.D. & Srolovitz, D.J. (2013). Statistical topology of three-dimensional Poisson-Voronoi cells and cell boundary networks. *Physical Review E*, **88**, 063309. 158

Liaw, A., Wiener, M. *et al.* (2002). Classification and regression by randomForest. *R news*, **2**, 18–22. 56

Luchnikov, V., Medvedev, N., Naberukhin, Y.I. & Schober, H. (2000). Voronoi-Delaunay analysis of normal modes in a simple model glass. *Physical Review B*, **62**, 3181. 14

Mackay, A. (1972). Stereological characteristics of atomic arrangements in crystals. *Journal of Microscopy*, **95**, 217–227. 14

Mallat, S. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **11**, 674–693. 100, 101

Marra, G. & Wood, S.N. (2011). Practical variable selection for generalized additive models. *Computational Statistics & Data Analysis*, **55**, 2372–2387. 53, 54

Meijering, J. (1953). Interface area, edge length, and number of vertices in crystal aggregates with random nucleation. *Philips Res. Rep.*, **8**, 270–290. 15

Møller, J. (2012). *Lectures on random Voronoi tessellations*, vol. 87. Springer Science & Business Media. 2, 13

Morlini, I. (2006). On multicollinearity and concurvity in some nonlinear multivariate models. *Statistical Methods and Applications*, **15**, 3–26. 48

Muche, L. (1996). Distributional properties of the three-dimensional Poisson Delaunay cell. *Journal of Statistical Physics*, **84**, 147–167. 158

Nason, G. (2008). *Wavelet methods in statistics with R*. Springer Science & Business Media, New York. 101

Nason, G., Jansen, M. & Silverman, B. (2004). *Simulations and examples for multivariate nonparametric regression using lifting*. Technical Report, Department of Mathematics, University of Bristol. 122

Nason, G.P. (1996). Wavelet shrinkage using cross-validation. *Journal of the Royal Statistical Society: Series B (Methodological)*, **58**, 463–479. 116

Nunes, M. & Knight, M. (2018). *adlift: An Adaptive Lifting Scheme Algorithm*. R package version 1.4-1. 111

Nunes, M.A., Knight, M.I. & Nason, G.P. (2006). Adaptive lifting for nonparametric regression. *Statistics and Computing*, **16**, 143–159. 102

Okabe, A., Boots, B., Sugihara, K. & Chiu, S.N. (2000). *Spatial tessellations: concepts and applications of Voronoi diagrams*, vol. 501. John Wiley & Sons, New York. 2, 13, 14

Pebesma, E.J. (2004). Multivariable geostatistics in S: the gstat package. *Computers & geosciences*, **30**, 683–691. 147

Peck, S.J. (2010). *Multiscale spatial imputation applied to crop infestation modelling*. Ph.D. thesis, University of Leeds. 152, 159

Pope, C.A., Gosling, J.P., Barber, S., Johnson, J.S., Yamaguchi, T., Feingold, G. & Blackwell, P.G. (2021). Gaussian process modeling of heterogeneity and discontinuities using Voronoi tessellations. *Technometrics*, **63**, 53–63. 100

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. 111

Ramella, M., Boschin, W., Fadda, D. & Nonino, M. (2001). Finding galaxy clusters using Voronoi tessellations. *Astronomy & Astrophysics*, **368**, 776–786. 14

Ripley, B.D. (1976). The second-order analysis of stationary point processes. *Journal of Applied Probability*, **13**, 255–266. 93

Ripley, B.D. (1977). Modelling spatial patterns. *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**, 172–192. 93

Ripley, B.D. (1988). *Statistical inference for spatial processes*. Cambridge University Press, Cambridge. 85

Ripley, B.D. (2005). *Spatial statistics*, vol. 575. John Wiley & Sons, New York. 12, 14, 85

Schlather, M., Ribeiro Jr, P.J. & Diggle, P.J. (2004). Detecting dependence between marks and locations of marked point processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **66**, 79–93. 157

Schoenberg, F.P., Barr, C. & Seo, J. (2009). The distribution of Voronoi cells generated by Southern California earthquake epicenters. *Environmetrics: The Official Journal of the International Environmetrics Society*, **20**, 159–171. 14, 44

Stacy, E.W. (1962). A generalization of the gamma distribution. *The Annals of Mathematical Statistics*, **33**, 1187–1192. 16

Stacy, E.W. & Mihram, G.A. (1965). Parameter estimation for a generalized gamma distribution. *Technometrics*, **7**, 349–358. 16

Stein, C.M. (1981). Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, **9**, 1135–1151. 115

Strauss, D.J. (1975). A model for clustering. *Biometrika*, **62**, 467–475. 6

Sweldens, W. (1995). Lifting scheme: a new philosophy in biorthogonal wavelet constructions. In *Wavelet Applications in Signal and Image Processing III*, vol. 2569, 68–79, International Society for Optics and Photonics. 100

Sweldens, W. (1996). The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, **3**, 186–200. 100

Sweldens, W. (1998). The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, **29**, 511–546. 10, 100, 102

Tanemura, M. (2003). Statistical distributions of Poisson Voronoi cells in two and three dimensions. *Forma*, **12**, 221–247. xii, 16, 17, 29, 32, 158

Tanemura, M. (2005). Statistical distributions of shape of Poisson Voronoi cells. *Voronoi's Impact on Modern Science. Book III. Proceedings of the 3rd Voronoi Conference on Analytic Number Theory and Spatial Tessellations*. 18

Tanemura, M. & Hasegawa, M. (1980). Geometrical models of territory I. Models for synchronous and asynchronous settlement of territories. *Journal of Theoretical Biology*, **82**, 477–496. 14

Turner, R. (2021). *deldir: Delaunay Triangulation and Dirichlet (Voronoi) Tessellation*. R package version 0.2-10. 111

Vidakovic, B. (1999). *Statistical modeling by wavelets*, vol. 503. John Wiley & Sons, New York. 100

Weaire, D., Kermode, J. & Wejchert, J. (1986). On the distribution of cell areas in a Voronoi network. *Philosophical Magazine B*, **53**, 101–105. 16, 31

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. 21

Wood, S. (2015). Package 'mgcv'. *R package version*, **1**, 29. 54

Wood, S.N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **73**, 3–36. 52

Wood, S.N. (2017). *Generalized additive models: an introduction with R*. CRC press, Boca Raton. 50, 52

Xu, T. & Li, M. (2009). Topological and statistical properties of a constrained voronoi tessellation. *Philosophical Magazine*, **89**, 349–374. 14

Yoshioka, S. & Ikeuchi, S. (1989). The large-scale structure of the universe and the division of space. *The Astrophysical Journal*, **341**, 16–25. 14

Yu, K. & Moyeed, R.A. (2001). Bayesian quantile regression. *Statistics & Probability Letters*, **54**, 437–447. 34