POINT PROCESS AND NONPARAMETRIC MODELLING FOR TOPOLOGICAL DATA ANALYSIS

by

Jordan Anson Schupbach

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

Doctor of Philosophy

in

Statistics

MONTANA STATE UNIVERSITY
Bozeman, Montana

May 2026

DEDICATION

I dedicate this disseration to my family, who have always supported me in my academic pursuits and encouraged me to follow my passions. Their love and guidance have been instrumental in my success, and I am forever grateful for their sacrifices and belief in me. This work is a testament to their influence and the values they instilled in me.

ACKNOWLEDGMENT

I am so very thankful to Dr. John Borkowski, for the many years of mentorship. His kindness and humor has always made learning statistics and doing research a joy. He inspired me to do research and his mountain of knowledge has always given me a new perspective. I thank Dr. John Sheppard, for all of his wisdom and support given to me over the years, and for always pushing me to be better researcher. Our many thought provoking conversations have always been fun and invaluable for building my intuition for research. I thank Dr. Mark Greenwood, for his many years of catching all my mistakes, for being such a great teacher and making the complicated easy to understand, and for always giving kind words of support and encouragement. I thank Dr. Brittany Fasy for her mentorship, for exposing me to new and exciting areas of research, for always pushing me to learn more, and for feeding me when I was hungry. I am thankful to Dr. Stacey Hancock, for her friendship, for being such a great teacher and role model, and for always pushing me to think critically. I am thankful to Dr. Andrew Hoegh, for his practical perspectives and for reminding me of the importance to take time for myself.

I thank Dr. Fernando Guillen and Dr. Megan Higgs, for whom I would not have started my journey in statistics without. I am grateful to have had such great mentorship throughout my time at Montana State University and for all of the educators who have inspired and encouraged me over the years. I thank Dr. Steve Cherry, Dr. Jim Robinson-Cox, and Dr. Warren Esty for teaching me how worthwhile it is to work hard for something and for helping me build a foundation to learn and grow as a student and as a researcher. I thank

ACKNOWLEDGMENT – CONTINUED

Dr. Joe Atwood, Dr. David Ayala, Dr. Katharine Banner, Dr. Anton Beckerman, Dr. Kenneth Bowers, Sandra Bowers, Dr. Quincy Brown, Corinne Casolara, Dr. Breschine Cummins, Dr. Lisa Davis, Dr. Jack Dockery, Dr. Robert Fleck, Dr. Sharon Fox, Dr. Tomáš Gedeon, Dr. Lukas Geyer, Dr. Gregory Gilpin, Charles Godwin, Dr. Ryan Grady, Dr. Kathi Irvine, Dr. Peter Lawson, Dr. Lillian Lin, Michael Malesich, Dr. Facundo Memoli, Dr. Atish Mitra, Dr. Shinjini Nandi, Dr. Dominic Parker, Dr. Jason Pearcy, Dr. Demi Qin, Dr. Randy Rucker, Dr. Wendy Stock, Dr. Chris Stoddard, Dr. Brian Summa, Dr. Miles Watts, Dr. Carola Wenk, and Dr. Binhai Zhu, all of whom have been great mentors and friends, and have helped me grow as a student, as a researcher, and as a person. I am thankful to Dr. Elizabeth Burroughs, Dr. John Paxton, Katie Sutich, Jane Crawford, Madison Maus, and Molly Patten, for keeping the ship afloat and making the Math and Computer Science departments a fun place to learn and grow.

I am thankful to my friend, Dr. David Millman, for his mentorship and for all the adventures we have had together getting lost in the woods. I thank my friends Dr. Amy Peerlinck and Dr. Chris Barbour, for all the fun times and keeping me sane during the stressful times. I thank all of my NISL lab mates, COMPTAG lab mates, and my Math Department friends. We have shared so many good times together. I thank my friends, Scotty Beathe, Grant Brunton, David Elsea, Alex Hoerger, Andy Russo, Dr. Matthew Skuntz and Tony Varillano. I am grateful for all of adventures and misadventures we have had together over the years. I thank Jeffery Nichols, for being such a good friend for all these years and

for always being there for me. I thank Tyler Bowman, for so many years of friendship. You are missed, my dear friend.

Most of all, I thank my family, for their unconditional love and support. I thank my parents, for raising me and teaching me the values that have helped me get to where I am today, for always being there to support me, and for always giving me a place to call home. I thank my grandparents, for teaching me, nurturing my curiosity, and giving me the love and appreciation for all the experiences this world has to offer. I thank my brothers, for always being there to support me and inspiring me to be the best version of myself.

Funding Acknowledgement

The research presented in this dissertation was partially supported by the National Science Foundation under grant nos. 1664858, 1657553, and 1955925, and by the SBIR/STTR program under grant no. N68335-21-C-0591.

TABLE OF CONTENTS

TABLE OF CONTENTS – CONTINUED

TABLE OF CONTENTS – CONTINUED

LIST OF TABLES

Table Page

LIST OF FIGURES

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF ALGORITHMS

Algorithm Page

ABSTRACT

Topological data analysis (TDA) is a relatively new interdisciplinary field that seeks to represent the shape of data using tools from algebraic topology. The field has developed several methods with the ability to represent and summarize complex structural information present in large and high-dimensional datasets. However, methods for analyzing these representations under non-trivial sampling designs are few and rarely employed in practice. In this dissertation, we propose methods to conduct estimation and hypothesis testing in the setting of hierarchical sampling designs for persistence intensity functions, a commonly used topological descriptor in TDA. In particular, we propose to use mixed-curve models to model collections of persistence intensity functions as replicated inhomogeneous Poisson point processes in both the parametric and nonparametric settings. Because of the complex nature of the topological summaries, we expect the mean intensity functions to be theoretically nonparametric in nature. Thus, we also investigate adaptive methods for the estimation of these persistence intensity functions to reduce potential estimation bias. We do this by applying stochastic search techniques to solving the B-spline knot-selection problem and by constructing a model building approach for truncated hierarchical B-spline parameterizations of surfaces. Finally, we investigate the task of classifying persistence intensity functions using random forest models and compare them to distance-based methods.

CHAPTER ONE

INTRODUCTION

Topological data analysis (TDA) has received a great deal of attention recently for its ability to give stable representations of the shape of high dimensional data. One can think of some of the techniques as providing a general tool for representing the shape of objects with certain invariance properties embedded into these representations [226, 227]. These flexible tools allow one to represent many kinds of data that can be challenging to work with, like graph, image, and pointset data, and to be able to incorporate various invariance properties into those representations (*e.g.*, rotation and translation invariance) and their analyses. For this reason, TDA has recently become useful for many applications across a variety of disciplines. These include biological applications like histopathology [172] and the characterization of tumor and plant morphology [65, 176], computer vision applications like texture analysis [193], engineering applications like classification of materials [131, 273, 290], cosmological applications like arguing against uniformity of the cosmic microwave background radiation [5, 239, 274, 275, 291], and more general problems of shape representation and reconstruction [24, 286]. For general review of TDA and its applications, consider texts by Kaczynski *et al.* [154], Edelsbrunner and Harer [84], Rabadan and Blumberg [242] and Carlson and Vejdemo-Johansson [51].

Successes of TDA to applications such as these have led to a substantial amount of research in developing methods for analyzing these representations and incorporating them into machine learning algorithms [50, 52, 135, 250]. A common approach is to embed persistence diagrams, the predominant representation in TDA, into a Hilbert space and subsequently build machine learning models using these functional summaries of the underlying

topological descriptors (*e.g.*, persistence intensity functions/persistence images, smooth Euler characteristic transforms, persistence landscapes and persistence silhouettes) [44, 55, 65, 167, 198, 250, 255]. This general approach makes sense, as these functional summaries can be inherently high dimensional (*i.e.*, a large number of parameters, p) and statistical learning techniques such as random forests and neural networks are well suited to handle high dimensional data such as these. Although these models can have great predictive capabilities, there are important limitations with taking this approach, in particular with hypothesis testing and the related problem of uncertainty estimation, especially in the setting of where samples are not independent and identically distributed (iid).

In TDA, limited work has been done in hypothesis testing [256]. The dominant approach to conducting hypothesis testing in TDA is via a permutation test, either, by using a distance metric on persistence diagrams, like the bottleneck distance or more generally the p-Wasserstein distance, or by using a statistic calculated on a functional summary of the diagram (*e.g.*, L_1 or L_2 norm between persistence intensity functions). In general, hypothesis testing is not commonly practiced in the field. Moreover, the hypothesis testing approaches developed in the field have been limited to relatively simple study designs, like a two-sample test, a K-sample ANOVA test, or a paired test for differences. Notably, there is a lack of methodology for estimation and testing of effects in a mixed-model (more complex repeated measures, see Pinheiro and Bates [232], for example) setting, and this lack of methodology limits the utility of topological representations away from more complicated experimental designs and ultimately limits the breadth of questions one can answer with topological data. Furthermore, predictions using standard methods are rarely accompanied with estimates of uncertainty and almost never account for lack-of-independence in calculation of these standard errors. It is for these reasons, we propose to extend prediction and inference methods for topological data to more general experimental design settings using point process methodologies.

Persistence diagrams may be interpreted as an instance (realization) of a point process in two dimensions [4]. While much of the field of TDA has focused on methods that embed persistence diagrams into a Hilbert space, we conduct our analysis directly on this point pattern space, which results in a less ad-hoc modeling approach and allows for more structured estimates of standard errors of these topological descriptors. In particular, we use mixed-curve models (see text by Wu and Zhang [305]) for the estimation of replicated inhomogeneous point process models for modeling the intensity measures of collections of persistence diagrams arising from different populations. By taking this approach, we can more directly model data arising from hierarchical sampling designs that might arise from taking repeated measures on the same subject and/or from multiple places on the same image from the same subject. As a consequence, this modelling approach allows us to ask more sophisticated questions about topological data, like "How do the persistence diagrams vary across populations controlled for subject-to-subject variation?" or "Is there variability associated to the persistence diagrams sampled from some random population and how large is it?".

This approach of using replicated inhomogeneous point process models does, however, present computational challenges. Because the intensity measures of persistence diagrams are inherently inhomogeneous and nonparametric, and the number of points per diagram and number of diagrams can be large in practice, we find ourselves in the setting where both the number of parameters p and the number of observations n can be quite large. One way we propose to address this issue is to use local polynomial kernel regression estimators to estimate a mixed-curve model. This approach is "perfectly parallelizable" and locally requires drastically fewer parameters to estimate. To accomplish hypothesis testing with this model, we then apply Westfall-Young multiple-comparison corrections to globalize pointwise hypothesis tests across the entire persistence diagram. This enables us to conduct hypothesis testing on the persistence diagrams in a way that is more computationally feasible in some settings and allows us to account for the hierarchical nature of the data. These contributions

are the topic of Chapter 3.

Another challenge that arises in this setting is the problem of estimating the intensity measures of the persistence diagram point processes assuming spatially inhomogeneous intensity functions. To reduce bias in the estimation of these intensity functions, we propose to use adaptive methods for estimating these intensity functions. In particular, we propose the use of stochastic search techniques to solve the knot selection problem for B-spline estimation of these intensity functions. We also propose the use of truncated hierarchical B-spline bases for parameterizing these intensity functions and develop a model building approach for constructing these bases. In general, the problem of estimating these intensity functions is important for both hypothesis testing and prediction, as the mean intensity functions are of interest for hypothesis testing and are also the features used for prediction in machine learning models. These contributions are the topics of Chapters 5 and 6, respectively.

This research is motivated by a prostate cancer histopathology application where data are sampled according to a hierarchical sampling design. In particular, region of interest (ROI) color images are obtained from subdividing whole slide images (WSI) sampled from male patients from a particular hospital. Each ROI is expert-annotated with a cancer grade (Gleason score) and we are interested in understanding how the topological structure of the tissue (as represented by persistence diagrams) varies across cancer grades. However, obtaining quality data is expensive and time-consuming, as it requires expert annotation. One way we try to alleviate this burden is to develop a data generating model that can generate synthetic data similar to our application. With such synthetic data, we can generate as much data as we want and can use this to study the properties of proposed methods under controlled simulated experiments. Explanation of this application and the data generating model, along with a general background on TDA is the topic of Chapter 2.

Finally, we consider the problem of prediction of class labels associated to persistence diagrams by classifying persistence intensity functions. In particular, we consider the problem

of predicting cancer grade associated to a persistence intensity function using a random forest modeling approach. This gives us a way to predict class labels associated to persistence diagrams while also providing diagnostic tools to understand which regions of the persistence diagram are most important for prediction. We compare this approach to distance-based methods, namely with K-nearest neighbors classification. This contribution is the topic of Chapter 7. Finally, we conclude with a discussion of these contributions and future directions for research in TDA in Chapter 8.

CHAPTER TWO

DATA AND BACKGROUND

A main motivation for our work is the analysis of histopathology image data of prostate cancer tissue. We use persistent homology to give a stable representation of "shape" for these raw pixel images and wish to conduct statistical inference with these representations suggested by the sampling design. Because we are developing methods to be able to conduct analysis on these data, we must also understand how these methods perform in more controlled settings. Thus, we develop a parameterized data generation process to simulate nuclei location data with characteristics similar to the real world dataset we wish to analyze. This way, we are able to investigate the consequences of the assumptions we may make on our real world dataset. In this chapter, we describe both the real dataset and the synthetic dataset we will consider throughout this work. Because the primary focus of this work is on the analysis of topological summaries of data, namely persistence diagrams, persistence intensity functions and the related persistence image, we also provide necessary background on topological data analysis, persistence diagrams, persistence intensity functions, and persistence images. Finally, because the techniques we develop share methodology with techniques used in functional data analysis, we also provide a brief background on functional data analysis.

2.1 Real Dataset: Histopathology

Through the example of histopathology image data, we illustrate the problem of statistical inference for topological data. Histopathology is the microscopic study of tissue in order to determine manifestation of disease present in the tissue. Our specific data corresponds to tissue sampled from prostate cancer patients that have undergone radical prostatectomy. The data acquisition process involves taking the removed prostate tissue, staining it with

Hematoxylin and Eosin (H&E), and then embedding it into a wax paraffin. Then, thin slices of this stained and embedded tissue are taken and placed on a slide to be inspected visually under a microscope by a pathologist so that a grade can be assigned to the prostate cancer tissue. In our case, these "whole slide images" (WSIs) are digitized with high resolution cameras at 20x magnification to be stored and analyzed computationally. Then individual 512×512 pixel "regions of interest" (ROIs) were extracted from these whole slide images and subsequently graded by a pathologist giving a pure Gleason grade to each ROI.

The prostate is a male reproductive gland located near the base of the bladder, and is about the size and shape of a walnut. Structurally, a healthy prostate is made up of many finger-like tubular glands arranged in parallel, radially around the prostate. However, if prostate cancer is present, the organization and shape of these tubular glands deteriorates as cancer progresses. When a patient is suspected of having prostate cancer (often from elevated levels of Prostate-Specific Antigen (PSA) in the blood), the typical course of action is to take several needle core biopsies of prostate tissue. Cross sections of these biopsies are then stained with H&E and examined under a microscope by a pathologist to be graded using the Gleason grading system [74] (or the updated ISUP grading system [92, 93]).

Developed in the 1970's, the Gleason grading system assigns two numbers to a slide image on a scale from one to five based on a visual inspection of cancer severity, where the first and second numbers represent the most prevelant and second most prevalent highest grade patterns seen, respectively. The sum of these two numbers is referred to as the Gleason score. See Figure 2.1 for an overview of the Gleason grading system and how cellular structure changes as Gleason grade increases. It is this score that is the basis for deciding much of a patient's treatment recommendations. We refer the reader to [143] for a more thorough introduction to the Gleason grading system.

Broadly speaking, the pathologist will assign a grade to tissue based on a variety of visual patterns of the stained tissue. For example, telescoping glands are often graded as

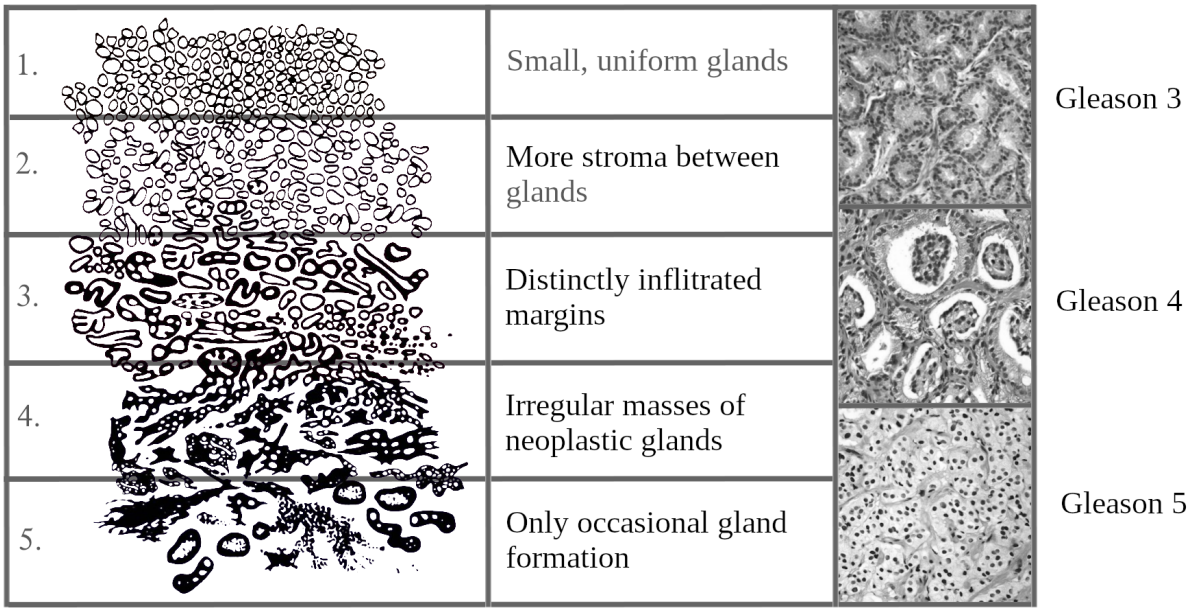


Figure 2.1: The Gleason grading system with examples of Gleason grades 3, 4 and 5. Cellular structure goes from being well differentiated, to moderately differentiated, to poorly differentiated and anaplastic.

Gleason grade 3 while, glomeruloid and cribriform are usually graded as Gleason grades 4 and cromedonecrosis glands are usually graded as Gleason grade 5 [91]. See Figure 2.2 for some examples of these visual patterns present in prostate cancer[1]. In general, as prostate cancer progresses, tissue becomes less differentiated and more anaplastic (*i.e.*, loses its structure and becomes more chaotic). It is important to note, however, that there are subtypes of patterns within Gleason grades, exhibiting a variety of visual patterns (see [143] for a more thorough discussion of Gleason grading patterns).

In the Gleason grading system, tissue is graded according to two numbers, the most predominant tissue type and the second most predominant tissue type and is written in order as a sum (*e.g.*, $3+4$ or $4+3$). Differentiating between $3+4$ and $4+3$ prostate cancer has

[1]Only Gleason grades 3, 4 and 5 are shown since grades 1 and 2 are no longer viewed as clinically significant [93]

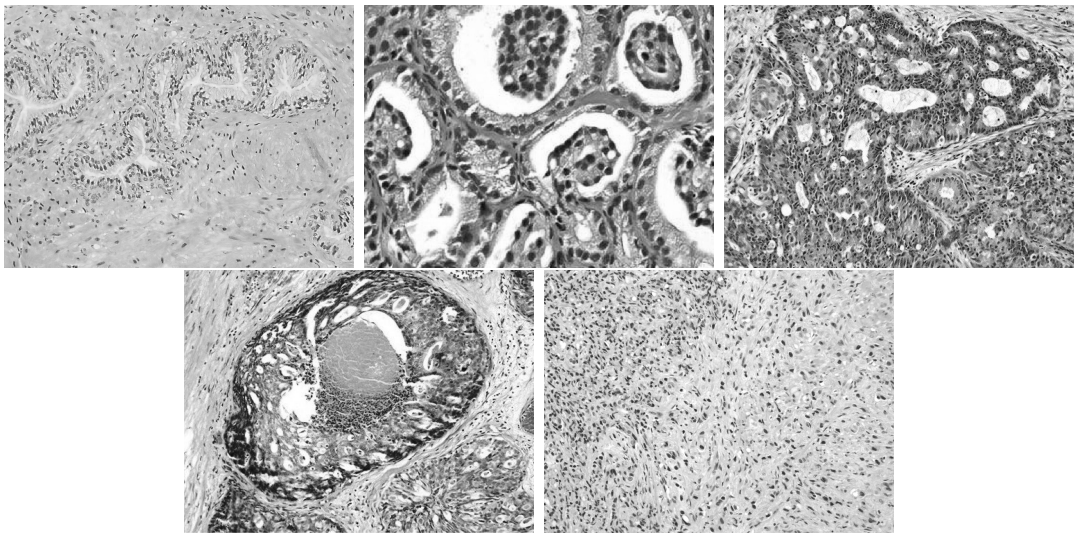


Figure 2.2: Examples of prostate cancer tissue at varying magnification. Top left: normal prostate cells. Top middle: glomeruloid glands (Gleason grade 4). Top right: cribriform glands (Gleason grade 4). Bottom left: cribriform glands with necrosis (Gleason grade 5). Bottom right: Gleason grade 4 (left half) and Gleason grade 5 (right half) cancer cells.

become an important topic in histopathology of prostate cancer as grade $4+3$ has been found to be much more aggressive than $3+4$ [276], having biochemical recurrence free survival (BFS) rates (*i.e.*, both survived and had non-clinically significant levels of Prostate Specific Antigen (PSA) in blood samples) of 69.7 and 88.1 after 5 years, respectively [231]. In addition, $3+4$ and $4+3$ Gleason grades have the greatest amount of inter and intra-grader variability [222], meaning that for these grades, pathologists are more likely to differ in grading and more likely to change the grading on a later examination. This has prompted efforts to help distinguish between these two grades more reliably [171].

For our data, the slides are captured digitally with high resolution cameras at $20\times$ magnification, the same magnification pathologists sometimes need to grade a tissue sample accurately. For computational purposes and the desire to prevent losing relevant features at the given resolution, these large images, which we refer to as whole slide images (WSIs), are partitioned into small 512×512 pixel size regions of interest (ROIs). Each of these

ROIs have been expertly graded by a pathologist as having either Gleason grade 3, 4, or 5. Further, multiple whole slide images are sampled within each patient and while not occurring in our data, one can easily imagine a scenario where multiple patients are obtained from different data providers (*e.g.*, hospitals). Such data are referred to as hierarchical data as we can organize the dependence among samples in a hierarchical fashion [107]. An important characteristic of hierarchical data such as these is that we should not make the assumption that each sampled ROI image is independent from one another, nor that each whole slide image or patient is independent from one another depending on how they relate in the sampling hierarchy. For example, because staining procedures can vary from hospital to hospital and nearby population characteristics, we would expect images obtained from patients at a given hospital to share some characteristics, and also that images sampled from the same individual are likely to be more similar to one another than images sampled from different individuals.

Once these images are obtained, we then compute persistence on each of the 512×512 images, transforming them into a topological descriptor called a persistence diagram that is represented by one or more collections of points in the plane. See Figure 2.3 for an example grayscale ROI image and its two associated persistence diagrams from a height filtration on the grayscale image. For the purpose of analysis, we consider our raw data to be these collections of points in the plane (as opposed to the raw image itself) along with any other covariates associated to each pointset/image.

Importantly, we assume each of these point patterns to be a realization of an inhomogeneous point process. That is, we think of the underlying objects, filtration, and sampling procedure as inducing a stochastic process that governs the location of points in these persistence diagrams. Thus, rather than a collection of images, our primary data consist of a collection of point patterns whose intensity measures we assume to vary spatially depending on what level in the hierarchy they exist (*i.e.*, hospital, patient, slide, ROI), what

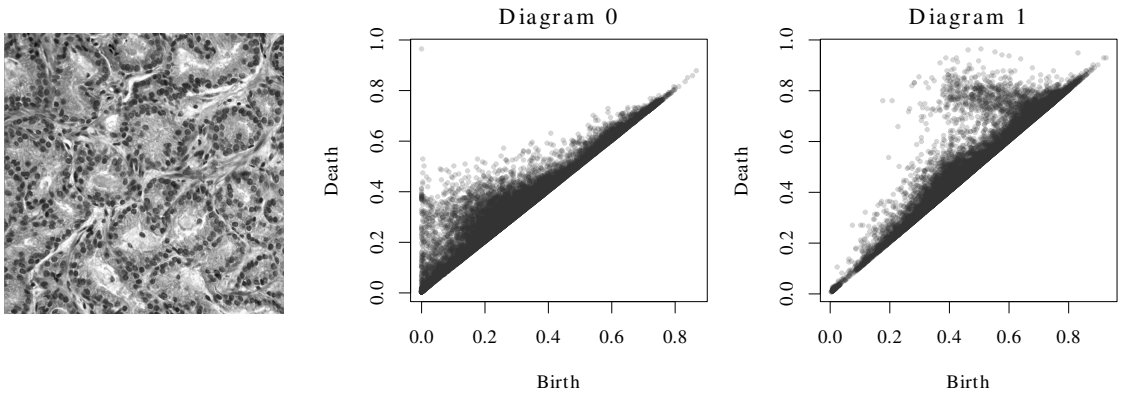


Figure 2.3: An example histopathology ROI image and its two associated persistence diagrams associated with the 0th and 1st homology classes from a height filtration on the grayscale image.

grade is assigned to the area of prostate cancer tissue (Gleason grade 3, versus Gleason grade 4) and potentially other covariates (*e.g.*, a treatment variable). Data organized in this way are known as multilevel/hierarchical data [107] and the lack-of-independence present in the data is typically modelled using mixed-effects models. As such, we consider modelling these point patterns with a (nonparametric) mixed-effect model where we account for lack of independence occurring at each level of the hierarchy. Driving our analysis of these data are two inferential questions and one predictive question. Our questions are the following:

1. Are the mean intensity measures of persistence diagrams from Gleason grades 3 and 4 images different from one another after accounting for patient-to-patient and sample-to-sample variability?

2. How much do persistence diagram intensity measures vary from patient-to-patient, and image-to-image? How much of the total variability in the intensity measures is attributable to each level of the hierarchy?

3. Can we predict whether a particular image is of Gleason grade 3 or 4 based on the intensity measure of its persistence diagram?

Question 1 is a standard inferential question. If we are using persistence diagrams as a general measure of shape, we want to know if the shape of cancer cells of one tissue type is different from the shape of another. Question 2 is motivated by the desire to understand how much variability in persistence diagrams exist at each level of the hierarchy. The purpose is twofold. First, we wish to determine if there is variation at different levels of the hierarchical data. Second, we wish to quantify the relative amount of variation in the system at each level so that we may understand how best to focus future sampling efforts. Question 3 is motivated by the practical desire to be able to automatically grade prostate cancer tissue. If we can accurately predict the grade of a tissue sample, we can potentially reduce the workload of pathologists, reduce the variability in grading between pathologists, and increase the quality of grading which can lead to downstream effects of more reliable treatment recommendations for patients. This is particularly important when differentiating between Gleason grade $3+4$ and $4+3$ prostate cancer as these two grades have very different prognoses and treatment recommendations.

From these questions of interest, we are interested in the following tasks with replicated point process data:

- estimate and test for fixed-effects,
- estimate and test for random-effects, and
- predict the class of a new point pattern.

We will address the first two items in Chapter 3 and the last in Chapter 6.

2.2 Synthetic Nuclei Data Generation

In this chapter, we also develop a synthetic dataset to be able to assess the reliability of our methodology on the previously described real dataset. In particular, we develop a

data generating procedure for the generation of nuclei locations in prostate cancer tissue. The result of this work is a library `glandmaker`[2] which is used to generate synthetic nuclei locations for prostate cancer tissue. In what follows, we describe the data generation processes for simulating nuclei locations in prostate cancer tissue.

2.2.1 Background

Our gland generation process involves simulating nuclei locations for individual glands using a disturbed model based approach (see Stoyan *et. al* [281]) and then sampling and packing many individual glands into a region to simulate an entire WSI or ROI. At a high level, we perturb points on a circle to obtain a rough gland boundary and then simulate points along the gland boundary according to a Diggle-Gratton [77] point process. Finally, we take a collection of simulated glands and pack them into a region to simulate an entire WSI or ROI. The exact data generating process varies depending on the type of tissue we wish to simulate.

We begin our description with some definitions. Instrumental in the simulation process are the use of a Gaussian bridge process and a Diggle-Gratton hard-core point process.

Definition 2.2.1 (Stochastic Process) *A* **stochastic process** *is a collection of random variables* $\{\mathbf{X_t}\}_{\mathbf{t}\in\mathcal{T}}$*, where* $\mathcal{T}$ *is an index set. If the dimension of* $\mathbf{t}$ *is* d*, then we say that* $\{\mathbf{X_t}\}$ *is a* d*-dimensional stochastic process. If* $d > 1$*, it is also common to refer to* $\{\mathbf{X}_t\}$ *as a random field. A stochastic process is considered strongly stationary if distributional properties of the process are shift invariant. In particular, if for a cumulative distribution function* $F_{\mathbf{X}}$*, we have*

$$F_{\mathbf{X}}(\mathbf{x}_{\mathbf{t}_1+\mathbf{h}}, \ldots, \mathbf{x}_{\mathbf{t}_n+\mathbf{h}}) = F_{\mathbf{X}}(\mathbf{x}_{\mathbf{t}_1}, \ldots, \mathbf{x}_{\mathbf{t}_n}),$$

[2]The library is written in R [240] and is located at https://www.github.com/jordanschupbach/glandmaker/

for all $\mathbf{h} \in \mathcal{T}$*, then the stochastic process is strongly stationary. Further, let* $\mathcal{T} = \mathbb{R}^d$ *and define the mean function* $\mathbb{E}[\mathbf{X_t}] = m_{\mathbf{X}}(\mathbf{t})$ *and the autocovariance function* $Cov(\mathbf{X_{t_1}}, \mathbf{X_{t_2}}) = \mathbb{E}[(\mathbf{X_{t_1}} - m_{\mathbf{X}}(\mathbf{t}_1))(\mathbf{X_{t_2}} - m_{\mathbf{X}}(\mathbf{t}_2))]$*. Then, the stochastic process* $\mathbf{X}_t$ *is said to be weakly (second-order) stationary if the first and second moments of the process are shift invariant. That is, a continuous stochastic process is weakly stationary if the following hold:*

1. $\mathbb{E}[\mathbf{X_{t+h}}] = \mathbb{E}[\mathbf{X_t}] = \mu$,
2. $Cov(\mathbf{X_t}, \mathbf{X_{t+h}}) = C(\boldsymbol{h})$, *and*
3. $\mathbb{E}[||\mathbf{X_t}||^2] < \infty$.

The above definition shows that a stochastic process is a general model with few restrictions. A Gaussian process adds the extra requirement that all finite subsets of the collection of random variables have a Gaussian distribution [247].

Definition 2.2.2 (Gaussian Process) *A* **Gaussian Processes***, is a stochastic process where any finite subset of* $\{\mathbf{Z_t}\}$ *has a Gaussian distribution. That is, a Gaussian process* $\mathbf{Z}_t$ *is a collection (vector) of random variables where for any subset* $\mathbf{Z}_{t^*} \subset \mathbf{Z_t}$ *with* $\mathbf{t}^* \subset \mathbf{t}$*, we have* $\mathbf{Z_{t^*}} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$*. That is, any subset of the collection of random variables is normally distributed. Moreover, a collection of random variables* $\{\mathbf{Z_t}\}_{\mathbf{t}\in\mathcal{T}}$*, indexed by* $\mathbf{t} \in \mathcal{T}$*, is said to be drawn from a Gaussian process with mean function,* $m(\cdot)$*, and covariance function,* $cov(\cdot, \cdot)$*, if for every finite collection of indices,* $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n \in \mathcal{T}$*, the random variables,* $\mathbf{Z}_{t_1}, \mathbf{Z}_{t_2}, \ldots, \mathbf{Z}_{t_n}$*, have distribution*

$$\begin{bmatrix} \mathbf{Z}_{t_1} \\ \vdots \\ \mathbf{Z}_{t_n} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{t}_1) \\ \vdots \\ m(\mathbf{t}_n) \end{bmatrix}, \begin{bmatrix} cov(\mathbf{t}_1, \mathbf{t}_1) & \ldots & cov(\mathbf{t}_1, \mathbf{t}_n) \\ \vdots & \ddots & \vdots \\ cov(\mathbf{t}_1, \mathbf{t}_n) & \ldots & cov(\mathbf{t}_n, \mathbf{t}_n) \end{bmatrix} \right)$$

We denote this via

$$\mathbf{Z_t} \sim \mathcal{GP}(m(\mathbf{t}), cov(\mathbf{t}, \mathbf{t}^*)).$$

Gaussian processes are completely specified by their mean and covariance functions, and thus a mean-zero Gaussian process is completely specified by its covariance function. Several parametric classes of covariance functions exist in the literature including the Matérn, Gaussian, Exponential, Whittle, and Spherical classes (see *e.g.*, [261]). For our purposes, we consider the Gaussian (squared exponential) covariance function defined as

$$C(\mathbf{h}; \boldsymbol{\theta}) = \boldsymbol{\theta}_1\{-\exp(-\mathbf{h}^2/\boldsymbol{\theta}_2^2)\},$$

where $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are variance and scale parameters, respectively. Figure 2.4 shows examples of Gaussian processes and the effect of varying scale parameter $\boldsymbol{\theta}_2$. We see the function appears less "wiggly" as the scale parameter increases.

There are many methods for sampling from a Gaussian process. For example, one can use a Cholesky decomposition of the covariance matrix [247], a Karhunen-Loeve expansion [108], spectral representation methods [267, 268], or a circulant embedding method [75, 303]. Our implementation uses the circulant embedding method described by Wood and Chan [303]. To perturb the circle, we are ultimately interested in a one-dimensional Gaussian bridge process so that the endpoints of the process can connect smoothly.

Definition 2.2.3 (Gaussian Bridge) *A* **Gaussian Bridge** *is a Gaussian process defined over the interval $\mathcal{T} = [a, b]$. where $\mathbf{Z}_a = \mathbf{Z}_b$. That is, a Gaussian bridge is a Gaussian process with equal endpoints. Given a mean zero Gaussian process $\mathbf{Z_t}$ defined over the index set*

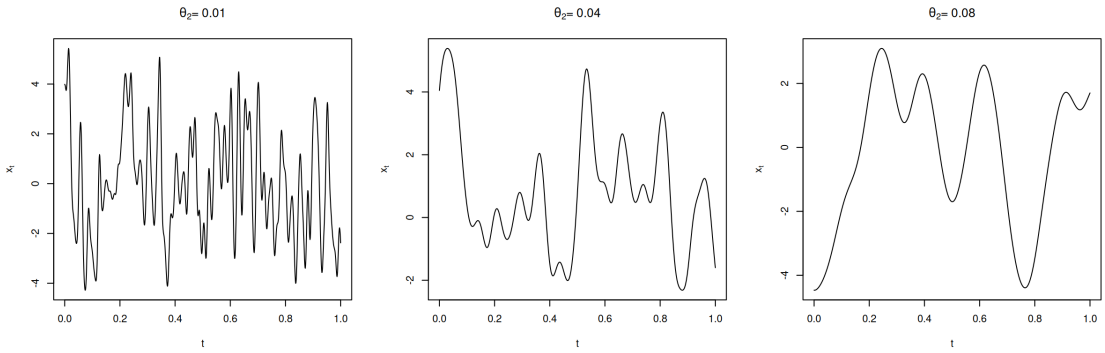


Figure 2.4: The scale of the GRF is controlled by θ_1. As shown above, increasing the scale parameter has the effect of reducing the overall "wiggliness" of the resulting GRF.

$\mathcal{T} = [0, T]$, *we can transform* $\mathbf{Z_t}$ *into a Gaussian Bridge via*

$$\mathbf{B_t} = \mathbf{Z_t} - \frac{\mathbf{t}}{T}\mathbf{Z}_T$$

where we have endpoints $\mathbf{B}_0 = \mathbf{B}_T = 0$.

We also make use of the ***Diggle-Gratton interaction model*** [77] in order to sample points (nuclei locations) along a curve. This model is an inhibitory point process model that specifies the probability of a point being in a given space as a function of its distance to neighboring points. Pairwise interaction point process models assume that the likelihood of n points being placed at specific locations, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, is proportional to the product of the pairwise interaction potentials between each pair of points,

$$\prod_{i<j} h(t_{ij}).$$

Given parameters δ, ρ, and β, the pairwise interaction for a Diggle-Gratton inhibitory

point process can be described by the pair potential

$$h(t) = \begin{cases} 0 \text{ for } t < \delta \\ \frac{t-\delta}{(\rho-\delta)^{\beta}}, \text{ for } \delta \leq t \leq \rho \\ 1 \text{ for } t > \rho \end{cases} .$$

The above expression indicates that a point will be at location t with probability 0 if it is closer than δ to its nearest neighbor, with probability 1 if it is further than ρ, and with varying probability if it lies within an annulus of inner and outer radius δ and ρ (of its nearest neighbor). Methods for generating points using this model such as thinning or MCMC-based approaches are discussed in Chui *et al.* [57]. In our approach, we start with a large number of candidate points along a curve sequentially placing points with probability determined by the above pairwise interaction function. The process is continued until a maximum number of attempted placements is reached or until no more points can be placed. The result is a collection of points along the curve that satisfy the conditions of a Diggle-Gratton inhibitory point process.

2.2.2 Single Gland Simulation Process

When generating large numbers of simulated tubular gland cross-sections and to ensure the simulation reflects natural variations of real-world data, it is necessary to incorporate various stochastic mechanisms into the process. The individual gland simulation process is as follows. We begin with a dense regular sample of n points, $\mathbf{X}_t = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, over a circle, $\mathcal{C}$, with radius, r, centered at $(0, 0)$, where the curve, $\mathcal{C}$, is parameterized by $\gamma(t) = (r\cos(t), r\sin(t))$ for $t \in [0, 2\pi]$. We then "disturb" the circle by pointwise multiplying the radius of the circle by a realization of a mean one Gaussian bridge process sampled over the same series as $\mathbf{X}_t$. That is, we generate a Gaussian bridge process $\{\mathbf{B}_t\}_{t\in[0,2\pi]}$ where

$\mathbb{E}[\mathbf{B}_t] = 1$ and define a disturbed curve as the pointwise product of the circle and the Gaussian bridge process, namely that $\tilde{\mathcal{C}} = \begin{bmatrix} \mathbf{B}_t \odot \mathbf{X}_{t[\cdot,1]} & \mathbf{B}_t \odot \mathbf{X}_{t[\cdot,2]} \end{bmatrix}$ where $\mathbf{B}_t$ is an $n \times 1$ vector and $\mathbf{X}_t$ is an $n \times 2$ matrix giving the sample coordinates of the circle, $\odot$ is the Hadamard product[3], and $\mathbf{X}_{t[\cdot,1]}$ and $\mathbf{X}_{t[\cdot,2]}$ are the first and second columns of $\mathbf{X}_t$ respectively. The result is a disturbed curve that roughly follows the shape of a circle but has natural variation in its shape.

To give the disturbed curve more natural variation, we impose further restrictions on the Gaussian bridge process. In particular, we bound the ratio of the derivative of the Gaussian bridge process at the endpoints. That is, we impose the restrictions that $|\mathbf{B}_0^{'}/\mathbf{B}_{2\pi}^{'} - 1| < \texttt{deriv\_tol}$ and $sign(\mathbf{B}_0^{'}) = -sign(\mathbf{B}_{2\pi}^{'})$. This has the effect of making the endpoints of the Gaussian bridge where connected in the disturbed curve have similar slopes so that the endpoints connect smoothly as it wraps around the circle. We also bound the maximum and minimum values of the Gaussian bridge process so that the disturbed curve does not vary too extremely. That is, the Gaussian Bridge is constrained by $0 \leq \mathbf{B}^l \leq \mathbf{B}_t \leq \mathbf{B}^u$ for all $t \in [0, 2\pi]$), so that the disturbed circle, $\tilde{\mathcal{C}}$ is constrained to be within the annulus defined by inner and outer radii $r \cdot \mathbf{B}^l$ and $r \cdot \mathbf{B}^u$, respectively. These two properties are achieved by repeatedly simulating a Gaussian bridge process and then numerically estimating the derivatives at the endpoints until the above criteria are met.

We then further disturb the resulting curve by applying STRETCH, SHEER, ROTATE and SCALE linear transformations. These transformations are defined as

$$\mathbf{T}_{\text{STRETCH}} = \begin{bmatrix} 1 & 0 \\ 0 & \eta \end{bmatrix}, \ \mathbf{T}_{\text{SHEER}} = \begin{bmatrix} 1 & \psi \\ 0 & 1 \end{bmatrix}, \ \mathbf{T}_{\text{SCALE}} = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix}, \text{ and } \mathbf{T}_{\text{ROTATE}} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix},$$

where $\eta \in [0, \infty)$ is the stretch parameter, $\psi \in [0, \infty)$ is the sheer parameter, $\alpha > 0$ is the

[3]See glossary for definition.

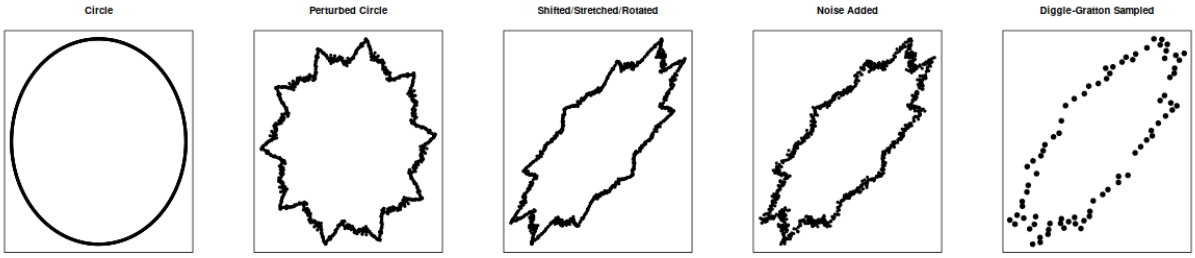


Figure 2.5: Flowchart of the process to generate a synthetic gland.

scale parameter, and $\phi \in [0, 2\pi)$ is the rotation parameter. Applying these transformations to the disturbed curve, $\tilde{\mathcal{C}}$, results in more natural variation in the shape of the gland boundary and results in an updated disturbed curve, $\mathcal{C}^* = \mathbf{T}_{\text{ROTATE}}\mathbf{T}_{\text{SHEER}}\mathbf{T}_{\text{STRETCH}}\mathbf{T}_{\text{SCALE}}\tilde{\mathcal{C}}$. We then add random noise to the sample points along the resulting curve to further increase natural variation resulting in a point pattern over the space following along the disturbed curve $\mathcal{C}^*$. Finally, we thin the point pattern according to a Diggle-Gratton inhibitory point process. This is done by starting with a large number of candidate points along the disturbed curve and sequentially placing points with probability determined by the pairwise interaction function of the Diggle-Gratton process. This results in a collection of points along the disturbed curve representing nuclei locations for an individual gland. See Figure 2.5 for a flowchart of this process resulting in a single gland.

This general disturbance process provides a flexible mechanism for generating a variety of gland shapes by varying the parameters of the circle, the Gaussian bridge process (*e.g.*, scale and variance parameters of the covariance function), the restrictions placed on the Gaussian bridge process, the amount of noise added to the sample points along the disturbed curve and the parameters of the Diggle-Gratton point process used later in the simulation. As we will describe later, we set these parameter's distributions to provide variability across a whole slide or ROI.

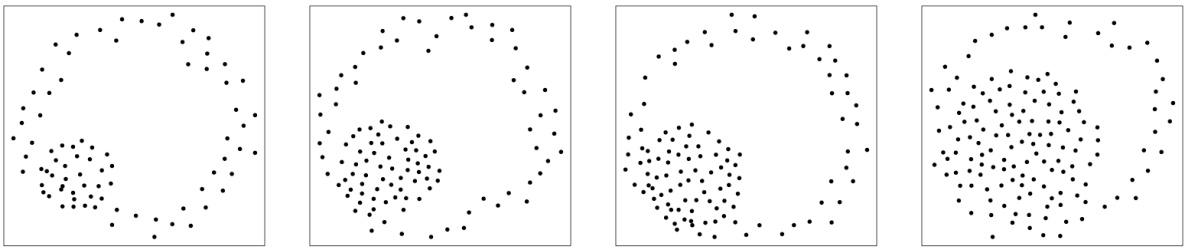


Figure 2.6: Examples of simulated glomeruloid patterns.

2.2.3 Simulating Telescoping and Glomeruloid Patterns

While the above process provides a general mechanism for simulating individual glands, for various patterns we need to modify the process. For example, to simulate Gleason grade 3 telescoping glands and Gleason grade 4 glomeruloid patterns, we take a two-stage approach where we first simulate a gland using the above process and then subsequently simulate secondary structures within the gland. The glomeruloid growth pattern features enlarged malignant glands with intraluminal cribriform extensions that adhere to one side of the gland wall, resembling a renal glomerulus [93]. Less technically, glomeruloid patterns are characterized by small secondary circular structures within a larger gland structure (see Figure 2.2 for an example). Telescoping glands mimic this shape, but are are sometimes just an artifact and are not surrounded by other nearby glomeruloid and/or cribriform patterns.

To simulate these structures, we first simulate a large gland using the above process. Then, we randomly select a proportion of the radius for a smaller secondary gland within the larger gland and a random location so that the inner gland is fully contained within the larger gland. We then simulate the secondary gland within the larger gland using the same process as above, but allow the sample points to lie within the disc defined by the smaller gland boundary. We sample nuclei locations within this disc according to a Diggle-Gratton point process. See Figure 2.6 for some examples of simulated glomeruloid patterns.

2.2.4 Whole Slide Simulation

To simulate an entire WSI or ROI, we first generate a collection of individual glands according to the processes described above and then pack them together into a region with a packing algorithm. To give variation across glands in a WSI or ROI, we specify distributions for some of the parameters involved in the gland generation process. For example, we may specify that the radius of each gland is drawn from a truncated normal or gamma distribution with shape and scale parameters chosen to reflect the desired average gland size and variability. Similarly, we may specify that the scale and variance parameters of the Gaussian bridge process are drawn from certain distributions to reflect the desired variability in the gland boundary shapes. By sampling from these parameter distributions for each gland, we can generate a diverse collection of glands that mimic the natural variability observed in real histopathology images. Distributions and/or values for each of the parameters involved in the gland generation process are summarized in Table 2.1 along with their default values (provided for a Gleason grade 3 gland).

Once the values and distributions for each parameter are specified, we sample from these distributions and generate a gland using the process described above. We repeat this process many times to obtain a collection of individual glands. Finally, we pack these glands together using circle packing techniques to generate a whole slide or region-of-interest. In particular, we use the `circleProgressiveLayout` function from the `packcircles` R package [21] to pack the glands together. This function implements an efficient space-filling approach described in [298] that packs circles together by iteratively adding circles along the boundary of the existing packing according to a layout algorithm. The result is a collection of packed circles representing the glands in a WSI or ROI.

In Figure 2.7 we provide an example ROI for three tissue types from nuclei extractions of real data along with realizations of their associated simulation model. Because these simulation models are fully generative and parametric, we can investigate the performance

Parameter	Description	Default
n	# of potential nuclei	1000
r	Radius of gland	$\sim N(2, 0.2)$
$\boldsymbol{\theta}_1$	scale of GB; affects wigglyness of GB	$\sim N(2, 0.1)$
$\boldsymbol{\theta}_2$	variance of GB; affects amplitude of GB	$\sim (0.1, 0.01)$
`deriv_tol`	tolerance for derivatives of GB endpoints	0.1
δ	lower bound for distance in Diggle-Gratton process	0.25
ρ	upper bound for distance in Diggle-Gratton process	0.35
β	inhibition parameter in Diggle-Gratton process	2
k	max number of Diggle-Gratton attempts to place nuclei	500
η	Stretch parameter	0
ψ	Sheer parameter	0
α	Scale parameter	1
ϕ	Rotation parameter	$\sim unif(0, 2\pi)$

Table 2.1: Parameters involved in the gland generation process along with their descriptions and default values.

of our statistical models on these synthetic datasets to understand the limitations of our developed models.

2.2.5 Future Work

While this simulation model provides a flexible mechanism for generating a variety of gland shapes and tissue types, there are several avenues for future work to improve the realism of the generated data. First, the current model does not account for the presence of stromal tissue and other non-glandular structures that are present in real histopathology

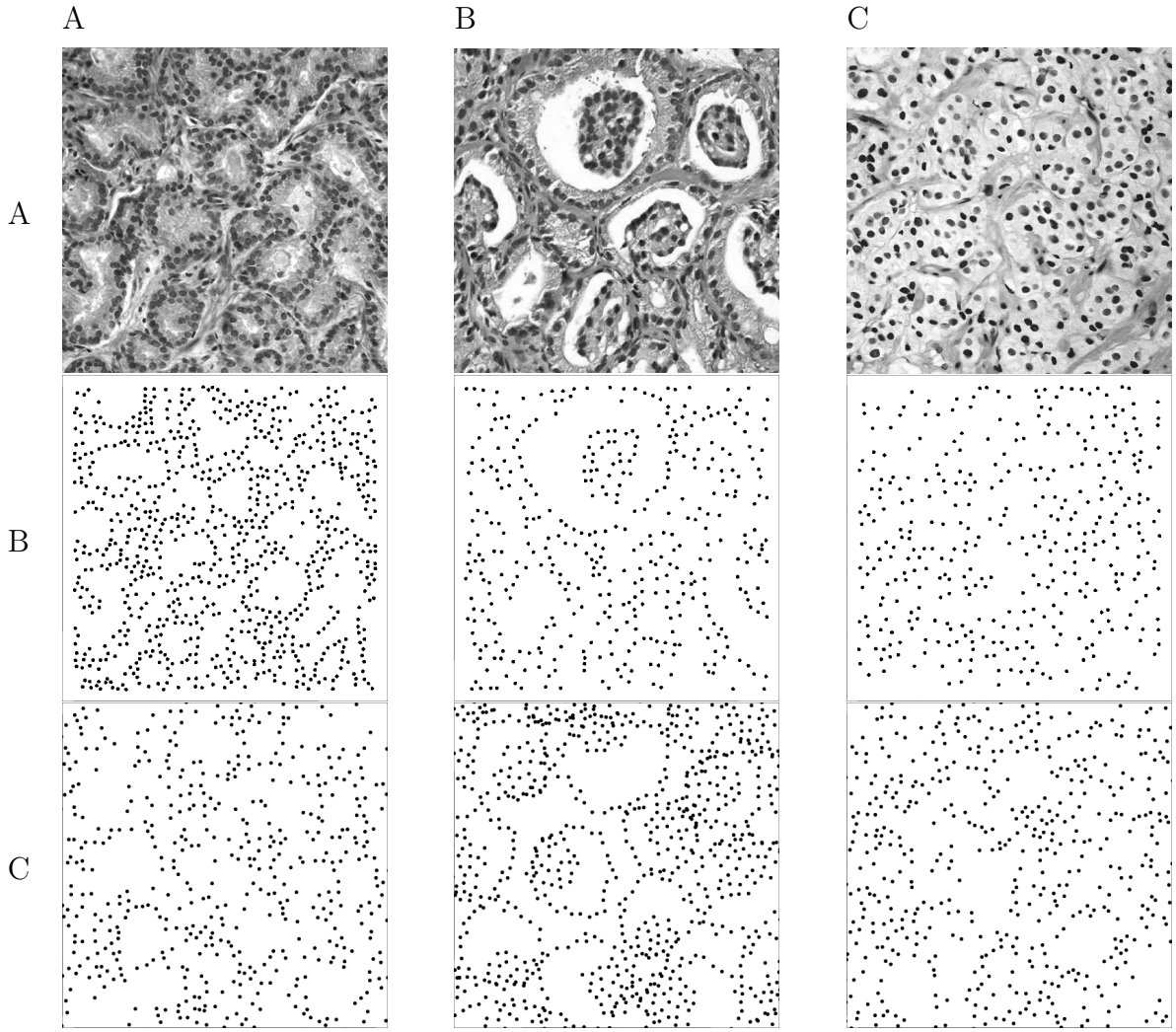


Figure 2.7: Examples of real histopathology images along with extracted nuclei locations and a simulated version. Column A shows a Gleason grade type 3, Column B shows Glomeruloid patterns of Gleason grade 4, and Column C shows Gleason grade pattern 5. Row A shows raw images, row B shows extracted nuclei for the raw images and row C shows a realization of nuclei locations under the generative simulation model.

images. Future work could involve incorporating additional structures into the simulation model to better mimic the complexity of real tissue. Second, the current model assumes that glands are independent of one another. However, in real tissue, glands may interact with one another and exhibit spatial dependencies. Future work could involve incorporating spatial dependencies between glands into the simulation model. Additionally, the current model does

not account for the presence of noise and artifacts that are present in real histopathology images. Future work could involve incorporating noise and artifacts into the simulation model to better mimic the challenges of real-world data analysis.

Another important avenue for future work is to improve the packing algorithm used to generate whole slides and ROIs. The current packing algorithm may result in unrealistic gland arrangements and overlaps. Future work could involve developing more sophisticated packing algorithms that better mimic the spatial organization of glands in real tissue. In particular, polygonal packing algorithms (*e.g.*, [11]) could be explored to better mimic the shapes of glands in real tissue. Finally, specific patterns like the healthy gland tissue shown in Figure 2.2 are not well captured by the current model. Future work could involve developing specialized models for simulating healthy gland tissue to better capture the unique characteristics of this tissue type. In particular, healthy gland tissue sometimes exhibit branching structures (*e.g.* see Figure 2.2 top-left) that are not well modeled by the current approach. A simple extension could involve simulating a graph structure to represent the branching glands and then simulating nuclei locations an average distance from the graph edges.

An additional direction for future work is to model progression of disease. The current model simulates static snapshots of tissue at a given point in time. However, in real tissue, disease progression can lead to changes in gland morphology and spatial organization over time. Future work could involve developing dynamic simulation models that can capture the progression of disease over time. This could involve incorporating temporal dependencies into the simulation model and simulating changes in gland morphology and spatial organization over time. This has the additional benefit of generating synthetic data that can be used to study disease progression and evaluate the performance of statistical models for predicting disease outcomes over time. Overall, while the current simulation model provides a useful starting point for generating synthetic histopathology data, there are many opportunities for future work to improve the realism and complexity of the generated data.

2.3 Topological Data Analysis Background

Topological Data Analysis (TDA) is a field of applied mathematics that emerged from the fields of algebraic topology and computational geometry in the early 1990s through the early 2000s after seminal works by Frosini [103], Robins [254], Edelsbrunner [86], and Carlson and Zomorodian [317]. The basic premise of TDA is that geometric and topological information of a space (*e.g.*, connectivity of a space) can give a robust qualitative and quantitative representation of the shape and structure of an object or dataset. One main tool of TDA is persistent homology, which is a method from algebraic topology to compute topological features at different spatial resolutions using topological invariants. In this section, we give an overview of the background needed to understand homology, persistent homology, and persistence diagrams. For a more thorough introduction to TDA, see textbooks by Edelsbrunner [84], Chazal *et al.* [54] and Kaczynski *et al.* [154].

2.3.1 Complexes

Before giving our description of homology and persistent homology, we describe some building blocks to our description, namely simplicial and cubical complexes.

Simplices A **Δ-simplex**, denoted Δ, is the set of all points x of $\mathbb{R}^n$ such that

$$x = \sum_{i=0}^{n} t_i v_i, \text{ where } \sum_{i=0}^{n} t_i = 1 \text{ and } t_i \geq 0.$$

The points v_i are the vertices of the simplex and we can denote an n-simplex as $\Delta = [v_0, \ldots, v_n]$ and the standard n-simplex as $\Delta^n = \{(t_0, \ldots, t_n) \in \mathbb{R}^n \mid \sum_i t_i = 1 \text{ and } t_i \geq 0 \text{ for all } i\}$ where vertices correspond to unit vectors along the coordinate axes. An n-simplex can be intuitively thought of as an n-dimensional generalization of a triangle. That is, a zero-simplex is a point, a one-simplex a line, a two simplex a triangle, a three-simplex a tetrahedron, and

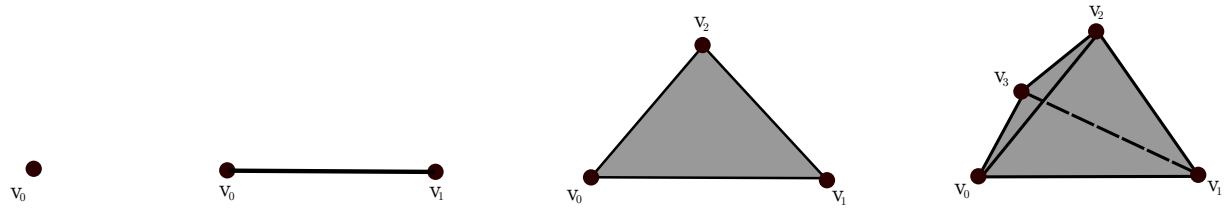


Figure 2.8: Examples of (from left to right) 0, 1, 2, and 3-simplices.

so on. We visualize these in Figure 2.8. Note that our choice in the ordering of the vertices induces orientation of the simplex and its faces which is important to keep track of when computing the homology of a simplicial complex.

Note that the **boundary** of an n-simplex, denoted $\partial(\Delta)$, is comprised of $(n+1)$ simplices of one order lower (*i.e.* $(n-1)$-simplices). A **face** σ of a simplex Δ is a subset of the simplex that is also a simplex. The **interior** of an n-simplex, denoted $int(\Delta)$, is the simplex without its boundary and is also called an "open" simplex.

Simplicial Complexes A **simplicial complex** X is a set of simplices such that every face of the simplices in X are also in X and the intersection of any two simplices in X is either empty or a face of both. Intuitively, a simplicial complex is a collection of simplices that can be glued along the faces of the simplices that comprise the collection. A **subcomplex** is a subset of the collection of simplices in a simplicial complex that contains all of the faces of its elements. For example, consider the simplicial complex in Figure 2.9 which contains 3, 2, 1 and 0-simplices as subcomplexes of the whole.

Cubical Complex A Cubical complex is a type of complex that naturally arises with image data as vertices are arranged on an integer lattice. For an extensive treatment of cubical complexes, see the text by Kaczynski, Mischaikow, and Mrozek [154]. For some $l \in Z$,

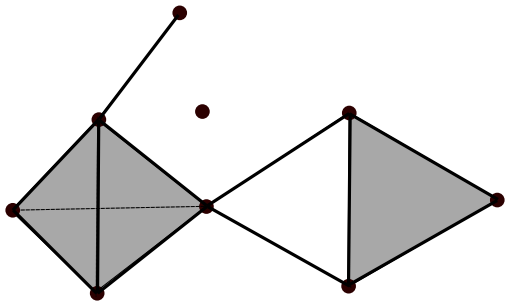


Figure 2.9: An example of a simplicial complex containing 0, 1, 2 and 3-dimensional simplices.

an **elementary interval** is a closed interval of the form

$$I = [l, l+1] \text{ or } I = [l, l].$$

An **elementary cube** Q is a finite product of elementary intervals

$$Q = I_1 \times \cdots \times I_d \subset \mathbb{R}^d.$$

A set $X \subset \mathbb{R}^d$ is **cubical** if it can be written as a finite union of elementary cubes. Note that as with simplicial complexes, boundaries (faces) of a an elementary cube are comprised of elementary cubes of a lower dimension. An example of a cubical complex is in Figure 2.10 which contains elementary cubes in 0, 1 and 2 dimensions.

Vietoris-Rips Complex A Vietoris-Rips (VR) complex is a simplicial complex constructed by connecting vertices when vertices are less than or equal to distance r of one another. In particular, let S be a finite set of points. The diameter of a subset $\sigma \subseteq S$ is defined as $\text{diam}(\sigma) = \max_{x,y \in \sigma} d(x, y)$ where d is a distance function. Then, the VR-complex of S and r is the collection of all subsets of diameter at most $2r$:

$$\text{VR}(r) = \{\sigma \subseteq S | \text{diam}(\sigma) \leq 2r\}.$$

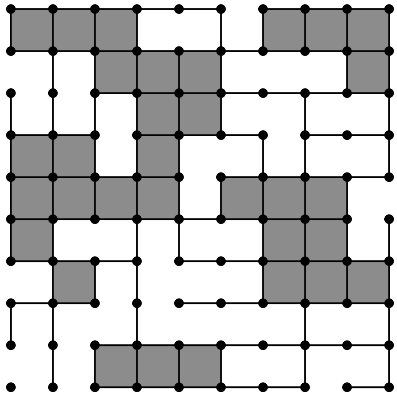


Figure 2.10: An example of a cubical complex containing 0, 1, and 2 dimensional elementary cubes.

Notice that the VR-complex written in this way is a collection of collections of points. Each collection of points in this set can be substituted with its simplex representation and thus this collection of simplices comprises a simplicial complex. Figure 2.11 contains an example of the simplices and simplicial complexes that can be constructed in this way and shows how to construct a 1, 2 and 3-simplex as well as a simplicial complex starting from a pointset. It is this construction we use when working with nuclei locations in histopathology image data.

2.3.2 Homology

Now with some introduction to and equipped with some examples of complexes, we are prepared to discuss homology. Here, we give a brief background and for a more thorough introduction, see the introductory texts by Edellsbrunner [84], Hatcher [130] or Munkres [211]. Homology is a tool from algebraic topology that allows one to characterize a topological space through calculation of invariants of those topological spaces, namely how many d-dimensional holes are in each dimension d. As an example, consider the difference between two manifolds (a topological space that looks locally Euclidean), a circle and a disk. A circle

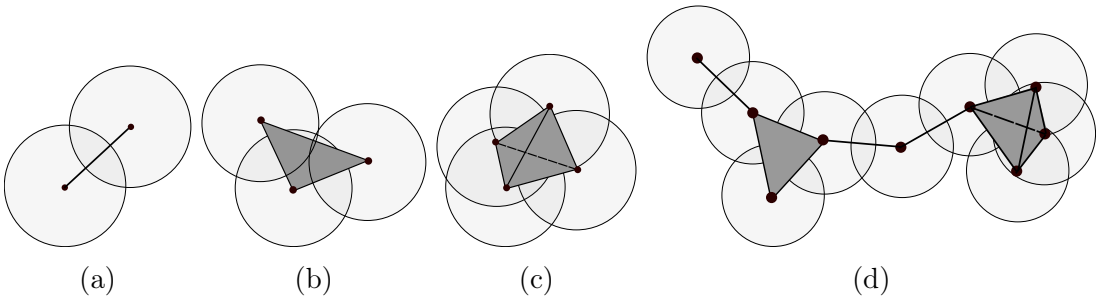


Figure 2.11: Simplices and Complex from a Vietoris-Rips filtration. (a) A 1-simplex (edge) is formed between two vertices when they are within a distance r of one another. (b) A 2-simplex (triangle) is formed between three vertices when all three vertices are pairwise within a distance r of one another. (c) A 3-simplex (tetrahedron) is formed between four vertices when all four vertices are pairwise within a distance r of one another. (d) A simplicial complex is formed by connecting all vertices that are pairwise within a distance r of one another.

has a hole in it, while the disk does not. A ***cycle*** of our manifold[4] is a closed submanifold[5] of our manifold of interest, a ***boundary*** is a cycle which is also the boundary of a submanifold. The ***homology*** of our manifold, which represents holes, is given by the equivalence class of cycles modulo boundaries. In this way, a homology class is represented as a cycle of a manifold which is not the boundary of any submanifold. More intuitively, we can think of it as the hypothetical manifold whose boundary would be a cycle but is not actually a submanifold.

A common question for topology is to determine if two spaces are topologically equivalent. It answers this question through calculation of homology and determining homological equivalence. Here, we focus on the easier question of whether two spaces are homologically equivalent (as opposed to homotopic equivalence). A typical simplifying assumption for computing homology is to restrict to computing homology with coefficients in the integers modulo two, denoted $\mathbb{Z}/2\mathbb{Z}$, and the space of simplicial complexes, for which we give a

[4]See glossary for definition
[5]See glossary for definition

description here. Note however that homology can be described in other settings.

Simplicial Homology Homology groups can be defined for simplical complexes in the following way. Let $\mathbb{X}$ be a Δ-complex and let $C_q(\mathbb{X})$ be the free abelian group[6] with basis set containing the open q-simplices of $\mathbb{X}$. Here, the elements of $C_q(\mathbb{X})$ are called **q-chains** and can be written as the formal sum $c = \sum_{i=1}^{q} a_i\sigma_i$ for a set of simplices $\{\sigma_1, \ldots, \sigma_q\}$ where we let a_i be elements of a fixed abelian group[7] G. It is typical in computation of homology to let $a_i \in G = \mathbb{Z}/2\mathbb{Z}$ which allows one to ignore orientation. An equivalent definition of a q-chain on $\mathbb{X}$ in this setting is a function c from the set of oriented q-simplices of Δ to the integers such that

a. $c(\sigma) = -c(\sigma')$ if σ and σ' are opposite orientations of the same simplex and

b. $c(\sigma) = 0$ for all but finitely many oriented q-simplices σ.

Note that in this setting C_q forms a group[8] under addition and also forms a vector space with coefficients in $\mathbb{Z}/2\mathbb{Z}$. Given a q-chain C_q, the homomorphism[9] defined by

$$\partial_q : C_q(\Delta) \to C_{q-1}(\Delta)$$

is called the **boundary operator**. If $\sigma = [v_0, \ldots, v_q]$ is an *oriented* q-simplex, then

$$\partial_q\sigma = \partial_q[v_0, \ldots, v_q] = \sum_{i=0}^{q}(-1)^i[v_0, \ldots, \hat{v_i}, \ldots, v_q]$$

where $[v_0, \ldots, \hat{v}_i, \ldots, v_q]$ is the simplex with vertex i removed. For example, we have $\partial[v_0, v_1] = [v_1] - [v_0]$ and $\partial[v_0, v_1, v_2] = [v_1, v_2] - [v_0, v_2] + [v_0, v_1]$. That is, we have the boundary of the

[6]See glossary for definition
[7]See glossary for definition
[8]See glossary for definition
[9]See glossary for definition

1-simplex associated to vertices v_0 and v_1 is the 0-simplices $[v_0]$ and $[v_1]$ and the boundary of the 2-simplex are the 1-simplices $[v_1, v_2]$, $[v_0, v_2]$ and $[v_0, v_1]$. In general, $\partial_q(\mathbb{X})$ consists of the $(n-1)$-dimensional simplices associated to the removal of the ith vertex. In the sum above, the signs inserted are for taking orientation into account.

Additionally, we see that the boundary of a q-simplex $\sigma = [u_1, \ldots, u_q]$ of dimension q is the sum (or unions depending on context) of its $(q-1)$-dimensional faces of σ. Note that the boundary homomorphism maps a q-chain to a $(q-1)$-chain. The kernel of the boundary map, $\partial_q : C_q \to C_{q-1}(\Delta)$, denoted $Ker(\partial_q)$, is called the **group of q-cycles** and is denoted Z_q. That is, a q-cycle is a q-chain with an empty boundary (*i.e.* the kernel of the boundary mapping ∂, $Ker(\partial_q)$). Because ∂ commutes under addition, a q-cycle forms a group. Note that a q-cycle is a subgroup of C_q.

The image of $\partial_{q+1} : C_{q+1}(\Delta) \to C_q$, denoted $Im(\partial_q)$, is called the **group of q-boundaries** and is denoted B_q. That is, a q-boundary is a q-chain that is the boundary of a $(q+1)$-chain (*i.e.* the image of the boundary map $Im(\partial_{q+1})$. The quotient group formed as the q-cycle group modulo the q-boundary group forms the **homology group** H_q (*i.e.* $H_q = Z_q/B_q$))

Note that H_q also forms a vector space with coefficients in $\mathbb{Z}/2\mathbb{Z}$ in the case where field coefficients are in $\mathbb{Z}/2\mathbb{Z}$. Intuitively, we can think of the rank of H_q as measuring the number of elements in a q-dimensional topological subspace that have no boundary and are not a boundary of a $(q+1)$-dimensional subspace. This rank is referred to as the $(q-1)^{\text{st}}$ Betti number. Another way to think about this is that the $(q-1)^{\text{st}}$ Betti number corresponds to the number of generating q-dimensional homological features in a topological space.

A one-dimensional connected component corresponds to zero-dimensional topological feature, a two-dimensional cycle corresponds to a one-dimensional feature, a three-dimensional void corresponds to a two-dimensional feature, continuing in this fashion for q dimensions. The $(q-1)^{\text{st}}$ Betti number of a complex serves as a measure describing H_q. In the context of

a black and white image, the connected components corresponds to disjoint groups of black pixels and a cycle components corresponds to a loop formed with the pixels (*e.g.* the ring formed by a gland).

2.3.3 Persistent Homology

Persistent Homology (PH) provides a means of measuring the scale of topological features by tracking changes in homology through a nested sequence of simplicial complexes, known as a *filtration* that evolves with respect to a size parameter $t \in \mathbb{T} \subset \mathbb{R}$. Given a topological space $\mathbb{X}$, a **filtration**, denoted $Filt(\mathbb{X})$, is a (possibly infinite) sequence, $\mathbb{X}_0 \subset \mathbb{X}_1 \subset \ldots$, of subspaces such that their union equals the entire space $\mathbb{X}$. Given an index t, a **filtration function** $f : \mathbb{X} \to \mathbb{R}$, provides a filtration, and persistent homology studies the topological changes of the sublevel sets, $\mathbb{X}_t = f^{-1}(-\infty, t]$. That is, it is primarily concerned with how (and "when") topological changes occur throughout the filtration.

For a simplicial complex $\mathbb{K}$, let $f : \mathbb{K} \to \mathbb{T}$ be a monotonic function in the sense that for $a \leq b$, where $\mathbb{K}_b$ is a subcomplex of $\mathbb{K}$ and $\mathbb{K}_a$ is a subcomplex of $\mathbb{K}_b$, $\mathbb{K}_a \subset \mathbb{K}_b \subset \mathbb{K}$. By varying our "size" parameter $t \in \mathbb{T}$, we induce an ordered sequence of simplicial complexes, which we denote $Filt(\mathbb{K})$. In persistent homology, we track changes in the homology of our topological space throughout a given filtration by tracking changes in the rank of the homology group, H_q, throughout the filtration. From an intuitive perspective, the qth homology group, $H_q(\mathbb{X})$, describes the number of $(q+1)$-dimensional holes in a given topological space $\mathbb{X}$. As such, we refer to elements from the 0th homology group as "connected components", elements from the 1st homology group as "loop components", and elements from 2 and greater homology groups as "void components". In persistent homology, we track when and how these holes appear or disappear throughout the filtration. It may be easier to see this through illustration, so we now provide three examples of filtered topological spaces.

Example: Height Filtration

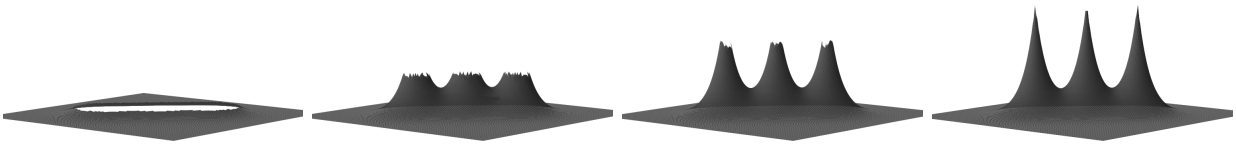


Figure 2.12: An example of a elements of a height filtration on a function of 2 variables. Here we give elements $\mathbb{M}_{0.05}$, $\mathbb{M}_{0.25}$, $\mathbb{M}_{0.45}$, and $\mathbb{M}_{0.7}$ of the filtration.

Our first example of a filtration is using a height function. Consider a two-dimensional manifold, $\mathbb{M}$, embedded in $\mathbb{R}^3$. Now, let the filtration function be the height function $f = f_u : \mathbb{M} \to \mathbb{R}$ in direction u defined by $f(x) = \langle x, u \rangle$. That is, f gives the height of a plane passing through $\mathbb{M}$ in direction u. Then, given a threshold $t \in \mathbb{R}$, the sublevel set consists of all the points in $\mathbb{M}$ with height less than t (in direction u), which we can write as $\mathbb{M}_t = f^{-1}(-\infty, t]$. For example, consider in direction z, the height filtration on the surface defined by the function

$$z(x,y) = \frac{2}{3\exp(\sqrt{(10x-3)^2+(10y-3)^2})} + \frac{2}{3\exp(\sqrt{(10x+3)^2+(10y+3)^2})} + \frac{2}{3\exp(\sqrt{(10x)^2+(10y)^2})}.$$

The height filtration consists of the sublevel-sets of this function, four of which are depicted in Figure 2.12. Throughout the filtration, critical changes in the topology of $\mathbb{M}_t$ occur. First, a single hole appears, then that hole is split into three holes which eventually get filled in.

Example: Vietoris-Rips Filtration

With a Vietoris-Rips (VR) filtration, for a pointset S, grow a t radius ball around each point $s \in S$ to construct the resulting VR complex. Every radius $t > 0$ produces a VR complex and the filtration consists of all possible VR complexes we can create from the pointset. Thus, changes occuring in our VR filtration occur for every unique distance between points $s \in S$. Consider as a concrete example of a pointset, the location of nuclei

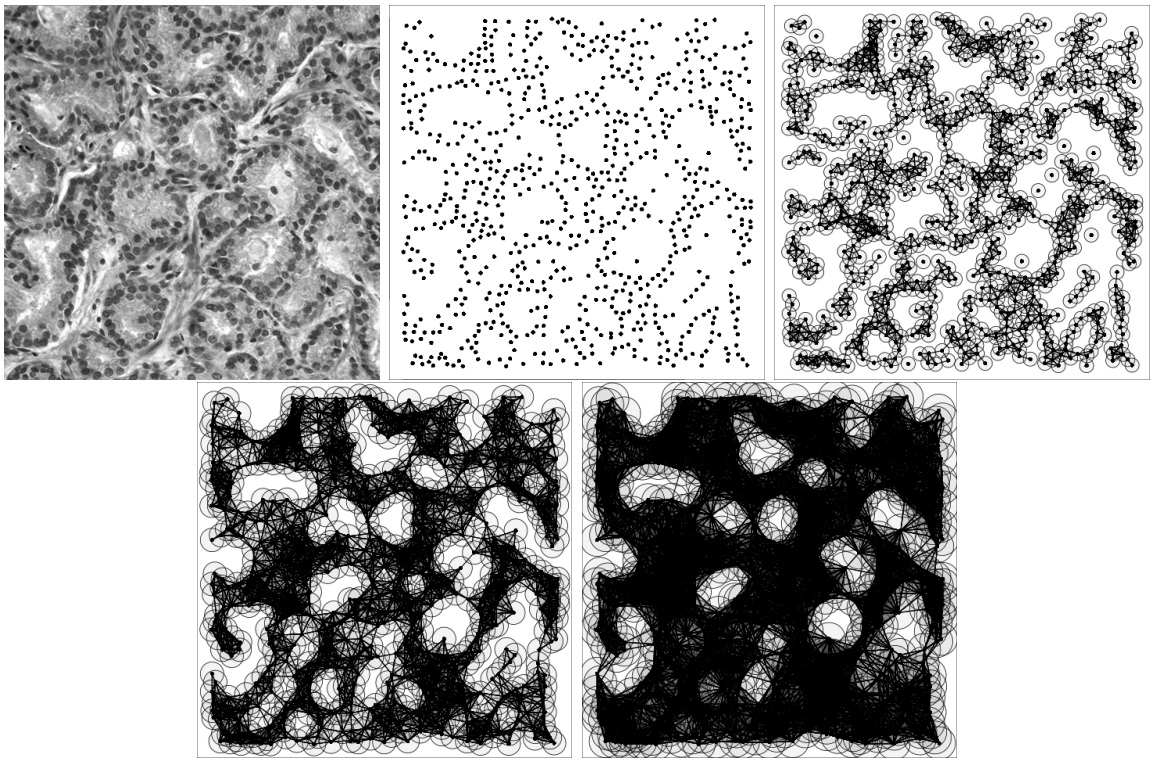


Figure 2.13: An example pointset obtained by extracting nuclei locations of a histopathology image of prostate cancer tissue. The raw image is given in the top-left, the extracted pointset of nuclei locations is given in the top-middle, and the remaining show a few of the elements of the Čech filtration.

in a histopathology image given in Figure 2.13. Throughout the filtration, disconnected components connect and may eventually form cycles (one-dimensional holes) that eventually get filled in.

Example: Filtration on Grayscale Images

For a two-dimensional digital image, there are two common ways (though others exist) for constructing a filtration. We represent each image as either a simplicial complex generated by a triangulation over the image pixel intensities or as a cubical complex. With an 8-bit grayscale image, we have pixel intensities occurring at 256 equally-spaced levels, and thus we have a finite filtration set, $Filt(\mathbb{X}) = \{\mathbb{X}_0, \ldots, \mathbb{X}_{255}\}$. One can visualize what is occurring

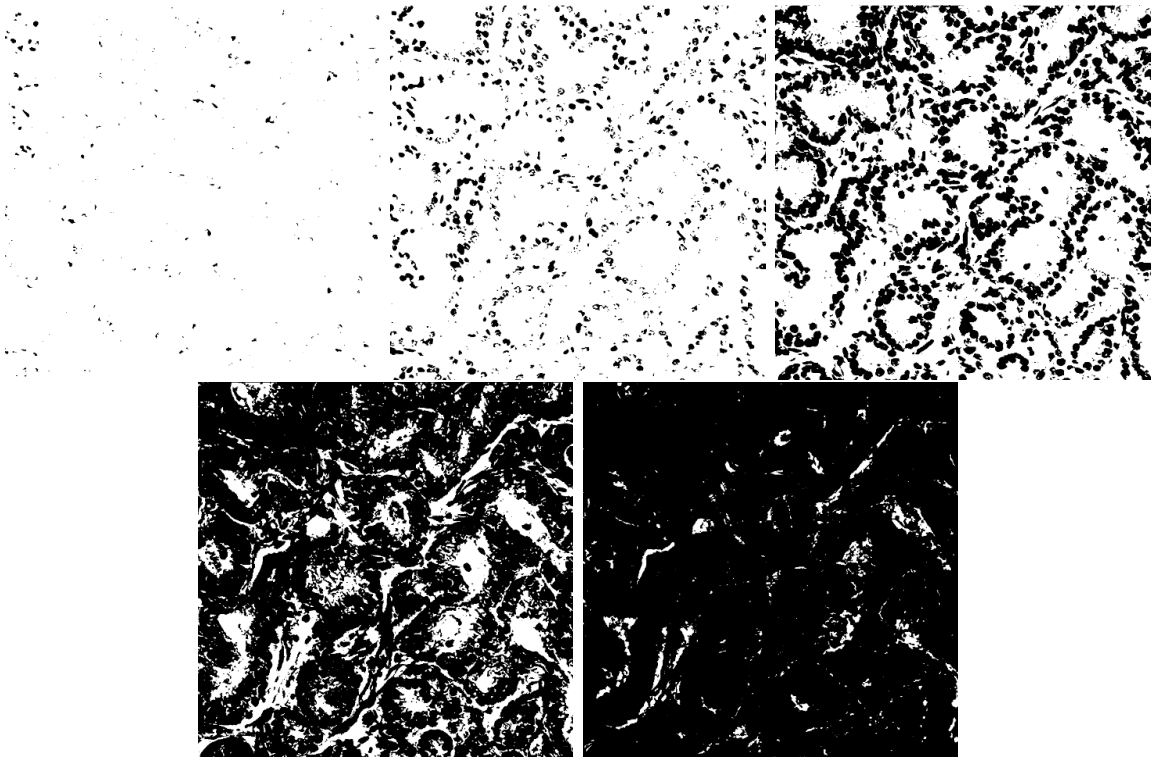


Figure 2.14: Sequence of binary masks applied to a grayscale histopathology image.

throughout this filtration as a sequence of images with varying levels of a binary mask being applied, an example of which is given in Figure 2.14. Notice that as the threshold for the binary mask increases, new connected components are born and eventually are absorbed into other connected components. Similarly, cycles are born and eventually are absorbed by other cycles.

2.3.4 Persistence Diagrams

Persistence diagrams, first introduced by Edelsbrunner *et al.* [86], give a representation of persistent homology that spurred a great deal of subsequent research in the field of TDA. A persistence diagram summarizes the "birth" and "death" times of topological features throughout a filtration of a topological space. For a simplicial complex, this may be when an edge gets connected to a vertex (a connected component) or when a cycle is born (a

loop component). An example of the persistence diagrams from a height filtration of a triangulated 1-D function is given in Figure 2.15. Since the function is one-dimensional, only the 0-dimensional homological groups (connected components) exist. The heights of critical points of the function correspond to the "times", $t \in \mathbb{T}$, where the topology of $\mathbb{M}_t$ changes throughout the filtration. Persistence diagrams give a pairing of these times.

It is important to note that the filtration function respects what is known as the "elder rule" [84] - topological features with earlier birth times absorb features that are born later. For example, in Figure 2.15, connected components are "born" at times -20.1, -11.9, and -7.7. These are critical points where new topological components appear in the filtration. The point in the sublevel set born at time -11.9 merges with the space in the sublevel set born at time -7.7 at a time of 1.4. At this time, a "death" event occurs, whereby the component which appeared at time -11.9 merges with the other born at -7.7 and persists as the representative component. Thus, we are left with a birth-death pair of $(b, d) = (-7.7, 1.4)$ in the persistence diagram. Similarly, this representative component then gets absorbed into the component born at time -20.1 at a time of 6.7, leaving us with another birth-death pair of $(b, d) = (-11.9, 6.7)$. Finally, the component born at a time of -20.1 never dies, and we are left with the birth-death pair $(b, d) = (-20.1, \infty)$. The lifetime of a particular topological feature is $d - b$ and topological features with longer lifetimes tend to be considered as more important features of the topological space as short-lived topological features can be born from random variation (see *e.g.* [96]).

Unfortunately, persistence diagrams are not amenable to statistical analysis. As an example, a unique Fréchet mean of a diagram need not exist [285]. For this reason, it is typical for diagrams to be summarized by a functional summary measure such as persistence landscapes [44], persistence intensity functions (PIFs) [56], or persistence images [3]. Our focus in this work is on the statistical estimation and modeling of persistence intensity functions, and classification of the related persistence images.

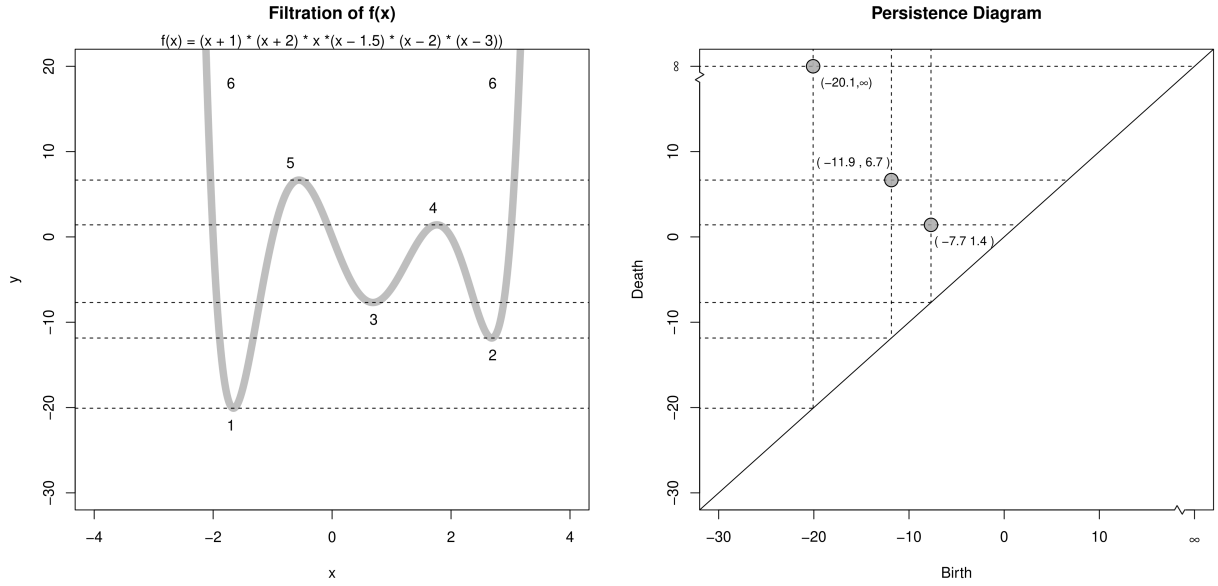


Figure 2.15: An example of a persistence diagram generated from the height filtration of a triangulation of a 1-d function.

2.3.5 Persistence Intensity Functions

Persistence intensity functions (PIFs), or persistence surfaces, were first described by Edelsbrunner *et al.* in 2012 [85] and further developed by Chen *et al.* in 2015 [56] and Adams *et al.* in 2017 [3]. PIFs give a functional representation of persistence diagrams as a 2D function. More specifically, the PIF is a weighted intensity measure of the persistence diagram. That is, it is a weighted measure of intensity for the presence of a topological feature for some birth-death combination. We can construct an estimate of the PIF in the same way as given in Chen *et al.* [56] by using a nonparametric kernel estimator of density (with a Gaussian kernel). A weighting function is then applied to this estimated function.

Consider a random sample of objects, $\{\mathcal{X}_1, \ldots, \mathcal{X}_N\}$, from population distribution, P, which are then given a topological representation such as simplicial or cubical complexes. This results in a collection of random topological spaces, $\{X_1, \ldots, X_N\}$. This random sample of topological spaces then induces a set of random functions, $(f_1, \ldots, f_N)$, which are the result

of applying a filtration to our sample of topological spaces from population distribution, P. Calculating persistent homology using these functions results in a random sample of persistence diagrams, $\mathcal{D}_i(f) = \{(b_j, d_j) : j = 1, \ldots, n\}$ for $i = 1, \ldots N$, one associated to each object.

The persistence intensity function for each diagram is then constructed as follows. Define the random measure $\Phi(\mathbf{x}) = \sum_{j=1}^{k} w_j \delta_{\mathbf{x}_j}(\mathbf{x})$, where $\delta_{\mathbf{X}}(\mathbf{x})$ is a point mass at $\mathbf{x} = (b, d)$ and $w_j > 0$ is a weight. Then, for each Borel set[10], B, of $\mathbb{R}^2$, the persistence intensity measure is defined as $R_P(B) = \mathbb{E}_P(\Phi(B)) = \mathbb{E}(\int_B \Phi(x, y)dxdy)$ where expectation is with respect to population distribution P. The persistence intensity function is then defined as $\kappa_P(x, y) = lim_{\tau \to 0} \frac{R_P(B((x,y),\tau))}{\pi\tau^2}$. Chen *et al.* [56] suggest using a weight function in Φ equal to lifetime $(d_j - b_j)$ or some power of lifetime to avoid boundary bias in their estimate of the PIF.

We do not know the distribution P, and hence $\kappa_P(x, y)$ must be estimated. Following Chen *et al.*, for a given PD, $\mathcal{D} = \{(b_j, d_j)\}_{j=1}^{n}$, the PIF can be estimated via a weighted 2D kernel density estimator

$$\hat{\kappa}_{\mathbf{H}}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} w(\mathbf{x}) |\mathbf{H}|^{-1/2} K_{\mathbf{H}}(|\mathbf{H}|^{-1/2}(\mathbf{x} - \mathbf{x}_i)) \tag{2.1}$$

where $\mathbf{x}$ is a birth-death pair (b, d), $w(\mathbf{x})$ is a weight function, $K_{\mathbf{H}}(\mathbf{x})$ is a 2D symmetric kernel function, and $\mathbf{H}$ is a 2×2 positive definite symmetric bandwidth (smoothing) matrix. An example estimate of the persistence image is in Figure 2.16 with weight equal to 1. We can estimate expectations on the population as averages of the estimated PIFs for each sample from the population

$$\hat{\kappa}_n(x, y) = \frac{1}{n} \sum_{i=1}^{n} \hat{\kappa}_{\mathbf{H},i}(x, y),$$

[10]See glossary for definition.

where $\hat{\kappa}_{\mathbf{H},i}(x, y)$ is the kernel density estimator of the diagram from the i-th dataset in our random sample. That is, it is the pointwise sample average over the individual kernel density estimates in our sample. Chen *et al.* [56] showed that this estimated PIF is pointwise asymptotically normal and an unbiased estimator of the persistence intensity measure. In addition, they constructed a two-sample permutation test for differences based on L1 distances between the mean PIFs of two samples.

The persistence image (PI) [3] is a related representation that is constructed by discretizing the persistence intensity function into a grid of point and gives a practical vectorized representation for conducting machine learning tasks like classification and clustering. Notably, Adams *et al.* showed that this representation is stable with respect to bottleneck distance [3].

Persistence Intensity Functions (PIFs) and the related Persistence Images (PIs) are the central focus of this work. In the next chapter, we will relax the assumption of independence made by Chen *et al.* [56] in the estimation of PIFs and give a method for estimation of PIFs in the case of dependent, hierarchically structured data. We will also give a method for testing for differences across groups in the ANOVA setting, accounting for this hierarchical dependence structure of the data. Chen *et al.* [56] also assume a fixed bandwidth matrix $\mathbf{H}$ in the kernel density estimator of the PIF. However, it is well-known that adaptive estimation of density and intensity functions can lead to improved estimation [2, 40, 42, 235, 293]. Hence, we will give a method for adaptive estimation of PIFs in Chapters 4 and 5 using B-spline and Truncated Hierarchical B-spline bases functional representations. Finally, in Chapter 6, we apply PIs and random forests for classification of prostate cancer histopathology images and compare to distance based kNN classification using L1 distance on PIs. Before we get to these methods, we will first give background on functional data analysis in the next section, which heavily relates to the methods developed in this work for estimation and modeling of PIFs.

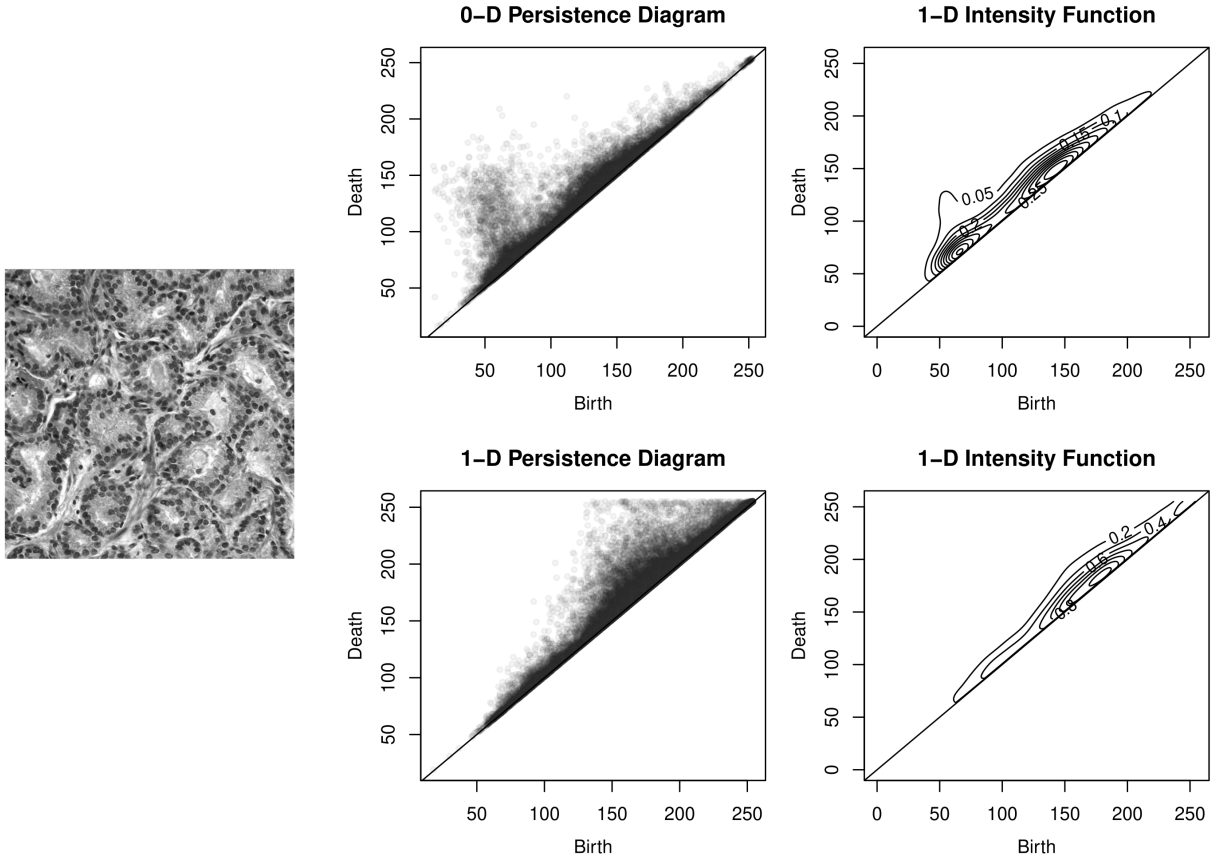


Figure 2.16: An example PIF constructed from a persistence diagram arising from the height filtration of a grayscale image. The top row gives the 0-dimensional (connected components) persistence and the bottom row gives the 1-dimensional (loop components) persistence

2.4 Functional Data Analysis Background

Functional Data Analysis (FDA) is a field of statistics that emerged in the mid-to-late twentieth century with the pioneering work of Grenander [119], Karhunen [155], Kleffe [161], Ramsay [245], and Ramsay and Silverman [246], and has since developed into a rich area of research [297] with a wide variety of applications [288]. It focuses on the statistical analysis where a datum, or "unit of analysis", is not a scalar or a point, but rather an entire smooth object, such as a function, curve, or shape. FDA treats each observation as a whole curve,

then asks how to summarize, compare, model, and infer from collections of curves. For example, in FDA, we might analyze growth curves of children, heart rate trajectories, or yearly temperature curves.

Consider a sample of curve data, $X_1(t), X_2(t), ..., X_n(t)$, observed over the interval $T = [a, b] \subset \mathbb{R}$, where $t \in T$ is a continuous variable, such as time. Assume each trajectory, $X_i(t)$, is a smooth nonparametric (*i.e.*, infinite-dimensional) functional. It is common to assume that $X_i(t)$ belongs to the Hilbert space of square-integrable functions, denoted L^2. We then assume our sample of curves to be independent and identically distributed (i.i.d.) from a population distribution of curves with mean function $\mu(t) = \mathbb{E}[X(t)]$ and covariance function $cov(s, t) = cov(X(s), X(t))$, for any $s, t \in T$.

We treat each trajectory (curve), $X_i(t)$, as a realization of a stochastic process that we observe discretely and with noise. As such, smoothing or function estimation is typically the first steps in FDA to recover the underlying smooth function from the noisy, discretely observed over t observations, where the goal is to estimate the mean and covariance functions. We often make some assumptions about the underlying stochastic process, such as assuming it is a Gaussian process, or has some basis representation.

In general, FDA encompasses a wide range of methods with analogues to its scalar counterpart, but adapted to the infinite-dimensional setting of functional data. Common tasks/analyses in FDA include smoothing and function estimation, exploratory analysis, regression and prediction, classification and clustering, hypothesis testing, registration and alignment (*i.e.*, removing phase variation due to shifts in time or event alignment), and dimension reduction (*i.e.*, summarizing infinite-dimensional objects using a small number of components). Various methods for smoothing and function estimation exist, such as kernel [25], spline [124], and wavelet smoothing [81]. Regression methods extend to functional data with function-on-function regression, scalar-on-function regression, and function-on-scalar regression [205]. Dimensionality reduction methods such as functional principal component

analysis (FPCA) [70, 119] are commonly used to summarize the dominant modes of variation among curves. Clustering [133] and classification [32] methods have been developed for functional data, such as hierarchical clustering [97] and functional logistic regression [147].

Hypothesis testing methods have been developed for comparing mean functions (*i.e.*, ANOVA). In particular, given a sample of curves from K groups, the functional ANOVA (fANOVA) [315] tests the null hypothesis that the mean functions of the groups are equal everywhere on the domain T:

$$H_0 : \mu_1(t) = \cdots = \mu_K(t) \quad \text{for all } t \in T,$$
$$H_a : \mu_i(t) \neq \mu_j(t) \quad \text{for some } i \neq j \text{ and some } t \in T.$$

Functional data analysis shares similarities to a common approach in topological data analysis, where we first summarize persistence diagrams by estimating a functional summary, such as a persistence intensity function, and then conduct statistical analysis on these functional summaries. We can think of estimation of persistence intensity functions from persistence diagrams as being directly analogous to the smoothing or function estimation task in FDA. Once estimates of the functional summaries are obtained, we are then able to directly apply FDA methods to these functional summaries, such as clustering, classification, regression, and hypothesis testing.

Our focus in this dissertation is on estimation of persistence intensity functions, conducting hypothesis testing for differences in mean persistence intensity functions, and classification of persistence intensity functions. In the next chapter, we will develop a method for conducting a functional ANOVA in the case of dependent, hierarchically structured data, which we then apply to persistence diagrams of Gleason grade prostate cancer histopathology images in Chapter 4. In Chapters 5 and 6, we will give a method for adaptive estimation of

PIFs using B-spline and Truncated Hierarchical B-spline bases for functional representations. Finally, in Chapter 6, we will classify PIFs of prostate cancer histopathology images using random forests and compare to distance based kNN classification using L1 distance between PIFs across various choices of filtration functions.

CHAPTER THREE

MIXED CURVE MODELS FOR HIERARCHICAL DATA

In the field of topological data analysis, hypothesis testing remains a challenging problem with limited solutions. Most current methods rely on the assumption of independent data [44, 56, 256], which is often violated in practice. In this chapter, we employ a local mixed-curve modelling approach paired with multiple comparison adjusted p-value surfaces to expand hypothesis testing to the setting of hierarchical topological data. This local approach provides a more computationally scalable solution compared to existing global mixed-effect model approaches.

3.1 Introduction

Topological data analysis (TDA) is an interdisciplinary field that is useful for extracting shape features from datasets. It has been successfully applied to a variety of scientific domains, including biology [306], neuroscience [277], cosmology [310], material science [273] and computer vision [193]. Hypothesis testing is a common task of statistical analysis, and is important in TDA for both understanding the importance of the topological features extracted from datasets and testing for differences between groups of topological data. However, hypothesis testing in TDA is an active area of research, and most available methods are limited by their assumption of independent observations. For example, Robinson and Turner [256] reviewed the state of hypothesis testing in the TDA literature and proposed the use of distance-based permutation tests for testing the null hypothesis that two sets of point clouds (*i.e.*, persistence diagrams) are drawn from the same distribution. This is the two-sample setting. Similarly, extensions to the ANOVA setting have been developed [53]. These methods, however, rely on the assumption that each topological descriptor is independent

from another, which is often not the case in real world data, such as when topological descriptors are obtained from the same subject multiple times (*i.e.*, a repeated measures design). In this work, we propose a methodology for topological data analysis that relaxes the assumption of independence of observations through hierarchical modelling of topological functional summaries.

In our work, we apply generalized nonparametic mixed-effect (GNPME) models, sometimes referred to as (generalized) mixed-curve (MC) models [49, 304, 305], to the analysis of topological data, in particular, for the estimation of persistence intensity functions. We estimate these models locally using generalized local polynomial kernel mixed-effect (LPKME) estimators [304] and globally using a B-spline basis expansion approach. For inference, we use multiple comparison procedures to globalize the local hypothesis tests in the former and parametric likelihood ratio testing for the latter. We begin with a brief introduction to the GNPME model and provide a more extensive review in the methods section.

Let

$$\{(y_{ij}, \mathbf{t}_{ij}) : j = 1, \ldots, n_i \; ; \; i = 1, \ldots, n\} \tag{3.1}$$

be a collection of (possibly discrete) observations y_{ij} on individual i measured longitudinally at time $\mathbf{t}_{ij}$[1]. Then, conditional on subject i, the expected value and variance of the response value y_{ij} at time $\mathbf{t}_{ij}$ is given by

$$\mathbb{E}[y_{ij}|\mathbf{t}_{ij}] = \mu_{ij} \quad \text{and} \quad \text{Var}[y_{ij}|\mathbf{t}_{ij}] = \phi w_{ij}^{-1} V(\mu_{ij}), \tag{3.2}$$

where ϕ is a scale/dispersion parameter, w_{ij} is the weight for the jth observation from the

[1]It is common to refer to 1D functional data as longitudinal data and the domain as time, even though the domain may not actually refer to time, but some other continuous covariate. We denote this with a bold symbol to also indicate that this may be a higher dimensional index set.

ith individual, and $V : \mathbb{R} \to \mathbb{R}^+$ is a known variance function. We assume that, conditional on time $\mathbf{t}_{ij}$, the response, y_{ij}, follows a distribution from the exponential family with mean, μ_{ij}, and variance, $\phi w_{ij}^{-1} V(\mu_{ij})$. Locally, this is the traditional generalized linear model (GLM) [192]. Examples of such distributions include the Gaussian, binomial, and Poisson distributions where $V(\cdot)$ is constant, $n\mu(1-\mu)$, and μ, respectively.

In the GNPME model, the subject specific mean $\mu_{ij} := m_i(\mathbf{t}_{ij})$ at time, $\mathbf{t}_{ij}$, is assumed to be related to time $\mathbf{t}_{ij}$ via a known differentiable link function, $g(\cdot)$, with inverse $h(\cdot) = g^{-1}(\cdot)$, and unknown smooth mean function, $m_i(\cdot)$, such that

$$g(\mu_{ij}) = g(m_i(\mathbf{t}_{ij})) = \alpha(\mathbf{t}_{ij}) + \nu_i(\mathbf{t}_{ij}) = \eta_i(\mathbf{t}_{ij}) = \eta_{ij}, \tag{3.3}$$

where $\alpha(\cdot)$ is the population (fixed-effect) mean surface of the process and $\nu_i(\cdot)$ is the (random-effect) deviation of the ith individual away from the population mean surface. It is typical to assume each $\nu_i(\cdot)$ are independent smooth processes with mean zero and covariance function, $\gamma_\nu(\mathbf{s}, \mathbf{t})$, that describes within-subject variation.

Adding random and fixed effect terms to the model allows for additional dependency assumptions and population differences. For example, consider the following extension of the model in Equation 3.3:

$$\begin{aligned} g(\mu_{ijk}) &= \alpha_A(\mathbf{t}_{ijk}) + \alpha_{\Delta_B}(\mathbf{t}_{ijk}) I(G_{ij} = B) + \alpha_{\Delta_C}(\mathbf{t}_{ijk}) I(G_{ij} = C) + \nu_i(\mathbf{t}_{ijk}) + \zeta_{ij}(\mathbf{t}_{ijk}) \quad (3.4) \\ &= \eta_{ij}(\mathbf{t}_{ijk}) = \eta_{ijk}, \quad (3.5) \end{aligned}$$

where $\nu_i(\cdot)$ is the subject-level random-intercept effect assumed to be an independent mean zero smooth process with covariance function $\gamma_\nu(\mathbf{s}, \mathbf{t})$ that describes within-subject variation, and $\zeta_{ij}(\cdot)$ is the sample-level random-intercept effect assumed to be independent mean zero smooth processes with covariance function $\gamma_\zeta(\mathbf{s}, \mathbf{t})$ that describes within-sample variation.

The fixed-effect population mean surfaces are given by $\alpha_A(\cdot)$ for group A (baseline level of factor), $\alpha_B(\cdot) = \alpha_A(\cdot) + \alpha_{\Delta_B}(\cdot)$ for group B, and $\alpha_C(\cdot) = \alpha_A(\cdot) + \alpha_{\Delta_C}(\cdot)$ for group C, where $\alpha_{\Delta_B}(\cdot)$ and $\alpha_{\Delta_C}(\cdot)$ are the difference surfaces between groups B and C compared to group A. Which group an observation belongs to is determined by the observed categorical variable, G_{ij}, and is encoded in the model via the indicator functions $I(G_{ij} = B)$ and $I(G_{ij} = C)$.

With this model, we are capable of addressing various hypotheses of interest. For example, the hypothesis that the population mean surfaces are equal across groups after accounting for deviations that are due to the random effects is given by

$$\begin{aligned} H_0 &: \ \alpha_A(\mathbf{t}) = \alpha_B(\mathbf{t}) = \alpha_C(\mathbf{t}) \text{ for all } \mathbf{t} \\ H_1 &: \ \alpha_i(\mathbf{t}) \neq \alpha_j(\mathbf{t}) \text{ for some } i, j, \mathbf{t}, \end{aligned} \tag{3.6}$$

or equivalently,

$$\begin{aligned} H_0 &: \ \alpha_{\Delta_B}(\mathbf{t}) = \alpha_{\Delta_C}(\mathbf{t}) = 0 \text{ for all } \mathbf{t} \\ H_1 &: \ \alpha_i(\mathbf{t}) \neq 0 \text{ for some } i \in \{\Delta_B, \Delta_C\}, \text{ and some } \mathbf{t}. \end{aligned} \tag{3.7}$$

We may also ask if there is evidence of sample to sample differences conditional on accounting for individual and population level deviations:

$$\begin{aligned} H_0 &: \ \zeta_{ij}(\mathbf{t}) = 0 \text{ for all } i, j, \mathbf{t} \\ H_1 &: \ \zeta_{ij}(\mathbf{t}) \neq 0 \text{ for some } i, j, \mathbf{t}. \end{aligned} \tag{3.8}$$

Similarly, we might be interested in evidence of individual-level differences:

$$H_0 : \ \nu_i(\mathbf{t}) = 0 \text{ for all } i, \mathbf{t} \tag{3.9}$$
$$H_1 : \ \nu_i(\mathbf{t}) \neq 0 \text{ for some } i, \mathbf{t}.$$

It is these capabilities that make mixed-curve models a useful tool for analyzing functional data that have a hierarchical structure. In this work, to test these hypotheses in the local estimation case, we apply multiple comparison adjustments to collections of p-values obtained from pointwise tests of these hypotheses across the functional domain. In particular, for the LPKME model approach, we apply likelihood ratio tests at each point in the domain and then apply multiple comparison adjustments to the resulting collection of p-values using the Westfall-Young [301] permutation-based approach. We then take the minimum over the adjusted p-value surfaces to test the global null hypothesis (*e.g.*, the hypotheses discussed in Equations 3.6-3.9). This approach has the additional benefit of identifying regions of significance that have large enough observed mean differences locally to reject the global null hypothesis while controlling for the family-wise error rate across the tests. For the B-spline basis expansion approach, we use parametric likelihood ratio tests on collections of parameters associated to B-spline basis functions to test these hypotheses. That is, each function in Equation 3.4 is represented as a linear combination of B-spline basis functions, and the parameters associated with these basis functions are tested by fitting a full and reduced model and comparing the likelihoods of these models via a likelihood ratio test (see 5.4.3 of Wu and Zhang's textbook [315] for more details on this approach).

Several TDA methods produce functional data as shape signatures (*e.g.*, persistence intensity functions [56], persistence landscapes [44], persitence silhouettes [55], smooth Euler characteristic curves [65], *etc.*), and the sampling procedure may determine whether (and how) we can consider the collection of curves to be independent (or not). Thus, the use of

generalized mixed-curve modelling is a natural extension of the current methods used in TDA, as it is more broadly applicable to functional data. Our work, however, primarily focuses on estimating collections of persistence intensity functions (PIFs) [56], which we achieve by applying a fine pixel approximation [14] to persistence diagrams and estimating the intensity functions using a log-linear Poisson mixed-curve model. The local methods we consider in this chapter provide both a novel approach to estimating intensity functions for replicated inhomogeneous Poisson processes and a novel approach to hypothesis testing in the context of functional data analysis (FDA) and TDA.

3.2 Related Works

The use of hypothesis testing in topological data analysis (TDA) is a relatively under-explored area of research. We first review the literature on hypothesis testing in TDA, then we review the literature on hypothesis testing in the context of functional data analysis (FDA). Finally, we review literature on hypothesis testing for point processes.

3.2.1 Hypothesis Testing in TDA

Three main approaches to hypothesis testing currently used in TDA are a nonparametric randomization test based on permutation or resampling of the data, a test built off of a central limit theorem of a topological descriptor, or a nonparametric kernel based hypothesis test. In the randomization-based approach, a test statistic is computed on the observed data and on collections of permuted data, and the p-value is computed as the proportion of permuted test statistics that are as or more extreme as the observed test statistic. Many of the approaches in TDA take such a resampling or permutation based approach. Robinson *et al.* [256] reviewed the state of hypothesis testing for persistence diagrams. In this work, they summarized how to conduct randomization tests for persistence diagrams and illustrate this using summary statistics based on distances between persistence diagrams. In particular, they

cover randomized methods for null hypothesis testing and provide randomization tests that are based on the statistic of average distance between populations of diagrams. This allows them to conduct hypothesis testing in the two-sample and one-way ANOVA settings. The extension to the ANOVA setting was also considered by Cericola *et al.* [53] who developed a permutation test for the one-way ANOVA setting.

In work on persistence intensity functions, Chen *et al.* [56] used a randomization test for testing whether the PIF is different for two groups by using the test statistic of the integrated absolute deviation of the estimated intensity functions for two populations. That is, they construct a randomization two-sample test for differences based on the integrated absolute differences of two estimated mean intensity functions. This test statistic is compared to the randomization distribution of this test statistic constructed by randomly permuting which population a given persistence diagram comes from and then computing the test statistic. Similarly Berry *et al.* [30] used a permutation two-sample test based on pairwise distances of functional summaries. Blumberg *et al.* [34] describes an approach to conducting a two-sample test for persistence barcodes (a representation of persistence equivalent to persistence diagrams) based on constructing an empirical distribution via simulation and comparing that to a binomial distribution. Finally, Cisewski-Kehe *et al.* [58] compared a permutation paired two-sample test for differences with the independent two-sample test over a large collection of test statistics. The p-values tended to be smaller for the paired test compared to the independent test, indicating that accounting for dependence between paired diagrams may increased power of the test. With the exception of the work by Cisewski-Kehe *et al.*, these approaches all assume independence between the topological descriptors being compared.

These approaches are all examples of randomization tests such as the permutation test originally developed by Dwass [83]. There is another type of randomization test that has not been considered in TDA, but is applicable because of the various distance metrics that can be

defined on topological descriptors, which is the perMANOVA permutation-based procedure developed by Anderson [8]. This approach is an extension of the traditional ANOVA test to the case where a distance metric can be defined on the data, and is based on a test statistic that is a multivariate analogue to Fisher's F-ratio. This approach can generally be used in TDA for hypothesis testing in the two-sample and one-way ANOVA settings by using W_p-Wasserstein distance between diagrams (though this has yet to be done in the TDA literature). However, this approach does not have extensions to the hierarchical mixed-effect setting, which is our primary interest in this work.

One major drawback to randomization approaches is that they can be computationally expensive as they require the computation of a test statistic on a large number of permutations (or resamples) of the data. This can be problematic when computing the test statistic is expensive. We similarly take a permuation-based approach, however our approach is generally applicable to functional data and more easily extends to the mixed-effect setting. We note that it is possible to extend some of the above permutation-based tests to the mixed-effect setting, however the relevant test statistic is not as clear as in the case of the two-sample test and ANOVA test. For example, several approaches to hypothesis testing in the mixed-effect model setting have been developed in the nonparametric functional data analysis and point process settings (see sections below). In contrast, because our method relies on pointwise likelihood ratio tests, the test statistic and method of finding the p-value using it is clear and the extension to the mixed-effect setting is straightforward.

Various approaches to hypothesis testing in the two-sample setting have been constructed after providing central limit theorems for topological descriptors. For example, Bubenik *et al.* [44] constructed a two-sample Z-test for comparing populations of persistence landscapes [45]. Biscio *et al.* [33] constructed a central limit theorem for persistence diagrams, persistence Betti numbers, and the accumulated persistence functional; they subsequently used them for constructing a test for complete spatial randomness of point processes. In another

example, Meng *et al.* constructed a two-sample test for the smooth Euler characteristic transform based on a central limit theorem for this topological descriptor [194]. Finally, Hiraoka *et al.* [132] constructed a central limit theorem for persistence Betti numbers of stationary point processes. In general, these approaches have also not been extended to more complicated sampling designs.

Within TDA, there is also work on hypothesis testing with kernel-based methods. These methods rely on hypothesis testing procedures like those developed by Gretton *et al.* [120, 121] that use the maximum mean discrepancy as a test statistic. This approach has been used or extended in many recent works. For example, Kwitt *et al.* [168] use this approach for conducting two-sample hypothesis tests with the universal persistence scale-space kernel. Similarly, Kusano [166] applied this approach to the persistence weighted Gaussian kernel. Although this a promising line of research because of the prevalence of theoretically sound kernel-based methods in TDA, this kernel-based methodology for hypothesis testing in sampling designs more complicated than the two-sample test remain absent from the literature. Our local approach is also kernel-based, however the approach we take is more easily extended to the hierarchical setting.

Finally, there are approaches for hypothesis testing in the context of a single persistence diagram. These aim to test whether particular topological features (points) in a persistence diagram are topologically significant, as features with short lifetimes may just be due to noise. For example, Fasy *et al.* [96] used a resampling approach to construct confidence bands across the birth-death plane on the diagrams indicating whether individual points in the persistence diagram are due to noise or not. Additionally, Adler *et al.* [5] model a persistence diagram as a Gibbs process to create resamples of a persistence diagram, a procedure they called replicating statistical topology (RST). These replicates can then be used to simulate distributions of test statistics constructed from functional representations of the persistence diagrams for performing a hypothesis test. Performing hypothesis testing

on a single persistence diagram using these approaches allows one to determine if particular features in the diagram are real or spuriously due to noise. This is useful in practice for filtering out noise from the persistence diagram or determining outliers. Our work, however, is separate from this line of research as we are primarily interested in the analysis of collections persistence diagrams whereas these works are primarily interested in the analysis of a single persistence diagram.

In general, the approach we take is of a somewhat different flavor than the above approaches. We take a functional data analysis perspective to the analysis of collections of topological descriptors. This allows us to use well established functional data analysis methods and extend them to the TDA setting. In particular, we use generalized nonparametric mixed-effect models to model collections of functional topological descriptors locally, and use multiple comparison adjusted p-value surfaces for constructing global hypothesis tests based on pointwise tests or test statistics. This approach is more generally applicable to functional data and more easily extends to the mixed-effect setting compared to the above approaches.

3.2.2 Hypothesis Testing in FDA

Hypothesis testing in the context of functional data analysis (FDA), like TDA, primarily focuses on the two-sample and one-way ANOVA settings. One key feature of FDA is that the individual observations are functions (see Ramsay and Silverman's textbook [246] for an introduction to FDA). As a consequence, the hypothesis tests are conducted on collections of curves and are concerned with testing for differences between groups of curves (*i.e.* the so-called functional ANOVA). This is a well studied area of research with a large body of literature. As an example, the `fanova.tests` function in the `fdANOVA` package [118] in R [241] currently returns (as of version 0.1.2) twelve different test statistics for conducting the functional ANOVA (fANOVA) hypothesis test. We refer the reader to the text by Zhang [315] for a review of the fANOVA literature. All of these approaches can be used to

conduct hypothesis testing in the context of functional data, and hence in the TDA setting with functional topological descriptors. However, they also only handle the two-sample and one-way ANOVA settings and do not extend to the mixed-effect setting.

In this work, we are primarily interested in hypothesis testing of functional data in the nested (or hierarchical) mixed-effect model setting. While this is a less studied area of research, there are a few notable exceptions. For example, Guo [123] developed a basis-expansion approach to the functional linear mixed model for functional data with a nested structure. The model is estimated via a standard mixed model and inference is carried out via likelihood ratio tests. This testing was also considered by Wu and Zhang [305] in their work on mixed-curve models. We use this approach to compare to our local mixed-curve approach that uses family-wise error rate (FWER) adjusted p-value surfaces for hypothesis testing. Wu and Zhang did not consider hypothesis testing in the case where estimation is conducted via local polynomial kernel estimation (LPKE), nor did Cai and Wu [49] in their original work, though a need for these methods was noted in the text by Wu and Zhang [315]. We fill this gap in the literature by using FWER adjusted p-value surfaces for global hypothesis testing in this setting. To obtain the global test, we take the minimum over the adjusted p-value surfaces. Local procedures such as this are of interest as they pair well the local polynomial kernel estimation approach to estimating mixed-curve models.

Cox and Lee [62] used the Westfall-Young [301] permutation-based approach for FWER control in the context of local testing of functional data. However, this was seen more as a post hoc analysis for detecting regions of significance (or domain selection), also referred to as local testing, rather than providing a global hypothesis test [296]. However, we construct the global hypothesis test by taking the minimum over the adjusted p-value surface to test the global null hypothesis, which corresponds to a discretized version of the global test given in Equation 3.6.

Local hypothesis testing has been developed recently in the FDA setting, is an active

area of research, and there are many recent approaches applicable to the TDA setting that could be used for local testing. For example, interval-wise testing [233, 296], threshold-wise testing [209], multiple comparison adjustment [62, 309], and envelope [207] approaches have all been considered for local testing [208]. Vsevolozhskaya *et al.* [296] used an interval-based approach based off of the closure principal [189] to construct a coherent local/global testing procedure using a finite partition of the domain. Vsevolozhskaya *et al.* [294] later extended this approach to allow for pairwise comparisons between groups in the one-way ANOVA setting. Pini and Vantini [233] similarly developed interval-based approach based on a permutation procedure by Hall and Tajvidi [128] and an L_2 distance between restricted sample means. Their approach provides both pointwise adjusted p-values and interval-wise adjusted p-values, does not depend on a partition of the domain, but also does not allow for pairwise comparisons.

Mrkvička *et al.* constructed a graphical test as an extension to a global rank envelope-test for spatial data [212]. This approach allows for both global and local testing based on extreme rank lengths. Mrkvička *et al.* later compared this method to several other local testing approaches [207]. Abramowicz *et al.* [1] developed a threshold-based approach that considers sublevel and superlevel sets of the pointwise p-value curves. They developed tests for both the global null hypothesis and for local testing. Nichols and Holmes [217] developed an approach based on pointwise permutations of a maximum statistic. Pantazis *et al.* [223] took a similar approach based on the minimum p-value over a large number of permutations. Xu and Reiss [309] developed step-down based and extreme rank length adjustment methods to construct more powerful global envelope tests.

Finally, there are combining function approaches based on combining p-values (*e.g.*, Fisher's method) which can also be used for global hypothesis testing. For example, Vsevolozhskaya *et al.* [295] used a combination of resampling, combination functions, and the closure principle to construct a coherent global and local hypothesis testing procedure.

This is an interesting approach as there are many possible combining functions that could similarly be used. Recent approaches to combining dependent p-values include works by Poole *et al.* [236], Zhang and Wu [314], and Dai *et al.* [68].

None of these approaches have been applied to the TDA setting, nor have they been applied to the mixed-effect model setting. In this work, we fill this gap in the literature by applying the Westfall-Young permutation-based approach for multiple comparison adjustment to collections of p-values obtained from pointwise likelihood ratio tests in the context of generalized nonparametric mixed-effect models estimated via LPKME modelling. We then take the minimum over the adjusted p-value surface to test the global null hypothesis, applied to the TDA descriptor setting.

3.2.3 Point Processes

In our work, we analyze collections of persistence diagrams as a replicated inhomogeneous Poisson point process. Point process literature typically focuses on the analysis of a single point pattern, concentrating on the estimation of the intensity function of the point process or some other functional representation, such as the K-function [253] or the related L-function [31], which describe how regular, clustered, or random, a point pattern is at different spatial scales. This is a well studied area of research with many approaches. Typically, this analysis is done via a basis-expansion approach (*e.g.*, B-Spline bases [219]), nonparametric kernel estimation [76], or with Bayesian approaches such as the log Gaussian Cox process (LGCP) model [203]. In this single point-pattern setting, hypothesis testing is typically concerned with testing whether a point pattern is spatially homogeneous (*i.e.*, complete spatial randomness) or spatially related to some covariate, or for testing for the presence of clustering or inhibition in the point pattern.

Although replicated point pattern data are becoming increasingly prevalent, research within this area remains limited. Replicated point processes were considered from a general

and theoretical perspective in a book by Karr [156]. From a more applied perspective, early work by Howard *et al.* [139] conducted an analysis of replicated point patterns (though not formally developed) where an ANOVA was conducted on a discretized nonparametric estimator of density. Later, Diggle *et al.* [78] conducted an ANOVA of the distribution of pyrimidal neurons via nonparametric estimates of the reduced second moment measure (*i.e.*, Ripley's K-function) and a bootstrap testing procedure. This approach has been extended many times over the years. For example, Baddeley *et al.* [15] extended the method for 3-D point patterns using second-order summary measures (*i.e.*, K, G, and L-functions). It was extended to another ANOVA test statistic by Wilson *et al.* [302] and to the bivariate interaction case by Landau *et al.* [170]. Finally, it was extended to the mixed-effect modeling case by Landau *et al.* [169] and to a two-way ANOVA case by Ramón *et al.* [244]. More recently, Xu *et al.* [308] took a similar nonparametric approach to the estimation of the pair correlation function for replicated inhomogeneous Poisson point pattern data.

From a parametric paradigm, replicated point patterns were discussed by Baddeley and Turner in their paper on approximating the pseudolikelihood [17]. This approach was subsequently applied to a single-factor design by Mateu [191] where hypothesis testing was conducted via Monte-Carlo simulation of the likelihood ratio statistic. Both the nonparametric K-function approach and the parametric pseudolikelihood approaches were compared by Diggle *et al.* [79]. Finally, this maximum pseudolikelihood approach was extended to the mixed-model setting case by Bell and Grunwald [22, 23] and later redeveloped by Illian *et al.* [144]. King *et al.* [159] extended this approach to a Bayesian MCMC estimation approach, allowing for the estimation of credible intervals of model parameters.

A separate line of research involves modeling replicated point patterns in a Bayesian setting. Most of this research is conducted under the log Gaussian Cox (or doubly stochastic) point process model. Historically, these models were estimated via MCMC sampling [203]. More recently, a stochastic partial differential equation (SPDE) approach to estimation

using INLA has been developed [145, 260]. Illian *et al.* extended the LGCP model to the mixed-effect setting and used INLA for estimation of a point process with temporally varying effects [146].

3.3 Background

We begin the description of our methods by providing an overview of the necessary background material in splines, point processes, multiple comparison procedures, and multi-level mixed-effect models. In particular, we introduce B-splines and tensor product B-splines, and how to use them to represent functions in one and two (and greater) dimensions. While not the focus of this chapter, we use spline-based approaches as a direct competitor to the local polynomial kernel estimation approach taken in this work. We will also consider the spline-based approach for adaptive estimation of PIFs in Chapters 4 and 5. Next, because our modelling assumptions for persistence diagrams include spatial inhomogeneity and local independence (in birth-death space) of points, we introduce Poisson point processes and the fine-pixel approximation for estimating the intensity function of an inhomogeneous Poisson point process parametrically. The fine-pixel approximation allows us to use a log-linear Poisson regression model to estimate the intensity functions of an inhomogeneous Poisson point process and reduce the problem of estimating the intensity function to a generalized linear regression problem. We then introduce replicated inhomogeneous Poisson point processes and how to estimate their associated intensity functions. Next, we review multiple comparison adjustment procedures for hypothesis testing with collections of p-values. Finally, we review multi-level mixed-effect models that we use pointwise in our local polynomial kernel estimation approach to estimate the intensity functions of replicated inhomogeneous Poisson point processes.

3.3.1 Splines

Splines provide a flexible way to represent functions. There are many types of splines including B-splines, M-splines [67], I-splines [67], P-splines [88], natural splines, etc. In this work, we focus on B-splines (basis splines) and tensor product B-splines.

B-Splines B-splines are a flexible way to define a curve as a linear combination of polynomial basis functions of a given order [263]. B-spline basis functions can be defined by the Cox-de Boor [63, 71] recursive formulation in the following way.

Definition 3.3.1 (B-spline Basis Functions) *Let $U = \{u_0, \ldots, u_m\}$ be a non-decreasing sequence with $u \in \mathbb{R}$. We call each u_i a* **knot** *and U the* **knot vector**. *The i^{th}* **B-spline basis function** *of degree p (and order $p+1$), denoted by $N_{i,p}(u)$, is defined as*

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{else} \end{cases} \quad \text{and}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \tag{3.10}$$

To visualize this recursive definition, consider the first few basis functions of order 0, 1, and 2 for unit-spaced knot locations given in Figure 3.1. The process begins with step functions which get turned into piecewise linear functions by the recursive formulation given in Equation 3.10. These are then turned into piecewise quadratic functions by the same formulation. Each step of the recursive formula increases the degree of the polynomial using two lower-order basis functions.

It is common to use a knot vector that is "nonperiodic" or "clamped" to generate a B-spine basis, which means that we replicate the end points an additional $p+1$ times, giving

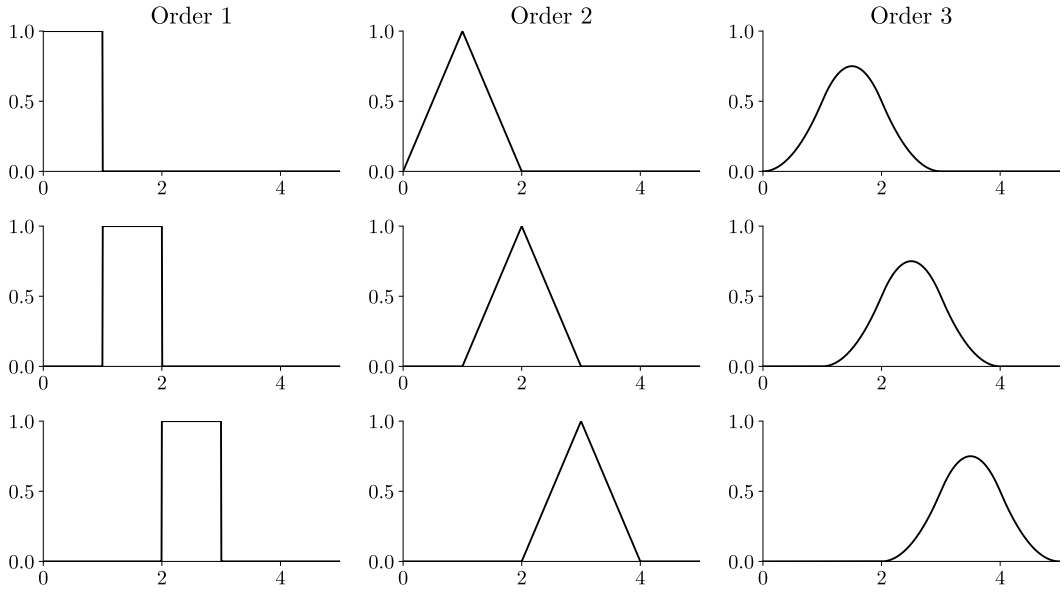


Figure 3.1: Examples of B-spline basis functions. Left to right gives basis functions of order one, two and three. From top to bottom gives the first, second and third basis functions under a knot vector with unit-spaced interior knot points.

a knot vector of the form

$$U = \{\underbrace{a, \ldots, a}_{p+1}, u_{p+1}, \ldots, u_{m+p+1}, \underbrace{b, \ldots, b}_{p+1}\}.$$

For example, consider the B-spline basis function set of degree 3 defined over the knot vector $U = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$ given in Figure 3.2. Notice that the addition of these knot points changes the first and last p functions from the standard basis functions depicted in Figure 3.1. These types of B-spline bases exhibit the partition-of-unity property over the entire domain $[a, b]$, meaning that the sum of the basis functions equals one at any point in the entire domain.

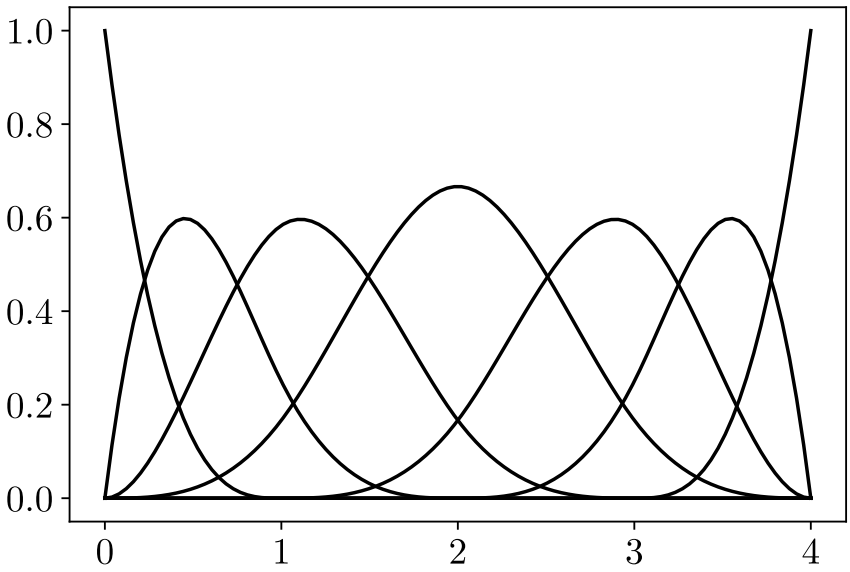


Figure 3.2: An example B-spline basis of degree 3 over clamped knot vector $U = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$.

Definition 3.3.2 (B-spline Curve) *A p-th degree* **B-spline curve** *is the curve defined by*

$$C(u) = \sum_{i=1}^{k} N_{i,p}(u) P_i \quad a \leq u \leq b \tag{3.11}$$

where $\{P_i\}_{i=1}^{k}$ *are control points and* $\{N_{i,p}\}_{i=1}^{k}$ *are the p-th degree B-spline basis functions defined on a nonperiodic (and possibly non-uniform) knot vector*

$$U = \{a, \ldots, a, u_{p+1}, u_{p+2}, \ldots, u_{m-p-1}, b, \ldots, b\}.$$

That is, a B-spline curve is defined by taking the linear combination of k B-spline basis functions with the set of control points. Note that a function space $\mathcal{V}$ is induced by the basis set, and a particular function from that function space is given by the choice of control points. We can induce a distribution of functions over that function space by giving a distribution

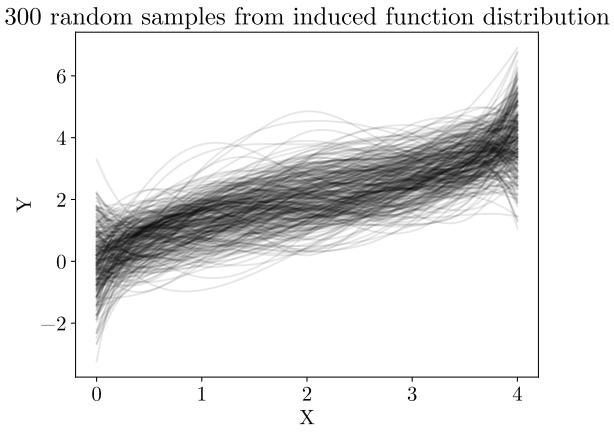


Figure 3.3: Set of 300 random B-spline curves induced by the basis set given in Figure 3.2 and with equally spaced control points ($\mathbf{P}_x = \{0, \frac{2}{3}, \frac{4}{3}, 2, \frac{8}{3}\frac{10}{3}, 4\}$) in the X-coordinate and sampled from a normal distribution along the $y = x$ line with a standard deviation of 1 (*i.e.* $P_{y,i} \sim N(P_{x,i}, 1)$).

on the control points. For example, suppose we consider the B-spline basis set given in Figure 3.2 and control points equally spaced across the domain. Suppose the y-coordinates of the control points follow a normal distribution with mean given by the x-coordinate and a standard deviation of 1. We show 300 random samples from this function distribution in Figure 3.3.

Tensor Product B-splines There are several way to construct splines in two or more dimensions. One common way is to take the tensor product of one-dimensional B-spline basis functions (see Sec. 3.4 of the NURBS book [230]). We obtain two-dimensional B-spline basis functions by taking the tensor product of two one-dimensional B-spline basis splines That is, given two basis function sets $\mathcal{B}^A = \{N_i^A(u)\}_{i=0}^m$ and $\mathcal{B}^B = \{N_i^B(v)\}_{j=0}^n$, the tensor product basis set is given by

$$\mathcal{B}^{A\otimes B} = \mathcal{B}^A \otimes \mathcal{B}^B = \{N_i^A(u)N_j^B(v)\}_{i=0,j=0}^{m,n}.$$

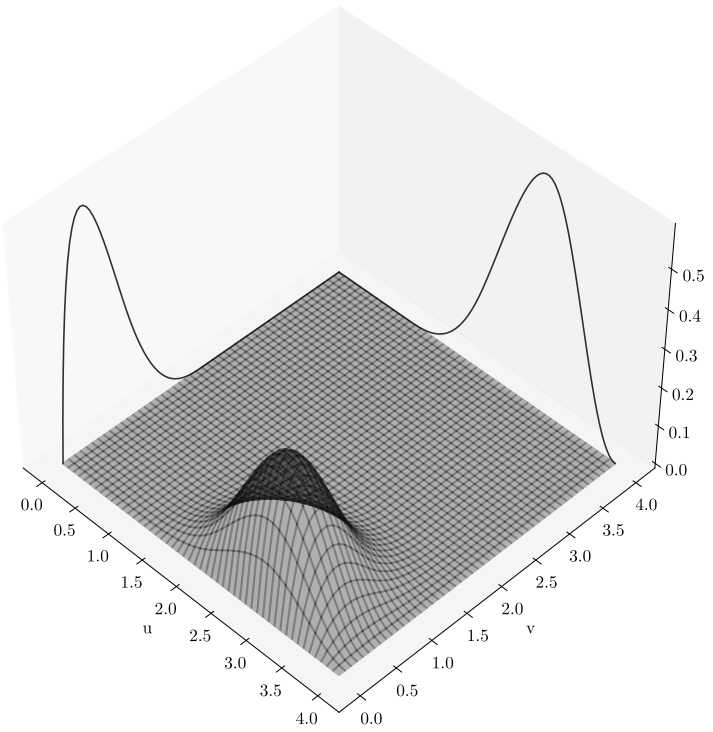


Figure 3.4: An example tensor product B-spline basis function formed by taking the tensor product of the 5th (in direction u) and 2nd (in direction v) order 2 B-spline basis functions from Figure 3.2.

For example, a tensor product basis formed by taking the tensor product of the the 2nd and 5th order-2 B-spline basis functions from the basis set given in Figure 3.2 is visualized in Figure 3.4. In practice, we can construct a two-dimensional B-spline model matrix given two one-dimensional B-spline model matrices, $\mathbf{B}_A$ and $\mathbf{B}_B$, as $\mathbf{B}_{A\otimes B} = (\mathbf{B}_A \otimes \mathbf{1}_n) \odot (\mathbf{1}_m \otimes \mathbf{B}_B)$ where $\odot$ is the hadamard product, $\otimes$ is the Kronecker product, and $\mathbf{1}_n$ is a row-vector of length n.

For (2d) B-spline surfaces, we obtain a similar construction as before with B-spline curves.

Definition 3.3.3 (Tensor Product B-spline Surface) *Given knot vectors*

$$U = \{a_u, \ldots, a_u, u_{p+1}, u_{p+2}, \ldots, u_{m-p-1}, b_u, \ldots, b_u\}$$

and

$$V = \{a_v, \ldots, a_v, v_{q+1}, v_{q+2}, \ldots, v_{n-q-1}, b_v, \ldots, b_v\},$$

the p, q-th degree **tensor product B-spline surface** *is defined by*

$$S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) P_{i,j} \quad a_u \leq u \leq b_u \quad a_v \leq v \leq b_v \tag{3.12}$$

where $\{P_{i,j}\}$ are the control points and $\{N_{i,p}\}$ and $\{N_{j,q}\}$ are the i and j-th, p and q-th degree B-spline basis functions respectively.

Similar constructions are made for B-spline surfaces in three or more dimensions by taking further tensor products of one-dimensional B-spline basis functions. However, the surfaces we consider in this work, persistence intensity functions, are at most two-dimensional and so we focus on the two-dimensional case.

3.3.2 Point Processes

In this section, we give a general introduction to Poisson point processes. We then introduce the likelihood model, log-linear models for intensity function representation, and the maximum-likelihood method for estimation. Finally, we introduce replicated mixed-effect point process models under the log-linear formulation.

Point Processes A point process is a type of stochastic process used to model the occurrence of events at the locations of points $\mathbf{s} = \{\mathbf{s}_i\}_{i=1}^{n}$ across a (possibly spatial) domain W. It is typical to consider a continuous spatial domain, $W \subset \mathbb{R}^d$. In this work, we will denote a point pattern, a realization of a point process, by $\mathbf{s}$.

Definition 3.3.4 (Poisson Point Process) *A* **Poisson point process** *is a point process where the number of points in any given region $B \subset W$ follows a Poisson distribution. In particular, let the number of points in space B, denoted $N(B)$, be Poisson distributed with mean $\mu(B)$, i.e.,*

$$P[N(B) = n] = \frac{(\mu(B))^n e^{-\mu(B)}}{n!}.$$

In this formulation, the mean number of points in any region $B \subset W$ is related to an intensity measure $\lambda(\mathbf{u})$ by $\mu(B) = \int_B \lambda(\mathbf{u}) d\mathbf{u}$. A ***homogeneous*** Poisson point process is an important special case where $\lambda(\mathbf{u}) = \lambda < \infty$ (*i.e.*, where intensity λ is constant over W.) As a result, the expected number of points to fall within a given radius around a particular point in a homogenous Poisson point process is constant across the spatial domain. Homogeneous point processes are important as they give a baseline comparison for hypothesis testing. That is, we are often interested in testing whether a given point pattern is completely spatially random (CSR) and various hypothesis testing methods exist for testing this.

Point Process Models There are many ways of constructing models for point processes. Our focus will be on the log-linear Poisson point process model (see [12] and Chapter 9 of [16]). Log-linear models consist of a flexible class of models that represent the intensity function of a point process as a log-linear function of some covariates. These models are popular because they parameterize the log of the intensity function $\lambda(\mathbf{u})$ linearly through its parameters. In particular, they give models of the form

$$\log \lambda(\mathbf{u}) = B(\mathbf{u}) + \boldsymbol{\theta}^\top \mathbf{X}(\mathbf{u})$$

or, equivalently,

$$\lambda(\mathbf{u}) = \exp(B(\mathbf{u}) + \boldsymbol{\theta}^\top \mathbf{X}(\mathbf{u})),$$

where $B(\mathbf{u})$ is a known "offset" (or "log baseline") function, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)$ is a vector of parameters and $\mathbf{X}^\top(\mathbf{u}) = (x_1(\mathbf{u}), \ldots, x_p(\mathbf{u}))$ is a vector of covariate functions that may depend on spatial location $\mathbf{u}$. This representation is linear only in the parameters and consists of a large and flexible class of functions, including the spline functions previously described.

Model Estimation Now that we have a model for representing point processes, we introduce estimation of this model via maximum likelihood. In general, computing likelihoods for point processes are intractable and so working with conditional intensity functions is required (see Chapter 7. of [69]). Here, the density of our point pattern, an unordered set of points, $\mathbf{s} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$, is given by the probability of observing points at these locations and no other locations, restricted to some finite space W. This is referred to as the local Janossy density [148], denoted $j_n(\mathbf{s}_1, \ldots, \mathbf{s}_n|W)$, for a point process restricted to W. That is, the local Janossy density is defined as

$$j_n(\mathbf{s}_1, \ldots, \mathbf{s}_n|W) = P\{\text{exactly } n \text{ points in } W \text{ at locations } \mathbf{s}_1, \ldots, \mathbf{s}_n \in W\}.$$

For the inhomogeneous Poisson process whose intensity belongs to the parametric family $\{\lambda_{\boldsymbol{\theta}}(\mathbf{u}) : \boldsymbol{\theta} \in \Theta\}$, the likelihood function is

$$L(\boldsymbol{\theta}; \mathbf{s}) \equiv \left(\prod_{i=1}^{n} \lambda_{\boldsymbol{\theta}}(\mathbf{s}_i) \right) \exp\left(- \int_W \lambda_{\boldsymbol{\theta}}(\mathbf{u}) d\mathbf{u} \right),$$

giving log-likelihood

$$\ell(\boldsymbol{\theta}; \mathbf{s}) \equiv \left(\sum_{i=1}^{n} \log(\lambda_{\boldsymbol{\theta}}(\mathbf{s}_i)) \right) - \int_W \lambda_{\boldsymbol{\theta}}(\mathbf{u}) d\mathbf{u}.$$

Daley and Vere-Jones [69] give a derivation of this likelihood. This function is important as it shows up in both likelihood and Bayesian estimation methods. Also, note that this likelihood

equation is convex for linear and log-linear models, providing guarantees for convergence to the global optimum in estimation.

Likelihood and MLEs for log-linear models Based on the previous definition of the model for the intensity function and the Poisson likelihood, the log-likelihood for the log-linear model is

$$\ell(\boldsymbol{\theta}; \mathbf{s}) = \left(\sum_{i=1}^{n} B(\mathbf{s}_i) \right) + \left(\boldsymbol{\theta}^\top \sum_{i=1}^{n} \mathbf{X}(\mathbf{s}_i) \right) - \int_W \exp\left(B(\mathbf{u}) + \boldsymbol{\theta}^\top \mathbf{X}(\mathbf{u}) \right) d\mathbf{u}. \tag{3.13}$$

If the data are such that $\sum_i \mathbf{X}_j(\mathbf{s}_i) \neq 0$ for all j, then the maximum likelihood estimator (MLE) exists and is unique. The MLE, which we denote $\hat{\boldsymbol{\theta}}$, is found the usual way by setting the score function equal to zero and solving for the parameters. Here, the score function is given by

$$\nabla \ell(\boldsymbol{\theta}; \mathbf{s}) = \left(\sum_{i=1}^{n} \mathbf{X}(\mathbf{s}_i) \right) - \int_W \mathbf{X}(\mathbf{u}) \lambda_{\boldsymbol{\theta}}(\mathbf{u}) d\mathbf{u}. \tag{3.14}$$

Note that the score is a vector-valued function with components

$$\{\nabla \ell(\boldsymbol{\theta}; \mathbf{s})\}_j = \sum_{i=1}^{n} \mathbf{X}_j(\mathbf{s}_i) - \int_W \mathbf{X}_j(\mathbf{u}) \lambda_{\boldsymbol{\theta}}(\mathbf{u}) d\mathbf{u},$$

for $j = 1, \ldots, p$. The integral on the right-hand side of Equations 3.13 and 3.14 is the Laplace-transform of $\mathbf{X}$ and is not always tractable in point process models, and so often the score cannot be solved analytically. As a consequence, estimation is typically done via an approximation method. This is not the case for Poisson processes; however, approximations are still used for computational purposes. As an MLE, $\boldsymbol{\theta}$ is consistent (*i.e.*, converges in probability to the true $\boldsymbol{\theta}$), asymptotically efficient (*i.e.*, attains the Cramer-Rao lower bound (CRLB) for $\boldsymbol{\theta}$), and asymptotically normal (with mean $\boldsymbol{\theta}$ and variance attaining the CRLB). In particular, Rathbun and Cressie [248] showed asymptotic consistency, normality, and efficiency as $n \to \infty$ and as $W \to \mathbb{R}^d$.

Fine Pixel Approximation One quadrature strategy (see Sec 9.9 of [16] and [14]) for approximating the Poisson process likelihood is to divide the window W into m small pixels (cells) of area, a, and to count the number of points that fall in cell w_j, which we denote $\mathbf{n}_j$. Then, the integral given in Equation 3.13 is approximated by its Riemann sum with midpoint rule:

$$\int_W \lambda_{\boldsymbol{\theta}}(\mathbf{u}) d\mathbf{u} \approx \sum_{j=1}^{m} a\lambda_{\theta}(\mathbf{c}_j), \tag{3.15}$$

where $\mathbf{c}_j$ is the center location of the cell, w_j. Next, we discard the exact location of the data points, $\mathbf{s}$, and move each point to the center of its residing cell, w_j, yielding the approximation

$$\sum_{i=1}^{n} \log \lambda_{\boldsymbol{\theta}}(\mathbf{s}_i) \approx \sum_{j=1}^{J} \mathbf{n}_j \log \lambda_{\boldsymbol{\theta}}(\mathbf{c}_j). \tag{3.16}$$

Combining Equations 3.15 and 3.16 shows the log-likelihood is approximated by

$$\log L(\boldsymbol{\theta}) \approx \sum_{j=1}^{J} [n_j \log \lambda_{\theta}(\mathbf{c}_j) - a\lambda_{\boldsymbol{\theta}}(\mathbf{c}_j)] = \sum_{j=1}^{J} (n_j \log \lambda_{\boldsymbol{\theta}}(\mathbf{c}_j) - a\lambda_{\boldsymbol{\theta}}(\mathbf{c}_j)). \tag{3.17}$$

Notice that the right-hand side of Equation 3.17 is the same form as the log-likelihood of independent random Poisson random variables with mean $a\lambda_{\boldsymbol{\theta}}(\mathbf{u}_j)$. For a log-linear Poisson point process, we now have the following form for the intensity function:

$$\lambda_{\boldsymbol{\theta}}(\mathbf{c}_j) = \exp(B(\mathbf{c}_j) + \boldsymbol{\theta}^\top \mathbf{X}(\mathbf{c}_j)) = \exp(B(\mathbf{c}_j) + \boldsymbol{\theta}^\top \mathbf{X}(\mathbf{c}_j)).$$

Thus, the right-hand side of Equation 3.17 is the log-likelihood of independent Poisson

Algorithm 3.1 FinePixelDataset($\mathbf{s}$, $\mathbf{V}$, W, a)

Input:

$\mathbf{s}$ Point pattern in $\mathbb{R}^d$
W Domain (window) of $\lambda(\mathbf{u})$
$\mathbf{V}$ Basis function set
a Area of cells in division of W

Output:

$\mathbf{Y}$ Vector of counts for cells in Fine-pixel approximation
$\mathbf{X}$ Fixed-effect model matrix

$\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_n\} \leftarrow partitionWindow(a, W)$ ▷ Split window into $n = |W|/a$ cells
for each $i \in 1, \ldots, n$ **do**
  $\mathbf{Y}_{\mathcal{C}_i} \leftarrow |\mathbf{s} \cap \mathcal{C}_i|$ ▷ Count number of points in cell i
  for each $j \in 1, \ldots, |\mathbf{V}|$ **do**
    $\mathbf{D}_{i,j} \leftarrow \mathbf{V}_j(centroid(\mathcal{C}_i))$ ▷ Evaluate basis functions at centroids
return $\mathbf{Y}$, $\mathbf{X}$

random variables with means

$$a\lambda_{\boldsymbol{\theta}}(\mathbf{c}_j) = \exp(\log a + B(\mathbf{c}_j) + \boldsymbol{\theta}^\top \mathbf{X}(\mathbf{c}_j)) = \exp(o_j + \boldsymbol{\theta}^\top \mathbf{X}(\mathbf{c}_j)),$$

where we have an offset term $o_j = \log a + B(\mathbf{c}_j)$.

Equipped with this knowledge, we can use the following procedure for fitting a log-linear Poisson point process model.

1. Divide window W into $n_{\mathcal{C}}$ cells, $\{w_i\}_{i=1}^{n_{\mathcal{C}}}$, of area a.

2. Count the number, $\mathbf{n}_i$, of data points falling into each cell, w_i.

3. For each cell w_i, evaluate each covariate function, $\mathbf{X}_j(\mathbf{c}_i)$, at their centroid, denoted $\mathbf{c}_i$.

4. Use standard statistical software to estimate a log-linear Poisson regression model with responses $\mathbf{y} = \{\mathbf{n}_i\}_{i=1}^{n_{\mathcal{C}}}$ and covariates $\mathbf{X} = (\mathbf{X}(\mathbf{c}_1), \ldots, \mathbf{X}(\mathbf{c}_{n_{\mathcal{C}}}))$.

Algorithm 3.2 ReplicatedFinePixelDatasetWGroups($\mathbf{S}$, $\mathbf{g}$, $\mathbf{V}$, W, a)

Input:

- $\mathbf{S}$ Collection of Point patterns $\{\mathbf{s}_1, \ldots, \mathbf{s}_R\}$ in $\mathbb{R}^d$
- $\mathbf{g}$ Vector of group labels $\mathbf{g}_i \in 0, \ldots, G$
- $\mathbf{V}$ Basis function set
- W Domain (window) of λ
- a Area of cells in division of W

Output:

- $\mathbf{Y}$ Vector of counts for cells in Fine-pixel approximation
- $\mathbf{X}$ Fixed-effect model matrix

$\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_n\} \leftarrow partitionWindow(a, W)$ ▷ Split window into $n = |W|/a$ cells
$\mathbf{Y} \leftarrow \emptyset$
$\mathbf{X} \leftarrow \emptyset$
for each $r \in 1, \ldots, R$ **do**
  $\mathbf{Y}^r \leftarrow \mathbf{0}_n$
  $\mathbf{X}^r \leftarrow \mathbf{0}_{n,|\mathbf{V}|}$
  for each $i \in 1, \ldots, n$ **do**
    $\mathbf{Y}^r_i \leftarrow |\mathbf{p}_r \cap \mathcal{C}_i|$ ▷ Count number of points in $\mathbf{p}_r$ in cell i
    for each $j \in 1, \ldots, |\mathbf{V}|$ **do**
      $\mathbf{X}^r_{i,g_r \times |V| + j} \leftarrow \mathbf{V}_j(centroid(\mathcal{C}_i))$ ▷ Evaluate basis functions at centroids
  $\mathbf{Y} \leftarrow \begin{bmatrix}\mathbf{Y} & \mathbf{Y}^r\end{bmatrix}^\top$
  $\mathbf{X} \leftarrow \begin{bmatrix}\mathbf{X} & \mathbf{X}^r\end{bmatrix}^\top$
return $\mathbf{Y}$, $\mathbf{X}$

5. Approximate the maximum likelihood estimates for the log-linear Poisson point process model using the fitted coefficients, $\hat{\boldsymbol{\theta}}$, from the log-linear Poisson regression model.

6. Use estimated coefficients and basis functions to construct the estimated intensity measure, $\hat{\lambda}(u)$.

This process is summarized in further detail in Algorithm 3.3. The process of constructing the design matrix $\mathbf{X}$ and response vector $\mathbf{Y}$ for the log-linear Poisson regression model is given in Algorithms 3.1 and 3.2 for the single and replicated cases, respectively. Essentially, we construct a design matrix $\mathbf{X}$ by evaluating the basis functions at the centroids of the cells

Algorithm 3.3 LogLinearModelEstimation($\mathbf{s}$, $\mathbf{V}$, W, a, $\mathbf{u}$)

Input:

- $\mathbf{s}$ Point pattern in $\mathbb{R}^d$
- $\mathbf{V}$ Basis function set
- W Domain (window) of λ
- a Area of cells in division of W
- $\mathbf{u}$ Test point to evaluate intensity function

Output:

- $\hat{\lambda}(\mathbf{u})$ Estimaged Intensity at test points $\mathbf{u}$

1: $\mathbf{X}, \mathbf{Y} \leftarrow$ `FinePixelDataset`$(\mathbf{s}, W, \mathbf{V}, a)$ ▷ See Algorithms 3.1 and 3.2
2: $\hat{\boldsymbol{\theta}} \leftarrow GLM(\mathbf{Y}, \mathbf{X})$ ▷ Estimate coefficients w/ log-linear Poisson model
3: $\hat{\lambda}(\mathbf{u}) \leftarrow \exp\left(\hat{\boldsymbol{\theta}}^\top \mathbf{V}(\mathbf{u})\right) \times \frac{|W|}{a}$ ▷ Backtransform coefficients to evaluate intensity
4: **return** $\hat{\lambda}(\mathbf{u})$

in the fine-pixel approximation and construct a response vector $\mathbf{Y}$ by counting the number of points that fall in each cell.

Mixed-Effect Replicated Point Processes Models Mixed-effect models [232] are a class of models that are used to model data with various dependency structures, such as repeated measures data, longitudinal data, and hierarchical data. They allow us to model point process data with multiple replicates, where each replicate is an independent realization of a point process. In particular, we can model the logs of the intensity measures for the replicated point process data via a mixed-effect model, and estimate parameters through maximization of the pseudolikelihood. In the mixed-effect model setting, the logs of the intensity measures for the replicated point process data are modeled via a mixed-effect model and parameters are estimated through maximization of the pseudolikelihood.[2] In particular, a quadrature scheme is used to estimate local intensities as in the Poisson log-linear model described in the previous

[2]Typically, the pseudolikelihood is more tractable than the likelihood, however, in the case of an inhomogeneous Poisson point process, the pseudolikelihood is the same as the likelihood as given by the Janossey product density

section. We briefly summarize this construction of the mixed-effect model for point patterns developed by Grunwald and Bell [23] who used a Berman-Turner approximation to the pseudolikelihood to fit the model. Note that in our work we use the fine-pixel approximation instead of the Berman-Turner approximation, but the two approaches are similar in nature.

Consider a set of R point patterns, $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_R\}$. Each individual point pattern, $\mathbf{s}_r$ (replicate $r \in 1, \ldots R$), consists of a set of locations $\{\mathbf{s}_{r,1}, \ldots, \mathbf{s}_{r,n(\mathbf{x}_r)}\}$, where $n(\mathbf{s}_r)$ are the number of points in point pattern $\mathbf{s}_r$. Before considering the replicated case, let us briefly recall the likelihood approach for a single replicate. The likelihood is defined as

$$L(\boldsymbol{\theta}; \mathbf{s}_r) = \left(\prod_{i=1}^{n(\mathbf{s}_r)} \lambda_{\boldsymbol{\theta}}(\mathbf{s}_{r,i}; \mathbf{x}_r) \right) \exp\left(- \int_W \lambda_{\boldsymbol{\theta}}(\mathbf{u}; \mathbf{s}_r) d\mathbf{u} \right), \tag{3.18}$$

where $\lambda_{\boldsymbol{\theta}}(\mathbf{u}_i; \mathbf{s}_r)$ is the conditional intensity [224] surface of the point process for replicate $\mathbf{s}_r$ parameterized by vector $\boldsymbol{\theta}$ and evaluated at point $\mathbf{u}$.

In the log-linear model, we model the log of the intensity surface as a linear function of parameters $\boldsymbol{\theta}$ at point $\mathbf{u}$ via

$$\lambda_{\boldsymbol{\theta}}(\mathbf{u}; \mathbf{s}_r) = \exp\left(\boldsymbol{\theta}^\top \mathbf{X}(\mathbf{u}; \mathbf{s}_r) \right),$$

where $\mathbf{X}^\top(\mathbf{u}; \mathbf{s}_r)$ is a vector of (possibly spatially varying) covariates for any point $\mathbf{u} \in W$ for replicate $\mathbf{s}_r$. Taking a maximum-likelihood based approach, we can estimate parameter vector $\boldsymbol{\theta}$ by maximizing the likelihood [17]. That is, we have

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta} | \mathbf{s}_r).$$

Similar to the fine-pixel approximation, Berman and Turner [29] developed an approximation to the likelihood that converts the problem into a generalized Poisson log-linear model. The integral in Equation 3.18 can be approximated via numerical quadrature. The pattern

window, W, is discretized into a $k \times k$ grid with dummy points, w_j, being placed at the center of each grid. We then have

$$\int_W \lambda_{\boldsymbol{\theta}}(\mathbf{u}; \mathbf{s}_r) \approx \sum_{j=1}^{m} \lambda_{\boldsymbol{\theta}}(\mathbf{u}_j; \mathbf{s}_r) w_j,$$

where $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ is the set of points containing our data $\mathbf{s}_r$, and the $k \times k$ dummy points in W (*i.e.*, $m = k \times k + n(\mathbf{s}_r)$). The log of the likelihood in Equation 3.18 is then approximated by

$$\log\ L(\boldsymbol{\theta}; \mathbf{s}_r) \approx \sum_{j=1}^{m} (y_j \log \lambda_{\boldsymbol{\theta}}(\mathbf{u}_j; \mathbf{s}_r) - \lambda_{\boldsymbol{\theta}}(\mathbf{u}_j; \mathbf{s}_r)) w_j,$$

which is the log-likelihood for a weighted log-linear Poisson regression model.

Now, reconsider our set of point patterns, $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_R\}$, and let $\lambda_{\boldsymbol{\theta}}(\mathbf{u}; \mathbf{s}_r)$ be the conditional intensity for replicate $\mathbf{s}_r$ at location $\mathbf{u}$. We model this intensity log-linearly as the mixed effect model

$$\log(\lambda_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\mathbf{u}; \mathbf{s}_r)) = \boldsymbol{\theta}^\top \mathbf{X}(\mathbf{u}; \mathbf{s}_r) + \boldsymbol{\gamma}^\top \mathbf{Z}(\mathbf{u}; \mathbf{s}_r),$$

where $\mathbf{X}$ is a fixed-effect model vector, $\mathbf{Z}$ is the random-effect model vector, and $\boldsymbol{\gamma}$ is a vector of random-effect parameters. We make the assumption that $\boldsymbol{\gamma}^\top \sim N(\mathbf{0}, \mathbf{G})$. Now, we write the log-likelihood over all replicates as

$$\log L(\boldsymbol{\theta}, \boldsymbol{\Gamma}; \mathbf{S}) = \sum_{r=1}^{R} \left[\left(\sum_{i=1}^{n(\mathbf{s}_r)} \mathbf{X}^\top(\mathbf{s}_{r,i}; \mathbf{s}_r)\boldsymbol{\theta} + \mathbf{Z}^\top(\mathbf{s}_{r,i}; \mathbf{s}_r)\boldsymbol{\gamma} \right) - \int_W \exp\left(\mathbf{X}^\top(\mathbf{u}; \mathbf{s}_r)\boldsymbol{\theta} + \mathbf{Z}^\top(\mathbf{u}; \mathbf{s}_r)\boldsymbol{\gamma} \right) d\mathbf{u} \right],$$

where $\boldsymbol{\Gamma}$ contains the parameters associated with $\mathbf{G}$. Finally, after considering the Berman-

Turner approximation, we obtain

$$\log L(\boldsymbol{\theta}, \boldsymbol{\Gamma}; \mathbf{s}) \approx \sum_{j=1}^{M} (y_j \log \ \lambda_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\mathbf{u}_j; \mathbf{s}) - \lambda_{\boldsymbol{\theta},\boldsymbol{\gamma}}(\mathbf{u}_j; \mathbf{s})) w_j, \tag{3.19}$$

where $M = \sum_{s=1}^{S} m_s$ is the total number of quadrature points over all point processes. Here, notice that Equation 3.19 is the weighted log-likelihood for a Poisson log-linear mixed effect model and thus our parameters in $\boldsymbol{\theta}$ can be estimated using standard software for Generalized Linear Mixed Model (GLMM) modeling (*e.g.*, using a Penalized Quasi-Likelihood (PQL) based approach [41]).

Estimation of the replicated point pattern Poisson log-linear model using a fine-pixel approximation and B-spline parameterization is essentially no different than the fixed-effect case described in Section 3.3.2. We first discretize the spatial window, W, into a grid of cells and count the number of points in each cell for each replicate. We parameterize the surface $\lambda(\mathbf{u})$ by choosing knot vector, $\mathbf{U}$ (or multiple knot vectors depending on dimension of surface), that spans the window and evaluate the basis functions, $\mathbf{B}(\mathbf{u})$, at the centroids of each cell in $\mathcal{C}$ to construct model matrix, $\mathbf{X}$. Additionally, we form the count vector, $\mathbf{Y}$, by counting the number of points in each cell. We then use standard GLMM software, such as the `lme4` package [19] in R, to estimate the parameters, $\boldsymbol{\theta}$, and backtransform the coefficients to obtain the estimated intensity surfaces. We will compare this approach experimentally to the nonparametric hierarchical mixed-curve model presented in the next section. This estimation procedure is also used in the local setting we describe in the upcoming methods section. However, in that case, we use a low-order polynomial to parameterize the intensity surface locally instead of B-splines globally.

3.3.3 Multiple Testing

In our method, we compute point estimates of the intensity functions over a grid of points in the spatial domain along with pointwise p-values for testing the null hypothesis of no difference in intensity between groups at each point. However, since we are performing multiple hypothesis tests across the spatial domain, we need to account for the inflation of Type I error due to the multiple comparisons being made by controlling the family-wise error rate (FWER) of our testing procedure. To do this, we will use multiple testing corrections. In this section, we briefly review several multiple testing correction procedures that we consider in our experiments. The main approach we consider is the Westfall-Young procedure [301] as adapted to the functional data setting by Cox and Lee [62]. We also consider the Romano-Wolf procedure [258] as another resampling-based approach to controlling the family-wise error rate (FWER) in multiple hypothesis testing.

Bonferroni correction A good starting point for multiple testing correction is the Bonferroni correction [35], a method that controls for the family-wise error rate (FWER) in multiple hypothesis testing. Let $H_1, \ldots, H_m$ be a collection of m hypothesis tests with corresponding p-values $p_1, p_2, \ldots, p_m$. The FWER is defined as the probability of making one or more Type I errors (rejecting a true null hypothesis) among the m tests:

$$\text{FWER} = P\left(\bigcup_{j=1}^{m}\{\text{Reject } H_j | H_j \text{ is true}\}\right).$$

To control the FWER at level α, the Bonferroni correction adjusts the significance level for each individual test to be α/m. Equivalently, we can adjust the p-values by multiplying them by m and then compare the adjusted p-values, $\tilde{p}_j = \min(mp_j, 1)$, to α. That is, if $\tilde{p}_j < \alpha$, we reject hypothesis H_j. It can be shown that the Bonferroni correction bounds the FWER by

using Boole's inequality:

$$FWER = P\left(\bigcup_{j=1}^{m}\{\text{Reject } H_j | H_j \text{ is true}\}\right) \leq \sum_{j=1}^{m} P(\text{Reject } H_j | H_j \text{ is true}) \leq m \cdot \frac{\alpha}{m} = \alpha.$$

The Bonferroni correction is simple and easy to implement. However, it is also known to be conservative, especially when the tests are highly correlated [138], as it is derived assuming all tests are independent.

Holm-Bonferroni procedure An improvement over the Bonferroni correction is the Holm-Bonferroni procedure [137]. The Holm-Bonferroni procedure is a step-wise approach to controlling FWER. We begin by obtaining the m p-values for the m hypothesis tests and then sort them in increasing order: $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$, so that (i) gives the rank in $p_1, \ldots, p_m$. We then compare each p-value to a sequentially adjusted significance level. In particular, we compare $p_{(1)}$ to α/m, $p_{(2)}$ to $\alpha/(m-1)$, and so on, until we reach $p_{(m)}$, which is compared to α. We reject all hypotheses $H_{(j)}$ for which $p_{(j)} \leq \alpha/(m-j+1)$. In this way, the Holm-Bonferroni procedure is considered to be a step-down version of the Bonferroni correction, starting with the same adjusted smallest p-value, but applying a slightly smaller penalty to other p-values, which results in more power than the Bonferroni correction.

Hochberg procedure The Hochberg procedure [134] is a FWER adjustment closely related to the Holm-Bonferroni procedure providing a step-up version of the Bonferroni method. As with the Holm-Bonferroni procedure, we begin by obtaining the m p-values for the m hypothesis tests and then sort them in increasing order: $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$. If $p_{(m)} < \alpha$, we reject all hypotheses $H_{(1)}, \ldots, H_{(m)}$. Otherwise, we find the largest k such that $p_{(k)} < \alpha/(m-k+1)$ and reject all hypotheses $H_{(1)}, \ldots, H_{(k)}$.

	Reject H_0	**Do not Reject H_0**	**Total**
H_0 **True**	V	U	m_0
H_0 **False**	S	T	$m - m_0$
Total	R	$m - R$	m

Table 3.1: Multiple Hypothesis Testing Contingency Table.

Hommel procedure The Hommel procedure [138] is a step-wise approach to controlling the family-wise error rate (FWER) in multiple hypothesis testing. In the Hommel procedure, we begin by obtaining the m p-values for the m hypothesis tests and then sort them in increasing order: $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$. We then compare each p-value to a sequentially adjusted significance level. That is, we compare $p_{(j)}$ to α/k for each $j = 1, 2, \ldots, m$, where k is the largest integer j such that $p_{(m-k+j)} > j\alpha/k$ for all $j = 1, 2, \ldots, k$. We reject all hypotheses $H_{(j)}$ for which $p_{(j)} < \alpha/k$.

Benjamini-Hochberg procedure The Benjamini-Hochberg procedure [26] is a widely used approach to controlling the false discovery rate (FDR) in multiple hypothesis testing. The false discovery rate is defined as the expected proportion of false positives among all rejected hypotheses. In particular, consider Table 3.1 which summarizes the outcomes of multiple hypothesis tests. V is the number of false positives (Type I errors), S is the number of true positives, U is the number of true negatives, and T is the number of false negatives (Type II errors). The false discovery rate (FDR) is defined as

$$FDR = \mathbb{E}[Q] = \mathbb{E}[V/(V+S)] = E[V/R]$$

where we define $Q = V/R$ if $R > 0$ and $Q = 0$ if $R = 0$. Let q be a prespecified level upper-bound for the FDR (often 0.05). As with the previous FWER methods, we begin by

obtaining the m p-values for the m hypothesis tests and then sorting them in increasing order: $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$. We then compare each p-value to a sequentially adjusted significance level. That is, we compare $p_{(j)}$ to $(j/m)q$ for each $j = 1, 2, \ldots, m$. We reject all hypotheses $H_{(j)}$ for which $p_{(j)} < (j/m)q$. The procedure controls the FDR at level $q \cdot m_0/m$ under independent or positively dependent tests, where m_0 is the number of true null hypotheses.

Benjamini-Yekutieli procedure The Benjamini-Yekutieli procedure [27] is a two-stage adaptive extension of the Benjamini-Hochberg procedure that controls the false discovery rate (FDR). We again begin by obtaining the m p-values for the m hypothesis tests and then sorting them in increasing order: $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$. Next, we compute the constant k as the largest integer j such that $p_{(j)} \leq (j/m)(q/\sum_{i=1}^{m} 1/i)$. If no such j exists, we reject no hypotheses. Otherwise, we reject hypotheses $H_{(1)}, \ldots, H_{(k)}$. The procedure controls the FDR at level $q \cdot m_0/m$ under arbitrary dependence among the tests, where m_0 is the number of true null hypotheses.

Westfall-Young procedure Cox and Lee [62] proposed using the Westfall-Young procedure to adjust p-values for multiple hypothesis testing in the context of functional data. The Westfall-Young procedure [301] is a step-wise resampling approach to controlling the family-wise error rate (FWER) in multiple hypothesis testing. In the functional data analysis setting, we may observe a function, $y(x)$, over some finite collection of points $\mathbf{q} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m\} \subset W$. We can obtain locally valid p-values for each of the m locations, so that we can then use a step-wise approach to control the FWER of that collection of p-values. As described in Algorithm 3.4, we begin by obtaining the m p-values for the m locations and then sort them in increasing order: $\mathbf{p}_{(1)} \leq \mathbf{p}_{(2)} \leq \cdots \leq \mathbf{p}_{(m)}$. Let π be the permutation of the indices $1, 2, \ldots, m$ such that $\mathbf{p}_{(j)} = \mathbf{p}_{\pi(j)}$. That is, π is the permutation of the indices that sorts the p-values in increasing order.

Next, randomly permute the data in a subset pivotal way, resulting in a new collection

of p-values p^*. Note that a distribution P has the subset pivotality property if the joint distribution of the subvector $P_i : i \in I$ is identical under the restrictions $\cap_{i \in I} H_{0i}$ and H_0^C, the complete null hypothesis, for all subsets I of $\{i_1, i_2, \ldots, i_j\}$ of true null hypotheses [301]. In other words, the distribution of the test statistics for any subset of true null hypotheses is the same under the global null and the subset null. This assumption allows us to use resampling under the complete null hypothesis H_0^C as opposed to the partial hypothesis H_0^K.

Now, place the p^*-values in the same order as the original p-values, *i.e.*, $p^*_{(j)} = p^*_{\pi(j)}$. We repeat this process B times and collect all of the randomization p-values into a 2d array $\mathbf{p}^*$. We then transform the $\mathbf{p}^*$ values into another 2d array of $\mathbf{q}^*$-values by computing the minimum of the $\mathbf{p}^*$ values that are greater than $\mathbf{p}_{(j)}$ for each j. That is, $\mathbf{q}^*_{(j),\ell} = \min\{\mathbf{p}^*_{(s),\ell} : s > j\}$ for each j and ℓ. Finally, we compute the p-values for each of the m locations by computing the proportion of the B randomization p-values that are less than the original p-values. That is, we compute

$$r_{(j)} = \frac{1}{B} \sum_{\ell=1}^{N} I(\mathbf{q}^*_{(j),\ell} < \mathbf{p}_{(j)}),$$

which provide the m adjusted p-values.

The Westfall-Young procedure is a step-wise approach to controlling the family-wise error rate (FWER) in the multiple hypothesis testing being performed. In other words, with the Westfall-Young approach, we compute p-values at each of the m locations and then adjust them to control the FWER.

Romano-Wolf procedure Romano and Wolf [258] proposed a step-down method similar to the Westfall-Young procedure that relaxes the subset pivotality assumption in favor of a randomization hypothesis. Here, we let $\mathbf{G}$ be a group of transformations acting on the data $\mathbf{X}$ and we assume that the distribution of $\mathbf{X}$ is invariant under the transformations in $\mathbf{G}$ (*i.e.*, $\mathbf{X} \stackrel{d}{=} g\mathbf{X}$ for all $g \in \mathbf{G}$). This has the advantage of allowing for more general hypothesis tests to be performed based on the bootstrap, permutation, or other resampling techniques. Our

Algorithm 3.4 Westfall-Young for Functional Data (Cox and Lee, 2008)

Input:

$\mathcal{D}$ Data
p Sorted p-values for each location
π Indices

Output:

r Adjusted p-values for each location

---

for ℓ in 1 to N **do**
  $\mathcal{D}^* \leftarrow SubsetPivotalPermutation(\mathcal{D})$
  $p^* \leftarrow ComputePVals(\mathcal{D}^*)$
  $p^*_{(j),\ell} \leftarrow p^*\{\pi(j)\}$
for j in $1 \dots m$ **do**
  for ℓ in $1 \dots B$ **do**
    $q^*_{(j),\ell} \leftarrow \min\{p^*_{(s),\ell} : s > j\}$
  $r_{(j)} \leftarrow N^{-1} \sum_{\ell=1}^{N} I\left(q^*_{(j),\ell} < p_{(j)}\right)$
return r

---

description here follows the approach of Romano and Wolf [258]. Rather than ordering the p-values, the Romano-Wolf procedure focuses on ordering test statistics. Let $\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_m$ be a collection of m test statistics corresponding to m hypothesis tests $H_1, H_2, \dots, H_m$. Note that one can use $\mathbf{T}_j = 1 - \mathbf{p}_j$ as the test statistic for hypothesis H_j if $\mathbf{p}_j$ is the p-value for that test. We begin by sorting the test statistics in decreasing order: $\mathbf{T}_{(1)} \geq \mathbf{T}_{(2)} \geq \cdots \geq \mathbf{T}_{(m)}$. Similar to the Westfall-Young procedure, let π be the permutation of the indices $1, 2, \dots, m$ such that $\mathbf{T}_{(j)} = \mathbf{T}_{\pi(j)}$. That is, π is the permutation of the indices that sorts the test statistics in *decreasing* order.

In the step-down procedure, we are interested in testing the intersection hypothesis $H_K = \cap_{j \in K} H_j$ for $K_1 = \{\pi(1), \pi(2), \dots, \pi(m)\}$, $K_2 = \{\pi(2), \pi(3), \dots, \pi(m)\}$, and so on. Randomly sample B samples of the data by applying B transformations from $\mathbf{G}$ to the data,

resulting in B collections of test statistics $\mathbf{T}$. Let

$$\tilde{p}_{K_j} = \frac{1}{B}\left[1 + \sum_{i=1}^{B-1} I\left(T_{K_j}(g_i\mathbf{X}) \geq T_{K_j}(\mathbf{X})\right)\right].$$

Then, at stage j, we test the intersection hypothesis H_{K_j} and reject if $\tilde{p}_{K_j} \leq \alpha$ for $K_j = \{(j), (j+1), \ldots, (m)\}$.

The Romano-Wolf procedure is similar to the Westfall-Young approach in that it provides FWER control using a randomization approach, but differs in the level where the adjustment is developed (*i.e.*, test statistic vs. p-value) and the assumptions made (*i.e.*, subset pivotality vs. randomization hypothesis). This provides some advantages for the Romano-Wolf procedure in terms of allowing for more general hypothesis tests to be performed based on the bootstrap, permutation, or other resampling techniques. However, the Romano-Wolf procedure has not proven to control FWER in the functional data setting, while the Westfall-Young procedure has. Nonetheless, we will also see the Romano-Wolf procedure performs fairly well in most settings along with some of the other multiple comparison procedures reviewed above, but it also performs poorly in some settings.

Now that we have introduced each of the multiple testing correction procedures, we will use them to analyze the pointwise p-values obtained from our local mixed-curve models described in the methods section. Locally, we will be fitting a multi-level mixed-effect model and obtaining pointwise p-values for testing the null hypothesis of no difference between groups at each point. Thus, our final description of background material for our method will be to review the multi-level modeling tools we will use to analyze the fitted local mixed-curve models.

3.3.4 Multi-level Modelling

Because the local mixed-curve models represent curves hierarchically, we end up estimating a multi-level model pointwise across the domain. Consequently, we also use

some tools from hierarchical/multi-level modeling to analyze the fitted models pointwise. In particular, we use the intra-class correlation coefficients (ICCs) to assess locally how similar the curves are within the same group compared to curves in different levels of the hierarchical model. We also use pointwise hypothesis tests to test if the there are population level differences and group level differences locally in the fitted models. We briefly review these concepts here but refer the reader to texts by Gelman and Hill [107], Snijders and Bosker [271] and Goldstein [116] for more detailed reviews of hierarchical/multi-level modeling.

Multilevel models, also known as hierarchical linear models, use mixed-effect models to decompose the variance of a response variable attributable to different levels of a sampling hierarchy. For example, in a simple two-level model, we may have observations nested within groups, and we may be interested in understanding how much of the variance in the response variable is attributable to differences between groups versus differences within groups. In a more complex three-level model, we may have observations nested within groups, which are in turn nested within higher-level units, and we may be interested in understanding how much of the variance in the response variable is attributable to differences between groups, differences between higher-level units, and differences within groups. A classic example of a three-level model is students nested within classrooms, which are in turn nested within schools. In this example, we may be interested in understanding how much of the variance in student test scores is attributable to differences between students, differences between classrooms, and differences between schools. We might be interested in understanding how similar students are within the same classroom compared to students in different classrooms, which is captured by the intra-class correlation coefficient (ICC). Further, we may be interested in estimation of a treatment effect, say some new learning method that is being introduced at some level of the hierarchy (student, classroom, or school). We may then want to test the significance of that treatment effect while accounting for the variability structured by the hierarchical structure of the data.

We will build up a multi-level model to illustrate these concepts as they will also apply to our local mixed-curve model. First, consider a two-level model with observations y_{ij} nested within groups i for $i = 1, \ldots, m$ and $j = 1, \ldots, n_i$. We have a single covariate x_{ij} and we want to model the relationship between y_{ij} and x_{ij} while accounting for the fact that observations are nested within groups and allowing the relationship between y_{ij} and x_{ij} to vary across groups. We will model the relationship between y_{ij} and x_{ij} using a linear mixed-effect model with a random intercept and a random slope for x_{ij} across groups.

We can write the model as

$$y_{ij} = \beta_0 + \beta_1 x_{ij} + u_{0i} + u_{1i}\epsilon_{ij}$$

where we assume $\begin{bmatrix} u_{0i} \\ u_{1i} \end{bmatrix} \sim N(\mathbf{0}, \mathbf{G}_i)$ where $\mathbf{G}_i = \begin{bmatrix} \sigma^2_{u0} & \sigma_{u0}\sigma_{u1} \\ \sigma_{u0}\sigma_{u1} & \sigma^2_{u1} \end{bmatrix}$ and $\epsilon_{ij} \sim N(0, \sigma^2_\epsilon)$. Here, β_0 is the overall (population-level) intercept, β_1 is the overall (population-level) slope relating x_{ij} to y_{ij}, u_{0i} is the additional (*i.e.*, difference away from population-level) random intercept for group i, u_{1i} is the additional random slope relating x_{ij} to y_{ij} for group i, and ϵ_{ij} is the

residual error. In matrix form, we can write the model as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

$$= \underbrace{\begin{bmatrix} 1 & x_{11} \\ 1 & x_{12} \\ \vdots & \vdots \\ 1 & x_{1n_1} \\ 1 & x_{21} \\ 1 & x_{22} \\ \vdots & \vdots \\ 1 & x_{2n_2} \\ \vdots & \vdots \\ 1 & x_{m1} \\ 1 & x_{m2} \\ \vdots & \vdots \\ 1 & x_{mn_m} \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}}_{\boldsymbol{\beta}} + \underbrace{\begin{bmatrix} 1 & x_{11} & 0 & 0 & \dots & 0 & 0 \\ 1 & x_{12} & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_{1n_1} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & x_{21} & \dots & 0 & 0 \\ 0 & 0 & 1 & x_{22} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 1 & x_{2n_2} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & x_{mn_1} \\ 0 & 0 & 0 & 0 & \dots & 1 & x_{mn_2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & x_{mn_m} \end{bmatrix}}_{\mathbf{Z}} \underbrace{\begin{bmatrix} u_{01} \\ u_{11} \\ u_{02} \\ u_{12} \\ \vdots \\ u_{0m} \\ u_{1m} \end{bmatrix}}_{\boldsymbol{\gamma}} + \underbrace{\begin{bmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \vdots \\ \epsilon_{mn_m} \end{bmatrix}}_{\boldsymbol{\epsilon}}$$

where $\mathbf{y}$ is the vector of observations, $\mathbf{X}$ is the fixed-effect model matrix, $\boldsymbol{\beta}$ is the vector of fixed-effect parameters, $\mathbf{Z}$ is the random-effect model matrix, $\boldsymbol{\gamma}$ is the vector of random-effect parameters (where we assume $\boldsymbol{\gamma} \sim N(\mathbf{0}, \mathbf{G})$), and $\boldsymbol{\epsilon}$ is the vector of residual errors (where we assume $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I})$). Here, $\mathbf{G}$ is the covariance matrix of the random effects, which in this case would be a block diagonal matrix with elements equal to the individual covariance matrices $\mathbf{G}_i$ for each group i. That is, we have $\mathbf{G} = \oplus_{i=1}^{m} \mathbf{G}_i$.

Johnson [150] showed that the variability attributable to differences between groups (level u) in this more complicated setting that includes both random intercepts and random

slopes is captured by the mean observation-specific random effect $\sigma^2_{u,ij} = var(u_{0i} + u_{1i}x_{ij})$ for sample in group i and observation j. The random effect variance in this setting is then mean of the individual variances $\bar{\sigma}^2_u = \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n_i}\sigma^2_{u,ij}$, which is also computed as $\bar{\sigma}^2_u = Tr(\mathbf{ZGZ}^\top)/n$. Thus, in this case we can compute the intra-class corrleation coefficient (ICC) for the model as

$$ICC = \frac{\bar{\sigma}^2_u}{\bar{\sigma}^2_u + \sigma^2_\epsilon}$$

which in this case can be interpreted as how similar observations are within the same group compared to observations in different groups. A higher ICC indicates that observations within the same group are more similar to each other than to observations in different groups.

Estimation of the model parameters is carried out via maximum likelihood or restricted (or residual) maximum likelihood (REML) estimation depending on if we are interested in the estimation of fixed effects or random effects, respectively. This can be carried out using a variety of methods; for example, iterative generalized least squares (IGLS) can be used [115], as can Fischer scoring [184], the EM algorithm [180], profile likelihood methods [20], and MCMC sampling [43]. We can test the significance of the fixed effects by comparing the likelihood of a full model with the fixed effect to a reduced model without the fixed effect using a likelihood ratio test on parameters estimated using maximum likelihood estimation. We can also test the significance of the random effects u_{0i} or u_{1i} to determine if there are significant differences between groups. This can be done using a likelihood ratio test comparing the full model with the random effect to a reduced model without the random effect estimated using restricted maximum likelihood estimation.

Now, we can consider a more complex three-level model with observations y_{ijk} nested within groups j, which are in turn nested within higher-level units i. We can write the model as

$$y_{ijk} = \beta_0 + \beta_1 x_{ijk} + u_{0i} + u_{1i}x_{ijk} + v_{0ij} + v_{1ij}x_{ijk} + \epsilon_{ijk}.$$

Grouping all observations from nested group j together, we can write the model in matrix form as

$$\mathbf{y}_{ij} = \beta_0 + \beta_1 \mathbf{x}_{ij} + u_{0i} + u_{1i}\mathbf{x}_{ij} + v_{0ij} + v_{1ij}\mathbf{x}_{ij} + \boldsymbol{\epsilon}_{ij}.$$

Here, we assume that $\begin{bmatrix} u_{0i} \\ u_{1i} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_{u0}^2 & \sigma_{u0}\sigma_{u1} \\ \sigma_{u0}\sigma_{u1} & \sigma_{u1}^2 \end{bmatrix}\right)$, $\begin{bmatrix} v_{0ij} \\ v_{1ij} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_{v0}^2 & \sigma_{v0}\sigma_{v1} \\ \sigma_{v0}\sigma_{v1} & \sigma_{v1}^2 \end{bmatrix}\right)$ and $\epsilon_{ijk} \sim N(0, \sigma_\epsilon^2)$.

Collecting all of the random effects into a single vector $\boldsymbol{\gamma}$ and all of the fixed effects into a single vector $\boldsymbol{\beta}$, we can write the full model across all data in matrix form as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

where $\mathbf{Z} = \begin{bmatrix} \mathbf{Z}^{(u)} & \mathbf{Z}^{(v)} \end{bmatrix}$ gives design matrices for the random effects for level l_u and level l_v, $\boldsymbol{\gamma} \sim N(0, \mathbf{G})$ with $\boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{\gamma}^{(u)} \\ \boldsymbol{\gamma}^{(v)} \end{bmatrix}$ and $\mathbf{G} = \begin{bmatrix} \mathbf{G}^{(u)} & 0 \\ 0 & \mathbf{G}^{(v)} \end{bmatrix}$. In this model, the fixed effects β_0 and β_1 represent the overall (population-level) intercept and slope for x_{ijk}, while the random effects u_{0i} and u_{1i} represent the deviations of the intercept and slope for group i from the overall intercept and slope, respectively, and the random effects v_{0ij} and v_{1ij} represent the deviations of the intercept and slope for nested group j within group i from the intercept and slope for group i, respectively.

For this model, we then have two levels of variability attributable to differences between groups: the variability attributable to differences between groups i (level l_u) and the variability attributable to differences between groups j nested within groups i (level l_v). The variability attributable to differences between groups i is captured by the mean observation-specific random effect $\sigma_{u,ijk}^2 = var(u_{0i} + u_{1i}x_{ijk})$ for observation k in group j nested within group i. The variability attributable to differences between nested groups j within groups i is captured

by the mean observation-specific random effect $\sigma^2_{v,ijk} = var(v_{0ij} + v_{1ij}x_{ijk})$ for sample k in group j nested within group i.

The random effect variance for level l_u is then the mean of the individual variances $\bar{\sigma}^2_u = \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\sigma^2_{u,ijk}$, which is computed as $\bar{\sigma}^2_u = \mathrm{Tr}(\mathbf{Z}^{(u)}\mathbf{G}^{(u)}\mathbf{Z}^{(u)\top})/n$ where $\mathbf{Z}^{(u)}$ is the portion of the random-effect model matrix corresponding to the random effects for level l_u and $\mathbf{G}^{(u)}$ is the portion of the covariance matrix of the random effects corresponding to the random effects for level l_u. The random effect variance for level l_v is then the mean of the individual variances $\bar{\sigma}^2_{l_v} = \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n_i}\sum_{k=1}^{n_{ij}}\sigma^2_{l_v,ijk}$, which is similarly computed as $\bar{\sigma}^2_{l_v} = \mathrm{Tr}(\mathbf{Z}^{(v)}\mathbf{G}^{(v)}\mathbf{Z}^{(v)\top})/n$ where $\mathbf{Z}^{(v)}$ is the portion of the random-effect model matrix corresponding to the random effects for level l_v and $\mathbf{G}^{(v)}$ is the portion of the covariance matrix of the random effects corresponding to the random effects for level l_v. Thus, in this case we can compute the ICC for level l_u as

$$ICC_u = \frac{\bar{\sigma}^2_{l_u}}{\bar{\sigma}^2_{l_u} + \bar{\sigma}^2_{l_v} + \sigma^2_\epsilon}$$

and the ICC for level l_v as

$$ICC_v = \frac{\bar{\sigma}^2_{l_u} + \bar{\sigma}^2_{l_v}}{\bar{\sigma}^2_{l_u} + \bar{\sigma}^2_{l_v} + \sigma^2_\epsilon}$$

where ICC_u can be interpreted as how similar observations are within the same group i compared to observations in different groups i, and ICC_v can be interpreted as the how similar observations are within the same nested group j within group i compared to observations in different nested groups j within group i. We can also test the significance of the fixed effects and random effects in this model using likelihood ratio tests as described above for the two-level model.

3.4 Methods

In this section, we describe a local kernel-based methodology for the analysis of non-independent persistent diagram (replicated point pattern) data under a mixed-curve model. Essentially, instead of globally parameterizing the intensity functions with basis functions, we give a local polynomial parameterization to intensity functions. Our description follows a simple repeated measures design with a single categorical covariate, but the described method can be extended to more complex designs.

Essentially, we apply the Generalized Random Curve (GRC) (also referred to as Generalized Non-Parametric Mixed-Effect (GNPME)) model of Cai and Wu [49] to estimate the intensity functions of a replicated Poisson point process using a fine-pixel approximation. We apply this method to persistence diagram point pattern data, treating the data as coming from a replicated inhomegeneous Poisson process. This method involves fitting local models for each query point of interest. Thus, we obtain pointwise estimates of the intensity function along with associated p-values of interest and other parameter estimates for each query point. We then use these pointwise p-values and a multiple testing adjustment procedure to construct local and global hypothesis tests for mixed curve models. Finally, we obtain further information about the model by estimating pointwise mean, variance and intra-class correlation coefficient (ICC) surfaces for the fitted model.

We begin with a collection of persistence diagrams $\{\mathcal{D}_{ij}\}_{i,j=1}^{n,n_i}$, where $\mathcal{D}_{ij}$ is the persistence diagram of the jth replicate of the ith individual, resulting in a total of $N = \prod_{i=1}^{n} n_i = |\{\mathcal{D}_{ij}\}|$ diagrams. Each diagram $\mathcal{D}_{ij}$ is a finite collection of points in the plane $\{\mathbf{s}_{ijk}\}_{k=1}^{K_{ij}} \subset \mathbb{R}^2$ with the kth point in diagram $\mathcal{D}_{ij}$ denoted as $\mathbf{s}_{ijk} = (\mathbf{b}_{ijk}, \mathbf{d}_{ijk})$, where $\mathbf{b}_{ijk}$ is the birth time and $\mathbf{d}_{ijk}$ is the death time of the kth topological feature in the diagram. Associated to each diagram is an observed categorical covariate, G_{ij}, whose value is one of L levels. Further,

restrict[3] all of the diagrams to a common region, $W \subset \mathbb{R}^2$.

We then model the collection of persistence diagrams, $\{\mathcal{D}_{ij}\}_{i,j=1}^{n,n_i}$ as a replicated inhomogeneous Poisson point process with conditional intensity function, $\lambda_{ij}(\mathbf{s}|G_{ij} = g_{ij})$. For notational convenience, we represent birth and death times as a single point in $\mathbb{R}^2$ as $\mathbf{s} = (b, d)$ and drop the condition on G_{ij} to simply write $\lambda_{ij}(\mathbf{s})$ for the intensity function. We estimate the intensity functions $\lambda_{ij}(\mathbf{s})$ using a generalized nonparametric mixed-curve model and a fine-pixel approximation (see Section 3.3.2), which provide the smooth intensity function estimates for each observation j diagram on subject i.

3.4.1 Nonparametric Mixed-Curve Model

To extend this to the nonparametric setting, we follow an approach due to Cai and Wu [49], presenting it in the multidimensional case with a categorical predictor variable and an additional level of nesting in how observations are obtained. Using a fine-pixel approximation, we bin our diagrams, $\{\mathcal{D}_{ij}\}_{i,j=1}^{n,n_i}$, over W into m equally sized bins, $\{\mathbf{b}_{ijk}\}_{k=1}^{m}$, with volumes a, centroids at $\{\mathbf{c}_{ijk}\}_{k=1}^{m}$, and counts of $\{\mathbf{y}_{ijk}\}_{k=1}^{m}$. This provides us with spatially (and hierarchically) indexed data $(\mathbf{y}_{ijk}, \mathbf{c}_{ijk})$ (along with associated covariate G_{ij}) for each individual i, replicate j, and location $\mathbf{c}_{ijk}$. Conditional on subject i and replicate j, the marginal mean and variance are $E[\mathbf{y}_{ijk}|\mathbf{c}_{ijk}]/a = \lambda_{ij}(\mathbf{c}_{ijk}) = \boldsymbol{\mu}_{ijk}$ and $Var[\mathbf{y}_{ijk}|\mathbf{c}_{ijk}] = \phi \mathbf{w}_{ijk}^{-1} V(\boldsymbol{\mu}_{ijk})$, where ϕ is a scale parameter, $\mathbf{w}_{ijk}$ are weights, and $V(\mu_{ijk}) = \boldsymbol{\mu}_{ijk}$ is the known variance function for the Poisson distribution.

Consider a generic point $\mathbf{u} \in W$. We are ultimately interested in estimating and conducting inference on the following nonparametric model:

$$\eta_{ijk} = \log\left(\lambda_{ij}(\mathbf{u})\right) = \alpha_1(\mathbf{u}) + \sum_{l=2}^{L} \alpha_{\Delta_l}(\mathbf{u}) I(g_{ij} = l) + \nu_i(\mathbf{u}) + \zeta_{ij}(\mathbf{u}). \tag{3.20}$$

[3] This is not technically necessary, but is done for notational convenience. Mathematically, we would need to adjust the likelihood by integrating over the union of regions.

Here, we model the log of the intensity function associated with the diagram of the ith individual and the jth replicate as a sum of three component functions. The first component is comprised of the population-level functions $\alpha_l(\mathbf{u})$ that can vary across the $l = 1, \ldots, L$ levels of the categorical variable G_{ij}, which are constructed from the population function for group 1, $\alpha_1(\mathbf{u})$, and the deviation functions for groups 2 through L, $\alpha_{\Delta_l}(\mathbf{u})$. The second and third components are the individual deviation away from the population-level, denoted $\nu_i(\mathbf{u})$, that vary across individuals, and the replicate deviation from the individual level surface, $\zeta_{ij}(\mathbf{u})$, that vary across replicates.

Assume that locally at each point we can approximate each of the surfaces in Equation 3.20 via Taylor expansion. Let $f(\mathbf{u})$ be a scalar valued function of the $m \times 1$ dimensional vector $\mathbf{u}$ and denote the partial derivatives of $f(\mathbf{u})$ evaluated at $\mathbf{u} = \mathbf{u}_0$ by

$$
\begin{aligned}
f^{(1)}_{\mathbf{u}_0} &= \frac{\partial f(\mathbf{u})}{\partial \mathbf{u}}\bigg|_{\mathbf{u}=\mathbf{u}_0}, \\
f^{(2)}_{\mathbf{u}_0} &= \frac{\partial^2 f(\mathbf{u})}{\partial \mathbf{u}^\top \otimes \partial \mathbf{u}}\bigg|_{\mathbf{u}=\mathbf{u}_0}, \\
f^{(3)}_{\mathbf{u}_0} &= \frac{\partial^3 f(\mathbf{u})}{\partial \mathbf{u}^\top \otimes \partial \mathbf{u}^\top \otimes \partial \mathbf{u}}\bigg|_{\mathbf{u}=\mathbf{u}_0} \\
&\vdots = \vdots \\
f^{(p)}_{\mathbf{u}_0} &= \frac{\partial^p f(\mathbf{u})}{\underbrace{\partial \mathbf{u}^\top \otimes \cdots \otimes \partial \mathbf{u}^\top}_{p-1 \text{ times}} \otimes \partial \mathbf{u}}\bigg|_{\mathbf{u}=\mathbf{u}_0}.
\end{aligned}
$$

where $\otimes$ denotes the Kronecker product. For $p \geq 2$, the dimension of $f^{(p)}_{\mathbf{u}_0}$ is $m \times m^{p-1}$. Denote the powers of $(\mathbf{u} - \mathbf{u}_0)$ as $(\mathbf{u} - \mathbf{u}_0)^{\otimes l} = \bigotimes_{i=1}^{l} (\mathbf{u} - \mathbf{u}_0)$. Then, by Taylor's theorem, for $\mathbf{u}$ in neighborhood of $\mathbf{u}_0$, we have

$$
f(\mathbf{u}) \approx f(\mathbf{u}_0) + (\mathbf{u} - \mathbf{u}_0)^\top f^{(1)}_{\mathbf{u}_0} + \sum_{p=2}^{r-1} \frac{1}{p!} (\mathbf{u} - \mathbf{u}_0)^\top f^{(p)}_{\mathbf{u}_0} (\mathbf{u} - \mathbf{u}_0)^{\otimes p-1}.
$$

Then, in a neighborhood about a query point $\mathbf{q}^*$, we approximate each of our model functions according to their Taylor series expansion. For the first component of the population model, we have

$$\begin{aligned}\alpha_1(\mathbf{u}^{(ijk)}) &\approx \alpha_1(\mathbf{q}^*) + (\mathbf{u}^{(ijk)} - \mathbf{q}^*)^\top (\alpha_1)^{(1)}_{\mathbf{q}^*} + \sum_{p=2}^{r-1} \frac{1}{p!} (\mathbf{u}^{(ijk)} - \mathbf{q}^*)^\top (\alpha_1)^{(p)}_{\mathbf{q}^*} (\mathbf{u}^{(ijk)} - \mathbf{q}^*)^{\otimes p-1} \\ &= \mathbf{T}_{ijk}^\top \boldsymbol{\beta}^{(1)},\end{aligned}$$

where $\mathbf{T}_{ijk} = \begin{bmatrix} 1 \\ (\mathbf{u}_1^{(ijk)} - \mathbf{q}_1^*) \\ \vdots \\ (\mathbf{u}_m^{(ijk)} - \mathbf{q}_m^*) \\ (\mathbf{u}_1^{(ijk)} - \mathbf{q}_1^*)(\mathbf{u}_2^{(ijk)} - \mathbf{q}_2^*) \\ (\mathbf{u}_1^{(ijk)} - \mathbf{q}_1^*)(\mathbf{u}_3^{(ijk)} - \mathbf{q}_3^*) \\ \vdots \end{bmatrix}$ and $\boldsymbol{\beta}^{(1)} = \begin{bmatrix} (\alpha_1)(\mathbf{q}^*) \\ ((\alpha_1)^{(1)}_{\mathbf{q}^*})_1 \\ \vdots \\ ((\alpha_1)^{(1)}_{\mathbf{q}^*})_m \\ ((\alpha_1)^{(2)}_{\mathbf{q}^*})_1 \\ ((\alpha_1)^{(2)}_{\mathbf{q}^*})_2 \\ \vdots \end{bmatrix}$. Note, here, that we made a notational switch where we are considering a particular point $\mathbf{u}^{(ijk)}$ (from our spatially and hierarchically indexed data) in the neighborhood of $\mathbf{q}^*$ and approximating the function at that point. It is still generic, but is now written this way to provide concrete matrices that we will build up into the entire design matrix. We do the same for each of the deviation surfaces resulting in

$$\begin{aligned}(\alpha_{\Delta_l})(\mathbf{u}^{(ijk)}) &\approx \alpha_{\Delta_l}(\mathbf{q}^*) + (\mathbf{u}^{(ijk)} - \mathbf{q}^*)^\top (\alpha_{\Delta_l})^{(1)}_{\mathbf{q}^*} \\ &\quad + \sum_{p=2}^{r-1} \frac{1}{p!} (\mathbf{u}^{(ijk)} - \mathbf{q}^*)^\top (\alpha_{\Delta_l})^{(p)}_{\mathbf{q}^*} (\mathbf{u}^{(ijk)} - \mathbf{q}^*)^{\otimes p-1} \\ &= \mathbf{T}_{ijk}^\top \boldsymbol{\beta}^{(l)}, \qquad \text{for } l \in \{2, \ldots, L\}.\end{aligned}$$

where analogously $\boldsymbol{\beta}^{(l)} = \begin{bmatrix} (\alpha_{\Delta_l})(\mathbf{q}^*) \\ ((\alpha_{\Delta_l})^{(1)}_{\mathbf{q}^*})_1 \\ \vdots \\ ((\alpha_{\Delta_l})^{(1)}_{\mathbf{q}^*})_m \\ ((\alpha_{\Delta_l})^{(2)}_{\mathbf{q}^*})_1 \\ ((\alpha_{\Delta_l})^{(2)}_{\mathbf{q}^*})_2 \\ \vdots \end{bmatrix}$. Combining these equations results in

$$\alpha(\mathbf{u}) \approx \widetilde{\mathbf{T}}^{\top}_{ijk}\boldsymbol{\beta},$$

where

$$\widetilde{\mathbf{T}}^{\top}_{ijk} = \begin{cases} [\mathbf{T}_{ijk} \mid \mathbf{0}_{(L-1)\times|\beta^l|}] & \mathbf{g}_{ij} = 1 \\ [\mathbf{T}_{ijk} \mid \mathbf{T}_{ijk} \mid \mathbf{0}_{(L-1)\times|\beta^{(l)}|}] & \mathbf{g}_{ij} = 2 \\ [\mathbf{T}_{ijk} \mid \mathbf{0}_{(l-1)\times|\beta^l|} \mid \mathbf{T}_{ijk} \mid \mathbf{0}_{(L-(l+1))\times|\beta^l|}] & \mathbf{g}_{ij} = l, l \in \{2, \ldots, L-1\} \\ [\mathbf{T}_{ijk} \mid \mathbf{0}_{(L-1)\times|\beta^{(l)}|} \mid \mathbf{T}_{ijk}] & \mathbf{g}_{ij} = L \end{cases}$$

and $\boldsymbol{\beta} = [\boldsymbol{\beta}^{(1)} \ldots \boldsymbol{\beta}^{(L)}]$. Note here that $[\cdot|\cdot]$ is just denoting matrix augmentation for visual distinction.

Then, we approximate the individual deviation surfaces and the rep deviation surfaces

in the same way. We have:

$$\begin{aligned}
\nu_i(\mathbf{u}^{(ijk)}) &\approx \nu_i(\mathbf{q}^*) + (\mathbf{u}^{(ijk)} - \mathbf{q}^*)^\top (\nu_i)^{(1)}_{\mathbf{q}^*} + \sum_{p=2}^{r-1} \frac{1}{p!} (\mathbf{u}^{(ijk)} - \mathbf{q}^*)^\top (\nu_i)^{(p)}_{\mathbf{q}^*} (\mathbf{u}^{(ijk)} - \mathbf{q}^*)^{\otimes p-1} \\
&= \mathbf{T}_{ij} \mathbf{b}_i^{(1)} \text{ and} \\
\zeta_{ij}(\mathbf{u}^{(ijk)}) &\approx \zeta_{ij}(\mathbf{q}^*) + (\mathbf{u}^{(ijk)} - \mathbf{q}^*)^\top (\zeta_{ij})^{(1)}_{\mathbf{q}^*} + \sum_{p=2}^{r-1} \frac{1}{p!} (\mathbf{u}^{(ijk)} - \mathbf{q}^*)^\top (\zeta_{ij})^{(p)}_{\mathbf{q}^*} (\mathbf{u}^{(ijk)} - \mathbf{q}^*)^{\otimes p-1} \\
&= \mathbf{T}_{ij} \mathbf{b}_{ij}^{(2)}.
\end{aligned}$$

This results in the overall approximation

$$\begin{aligned}
g(\boldsymbol{\mu}_{ijk}) = \log\left(\lambda_{ij}(\mathbf{q}^*)\right) &\approx \underbrace{\tilde{\mathbf{T}}_{ijk}^\top \boldsymbol{\beta}}_{\alpha(\mathbf{q}^*)} + \underbrace{\mathbf{T}_{ijk} \mathbf{b}_i^{(1)}}_{\nu_i(\mathbf{q}^*)} + \underbrace{\mathbf{T}_{ijk} \mathbf{b}_{ij}^{(2)}}_{\zeta_{ij}(\mathbf{q}^*)} \\
&= \tilde{\mathbf{T}}_{ijk}^\top \boldsymbol{\beta} + \mathbf{Z}_{ijk}^{(1)} \mathbf{b}_i^{(1)} + \mathbf{Z}_{ijk}^{(2)} \mathbf{b}_i^{(2)} \\
&= \tilde{\mathbf{T}}_{ijk}^\top \boldsymbol{\beta} + \mathbf{Z}_{ijk} \mathbf{b},
\end{aligned}$$

where $\mathbf{Z}_{ijk}^{(1)} = \mathbf{T}_{ijk}$, $\mathbf{Z}_{ijk}^{(2)} = \mathbf{T}_{ijk} \otimes (\mathbf{I}_I e_i^{(I)})$, $\mathbf{Z}_{ijk} = \begin{bmatrix} \mathbf{Z}_{ijk}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_{ijk}^{(2)} \end{bmatrix}$, $\mathbf{b}_i^{(2)} = [\mathbf{b}_{11}^{(2)} \dots \mathbf{b}_{nn_I}^{(2)}]$ and $\mathbf{b} = [\mathbf{b}_i^{(1)} \mathbf{b}_i^{(2)}] = [\mathbf{b}_1 \ \mathbf{b}_2 \ \dots \ \mathbf{b}_I \ \mathbf{b}_{11} \ \dots \ \mathbf{b}_{In_I}]$. Here, we assume $\mathbf{b}^{(1)} \sim N(\mathbf{0}, \mathbf{D}_{(1)})$ and $\mathbf{b}^{(2)} \sim N(\mathbf{0}, \mathbf{D}_{(2)})$ so that $\mathbf{b} \sim N(\mathbf{0}, \mathbf{D})$, where $\mathbf{D} = \begin{bmatrix} \mathbf{D}_{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{(2)} \end{bmatrix}$ (*i.e.* $\mathbf{b}^{(1)} \perp \mathbf{b}^{(2)}$). Note that Fan and Gijbels [94] recommend using only a first-order approximation (*i.e.*, $r = 1$), and in our estimations we often only use the 0th order (*i.e.*, $r = 0$) which is essentially the Nadaraya-Watson estimator in the mixed model setting. There is a general assessment that is often made in these sorts of models that bandwidth selection is more important than the order of the local polynomial for these models.

We can then stack the data across individuals and replicates to form the full model

matrices:

$$\boldsymbol{\eta} = g(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} \quad \Longleftrightarrow \quad \boldsymbol{\mu} = g^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b})$$

where $\mathbf{X} = [\tilde{\mathbf{T}}_{111}^{\top} \dots \tilde{\mathbf{T}}_{nn_iK}^{\top}]^{\top}$ and $\mathbf{Z} = [\mathbf{Z}_{11}^{\top} \dots \mathbf{Z}_{nn_i}^{\top}]^{\top}$ (or stacked in any other suitable ordering). To estimate a generalized linear mixed-model (GLMM) model, Breslow and Clayton [41] proposed maximizing the penalized quasi-loglikelihood with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, which has the form

$$pql(\boldsymbol{\beta}, \boldsymbol{\theta}) = -\frac{1}{2\phi}\left[\sum_{l=1}^{N} d(\mathbf{y}_l, \mu_l)\right] - \frac{1}{2}\mathbf{b}^{\top}\mathbf{D}^{-1}\mathbf{b} \tag{3.21}$$

where $\boldsymbol{\theta}$ includes all parameters in $\mathbf{b}$ and $\mathbf{D}$ and where the deviance measure-of-fit function is

$$d(y, \mu) = -2\int_{y}^{\mu} \frac{y-u}{wV(u)}du.$$

Cai and Wu [49] combined this with the local quasi-likelihood approach of Fan and Gijbels, to obtain the Penalized Local Polynomial Quasi-Likelihood (PLPQL),

$$plpql(\boldsymbol{\beta}, \boldsymbol{\theta}) = -\frac{1}{2\phi}\left[\sum_{l=1}^{N} K_h(||\mathbf{c}_l - \mathbf{q}^*||)d(\mathbf{y}_l, \boldsymbol{\mu}_l)\right] - \frac{1}{2}\mathbf{b}^{\top}\mathbf{D}^{-1}\mathbf{b}, \tag{3.22}$$

where $K_h(\cdot) = K(\cdot/h)/h$ is a kernel function with bandwidth h, $\mathbf{q}^*$ is the query point of interest, and for mixed-effect Poisson log-linear regression, $V(u) = u$, $\boldsymbol{\mu}_l = \exp(\mathbf{X}_l\boldsymbol{\beta} + \mathbf{Z}_l\mathbf{b})$, and $\mathbf{X}_l$ and $\mathbf{Z}_l$ are the lth rows of the model matrices, $\mathbf{X}$ and $\mathbf{Z}$, respectively.

Differentiating Equation 3.22 with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ provides the score equations

$$\sum_{l=1}^{N} K_h(||\mathbf{c}_l - \mathbf{q}^*||)\frac{\mathbf{X}_l(\mathbf{y}_l - \boldsymbol{\mu}_l)}{\phi\mathbf{w}_l V(\boldsymbol{\mu}_l)g'(\boldsymbol{\mu}_l)} = 0, \tag{3.23}$$

and

$$\sum_{l=1}^{N} K_h(||\mathbf{c}_l - \mathbf{q}^*||)\frac{\mathbf{X}_l(\mathbf{y}_l - \boldsymbol{\mu}_l)}{\phi \mathbf{w}_l V(\boldsymbol{\mu}_l) g'(\boldsymbol{\mu}_l)} = \mathbf{D}^{-1}\mathbf{b}. \tag{3.24}$$

Define the following working response:

$$\tilde{\mathbf{y}}_l = g(\tilde{\boldsymbol{\mu}}_l) + g'(\tilde{\boldsymbol{\mu}}_l)(\mathbf{y}_l - \tilde{\boldsymbol{\mu}}_l). \tag{3.25}$$

Then, solution to the PLPLQ equation can be obtained via Fisher scoring by repeatedly solving the following linear system:

$$\begin{bmatrix} \mathbf{X}^\top \tilde{\boldsymbol{\Omega}}_h \mathbf{X} & \mathbf{X}^\top \tilde{\boldsymbol{\Omega}}_h \mathbf{Z} \\ \mathbf{Z}^\top \tilde{\boldsymbol{\Omega}}_h \mathbf{X} & \mathbf{Z}^\top \tilde{\boldsymbol{\Omega}}_h \mathbf{Z} + \tilde{\mathbf{D}}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^\top \tilde{\boldsymbol{\Omega}}_h \tilde{\mathbf{y}} \\ \mathbf{Z}^\top \tilde{\boldsymbol{\Omega}}_h \tilde{\mathbf{y}} \end{bmatrix}, \tag{3.26}$$

where $\tilde{\boldsymbol{\Omega}}_h = \oplus_{l=1}^{N} \tilde{\boldsymbol{\Omega}}_{lh}$ with $\tilde{\boldsymbol{\Omega}}_{lh} = K_h(\mathbf{c}_l - \mathbf{q}^*)\tilde{\mathbf{r}}_l$ and $\tilde{\mathbf{r}}_l = \phi \mathbf{w}_l V(\tilde{\boldsymbol{\mu}}_l)[g'(\tilde{\boldsymbol{\mu}}_l)]^2$. Equivalently, we repeatedly fit the model

$$\mathbf{K}_h^{1/2}\tilde{\mathbf{y}} = \mathbf{K}_h^{1/2}\mathbf{X}\boldsymbol{\beta} + \mathbf{K}_h^{1/2}\mathbf{Z}\mathbf{b},$$

where $\mathbf{K}_h = \oplus_{l=1}^{N} K_h(\mathbf{c}_l - \mathbf{q}^*)$, and then update $\tilde{\mathbf{y}}$ until convergence.

This approach is summarized in Algorithm 3.5. In the algorithm, we first initialize the set of parameters $\boldsymbol{\beta}$ and $\mathbf{b}$ and then iteratively update the parameters until convergence. At each step, we compute the working response $\tilde{\mathbf{y}}$, the working means $\tilde{\boldsymbol{\mu}}$, and the weighted working covariance $\tilde{\boldsymbol{\Omega}}_h$. The algorithm then solves the mixed model equations (MME) to obtain new working estimates $\tilde{\boldsymbol{\beta}}$ and $\tilde{\mathbf{b}}$ to recompute $\tilde{\boldsymbol{\mu}}$ with. The algorithm continues until the deviance between the current and previous estimates is less than some threshold, τ. This procedure can be simplified further by using a box kernel, filtering the data to be within a neighborhood of $\mathbf{q}^*$, and then fitting a standard GLMM with polynomial basis to the filtered data. This is computationally more efficient as it greatly reduces the sample size and hence

Algorithm 3.5 Local IRWLS/Fisher Scoring for GLMM (Cai and Wu (2002))

Input:

$\mathcal{M}$ Model
$\mathbf{y}$ Response
$\mathbf{c}$ Centroids
$\mathbf{q}^*$ Query Points

Output:

$\hat{\boldsymbol{\beta}}$ Estimated Fixed-effect parameters
$\hat{\mathbf{b}}$ Estimated Random-effect parameters
$\hat{\boldsymbol{\sigma}}$ Estimated Variance components
ℓ Estimated log-likelihood

$[\mathbf{X}, \mathbf{Z}] \leftarrow \text{constructModelMatrices}(\mathcal{M}, \mathbf{c}, \mathbf{q}^*)$ ▷ Construct model matrices
$\mathbf{K}_h \leftarrow K_h(\mathbf{s} - \mathbf{q}^*) I_N$ ▷ Compute the kernel weights
$[\tilde{\boldsymbol{\beta}}, \tilde{\mathbf{b}}] \leftarrow \text{initParameters}(\mathbf{y}, \mathbf{X}, \mathbf{Z})$ ▷ Initialize estimates
$\tilde{\boldsymbol{\mu}} \leftarrow g^{-1}(\mathbf{X}\tilde{\boldsymbol{\beta}} + \mathbf{Z}\tilde{\mathbf{b}})$ ▷ compute the working mean
$\mathbf{D}_{new}(\mathbf{y}; \tilde{\boldsymbol{\mu}}) = \sum_i (d_i)^2$ ▷ compute the deviance current model
$\mathbf{D}_{old}(\mathbf{y}; \tilde{\boldsymbol{\mu}}) = maxDouble$
while $((\mathbf{D}_{old} - \mathbf{D}_{new})/\mathbf{D}_{old} > \tau$ **do** ▷ IRLS steps
  $\mathbf{D}_{old}(\mathbf{y}; \tilde{\boldsymbol{\mu}}) = \mathbf{D}_{new}(\mathbf{y}; \tilde{\boldsymbol{\mu}})$ ▷ update deviance
  $\tilde{\boldsymbol{\mu}} \leftarrow g^{-1}(\mathbf{X}\tilde{\boldsymbol{\beta}} + \mathbf{Z}\tilde{\mathbf{b}})$ ▷ compute the working mean
  $\tilde{\mathbf{y}} \leftarrow \mathbf{K}_h[\boldsymbol{\eta} + g'(\tilde{\boldsymbol{\mu}})(\mathbf{y} - \tilde{\boldsymbol{\mu}})]$ ▷ compute the weighted working response
  $\tilde{\boldsymbol{\Omega}}_h \leftarrow [\oplus_{l=1}^{N} K_h(\mathbf{c}_l - \mathbf{q}^*)\phi w_l V(\tilde{\boldsymbol{\mu}}_l)] I_N$ ▷ compute weighted covariance matrix
  $\begin{bmatrix}\tilde{\boldsymbol{\beta}} \\ \tilde{\mathbf{b}}\end{bmatrix} \leftarrow \text{solve}\left(\begin{bmatrix}\mathbf{X}^\top\tilde{\boldsymbol{\Omega}}_h\mathbf{X} & \mathbf{X}^\top\tilde{\boldsymbol{\Omega}}_h\mathbf{Z} \\ \mathbf{Z}^\top\tilde{\boldsymbol{\Omega}}_h\mathbf{X} & \mathbf{Z}^\top\tilde{\boldsymbol{\Omega}}_h\mathbf{Z}\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta} \\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}\mathbf{X}^\top\mathbf{K}_h\tilde{\mathbf{y}} \\ \mathbf{Z}^\top\mathbf{K}_h\tilde{\mathbf{y}}\end{bmatrix}\right)$ ▷ Solve Mixed Model Eqns.
  $\mathbf{D}_{new}(\mathbf{y}; \hat{\boldsymbol{\mu}}) = \sum_i (d_i)^2$ ▷ compute the deviance current model
$[\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\Omega}}] \leftarrow [\tilde{\boldsymbol{\beta}}, \tilde{\mathbf{b}}, \tilde{\boldsymbol{\Omega}}]$ ▷ Set final parameter estimates
$\ell \leftarrow logLik(\mathbf{X}, \mathbf{Z}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\Omega}})$ ▷ Compute log-likelihood
$[\hat{\sigma}, \hat{\sigma}_{\mathbf{b}^{(1)}}, \hat{\sigma}_{\mathbf{b}^{(2)}}] \leftarrow varianceComponents(\mathbf{Z}, \hat{\boldsymbol{\Omega}})$ ▷ Extract variance components
return $[\hat{\boldsymbol{\beta}}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\sigma}_{\mathbf{b}^{(1)}}, \hat{\sigma}_{\mathbf{b}^{(2)}}, \ell]$

---

the size of the model matrices used for all of the matrix computations in each iteration of the algorithm. It also simplifies implementation considerably, as standard GLMM software can be used to fit the local model. For these reasons, this is the approach we take in our experiments and analysis.

This approach gives a pointwise model to apply to a collection of query points $\mathbf{q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_m\}$. For each query point $\mathbf{q}_i$, we can estimate the local models, $\lambda(\mathbf{q}_i)$, and store

parameters $\boldsymbol{\beta}_i$, $\mathbf{b}_i$, along with relevant standard errors and likelihoods for full and reduced models. With these likelihoods, we can then use pointwise likelihoods to carry out pointwise likelihood ratio tests and estimated standard errors to compute pointwise intraclass correlation coefficients. This results in unadjusted p-value surfaces and ICC surfaces. This is summarized in Algorithm 3.6.

At the end of the estimation algorithm, we obtain a set of estimated parameters that we can use to reconstruct the estimated surfaces of $\hat{\alpha}(\mathbf{q})$, $\hat{\nu}(\mathbf{q})$, and $\hat{\zeta}(\mathbf{q})$, whose exponentiated sum gives the scaled estimated intensity surface, $\hat{\lambda}(\mathbf{q})/a$. Additionally, we have pointwise estimates of the likelihoods, which we can compare to an estimated likelihood for a reduced model to carry out likelihood ratio tests, resulting in p-value curves. We also have the pointwise intraclass correlation coefficients, $\widehat{ICC}_{\text{subject}} = \frac{\hat{\sigma}^2_{\text{subject}}}{\hat{\sigma}^2_{\text{subject}}+\hat{\sigma}^2_{\text{rep(in subject)}}+\hat{\sigma}^2_{\text{error}}}$ and $\widehat{ICC}_{\text{rep(in subject)}} = \frac{\hat{\sigma}^2_{\text{subject}}+\hat{\sigma}^2_{\text{rep(in subject)}}}{\hat{\sigma}^2_{\text{subject}}+\hat{\sigma}^2_{\text{rep(in subject)}}+\hat{\sigma}^2_{\text{error}}}$ where $\hat{\sigma}^2_{\text{subject}}$, $\hat{\sigma}^2_{\text{rep(in subject)}}$, and $\hat{\sigma}^2_{\text{error}}$ are the estimated variances of the subject, replicate within subject and random error deviation respectively. Finally, to carry out the global hypothesis test, we use a Westfall-Young permutation procedure to adjust the pointwise p-values and then take the minimum over the adjusted p-value surface as our global p-value.

Note that there is clear reasoning for using of the minimum adjusted p-value as the global p-value. As stated by Xu and Reiss, "Clearly, the global hypothesis ... is just the intersection of all of the pointwise hypotheses" [309]. That is, if there is any point in the domain where we can reject the null hypothesis, then we must reject the global null hypothesis that there are no differences anywhere in the domain. More explicitly, the null-hypothesis for the global test in the functional setting corresponds to

$$H_0 : f(s) = 0 \ \ \forall s \in \mathcal{S},$$

for a function f defined over a domain $\mathcal{S}$. This is equivalent to a family of pointwise null

hypotheses,

$$\{H_0(s) : s \in \mathcal{S}\}.$$

Thus, if the collection of adjusted p-values is dense enough over the domain, then rejecting the global null hypothesis when the minimum adjusted p-value is less than the significance level α approximates and is asymptotically equivalent to rejecting the global null hypothesis at level α. This is essentially the same reasoning used by Olsen *et al.* [220] to prove that the Benjamini-Hochberg procedure controlls FDR in the functional setting. In our view, one only needs an appropriate, trustworthy, multiple testing adjustment procedure that controls the FWER over the collection of pointwise tests being carried out. Cox and Lee showed that Holm's [137] procedure is not an adequate approach for controlling the FWER in this setting, as the Type I error rates converge to one as the number of pointwise tests increases to infinity. In contrast, they showed that the Westfall-Young permutation-based approach controls the FWER in this setting. Our experience with this approach shows that it can maintain sufficient power to be useful in higher dimensional testing situations.

We end this chapter with an illustrative example of fitting the model to a simulated dataset. We simulate one-dimensional hierarchical mixed curve data according to the model:

$$m_{ij}(t_{ijk}|G_{ij} = l) = t_{ijk}^{l}(1 - t_{ijk})^{6-l} + \mathbf{b}_i(t_{ijk}) + \mathbf{b}_{ij}(t_{ijk}) + \epsilon_{ijk}$$

where $\mathbf{b}_i^{(1)}(t)$ follows a Brownian motion model $\mathbf{b}_i^{(1)}(t) \sim BM(0, 1/0.008)$, $\mathbf{b}_{ij}^{(2)} \sim GP(0, c(h))$, $\epsilon_{ijk} \overset{\text{iid}}{\sim} \text{N}(0, 0.008)$, $t_{ijk} \overset{\text{iid}}{\sim} \text{Uniform}(0, 1)$ and $l \in \{1, \ldots, 3\}$. Here, we use $c(h) = \exp(-|h|/0.0008)$ (*i.e.* a Gaussian covariance function with scale, 1, and variance 0.0008). We simulate $I = 10$ individuals with $n_i = 10$ replicates each and observe data at 30 uniformly distributed time points in $[0, 1]$. A plot of the simulated data is provided in Figure 3.5. In this plot, each curve corresponds to a replicate curve for an individual colored by the group level of the categorical variable. The true underlying population curves are shown in bold.

Algorithm 3.6 Estimation Pipeline

Input:

$\{\mathcal{D}_{ij}\}$ Persistence Diagrams
W Window
dx Cell size
$\{Q_i\}_{i=1}^m$ Queries
$\mathcal{M}_1$ Reduced Model
$\mathcal{M}_0$ Full Model

Output:

$\hat{\boldsymbol{\lambda}}$ Estimated PIFs
$\hat{\boldsymbol{\sigma}}^2$ Variance Components
p P-value Curves
$\widehat{\mathbf{ICC}}$ ICC Curves

$[\{\mathbf{y}_{ijk}\}, \{\mathbf{s}_{ijk}\}] \leftarrow \text{bin}(\mathcal{D}_{ij}, W, dx)$ ▷ Bin the diagrams
$[\mathbf{X}, \mathbf{Z}] \leftarrow \text{constructModelMatrices}(\mathbf{s}, \mathbf{s})$ ▷ Construct model matrices
for $Q_u = Q_1, \ldots, Q_m$ **do** ▷ Estimate for each query point
  $[\hat{\boldsymbol{\beta}}_1, \hat{\mathbf{b}}_1, \hat{\sigma}_\epsilon, \hat{\sigma}_{\text{subject}}, \hat{\sigma}_{\text{rep(in subject)}}, \ell_1]_u \leftarrow IRWLS(\mathcal{M}_1, \{\mathbf{y}_{ijk}\}, \{\mathbf{s}_{ijk}\}, Q_u)$ ▷ Fit full
  $[\hat{\boldsymbol{\beta}}_0, \hat{\mathbf{b}}_0, \hat{\sigma}_\epsilon, \hat{\sigma}_{\text{subject}}, \hat{\sigma}_{\text{rep(in subject)}}, \ell_0]_u \leftarrow IRWLS(\mathcal{M}_0, \{\mathbf{y}_{ijk}\}, \{\mathbf{s}_{ijk}\}, Q_u)$ ▷ Fit reduced
  $\mathbf{p}_i \leftarrow pchisq(2 \cdot (\ell_1 - \ell_0), \text{df}_1 - \text{df}_2)$ ▷ Compute p-value
  $\widehat{\mathbf{ICC}}_{u,\text{subject}} \leftarrow \frac{\hat{\sigma}^2_{\text{subject}}}{\hat{\sigma}^2_{\text{subject}} + \hat{\sigma}^2_{\text{rep(in subject)}} + \hat{\sigma}^2_{\text{error}}}$
  $\widehat{\mathbf{ICC}}_{u,\text{rep(in subject)}} \leftarrow \frac{\hat{\sigma}^2_{\text{subject}} + \hat{\sigma}^2_{\text{rep(in subject)}}}{\hat{\sigma}^2_{\text{subject}} + \hat{\sigma}^2_{\text{rep(in subject)}} + \hat{\sigma}^2_{\text{error}}}$ ▷ Compute ICCs
  $\hat{\boldsymbol{\lambda}}_u \leftarrow (1/dx)\exp(\mathbf{X}\hat{\boldsymbol{\beta}}_1 + \mathbf{Z}\hat{\mathbf{b}}_1)$ ▷ Reconstruct estimated intensity surfaces at Q_u
return $[\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\sigma}}^2_\epsilon, \hat{\boldsymbol{\sigma}}^2_{\text{subj}}, \hat{\boldsymbol{\sigma}}^2_{\text{ind}}, \mathbf{p}, \widehat{\mathbf{ICC}}_{\text{subj}}, \widehat{\mathbf{ICC}}_{\text{ind}}]$

We fit the model according to the approach described above using a Nadaraya-Watson estimator (*i.e.*, $r = 0$) with a box kernel with bandwidth of $h = 0.08$. The estimated population-level intensity surfaces are provided in Figure 3.6 along with raw and adjusted p-value curves. From the pointwise estimated models on a fine grid of points in t, we obtain pointwise estimates of fixed-effects, random-effects, variances, as well as likelihood ratio test statistics. Using those, we construct the estimated population curves, individual curves, standard deviation curves, and pointwise p-values. Note, we use a Kenward-Roger approximation [158] to compute the pointwise p-values for the fixed-effect curves, as it is

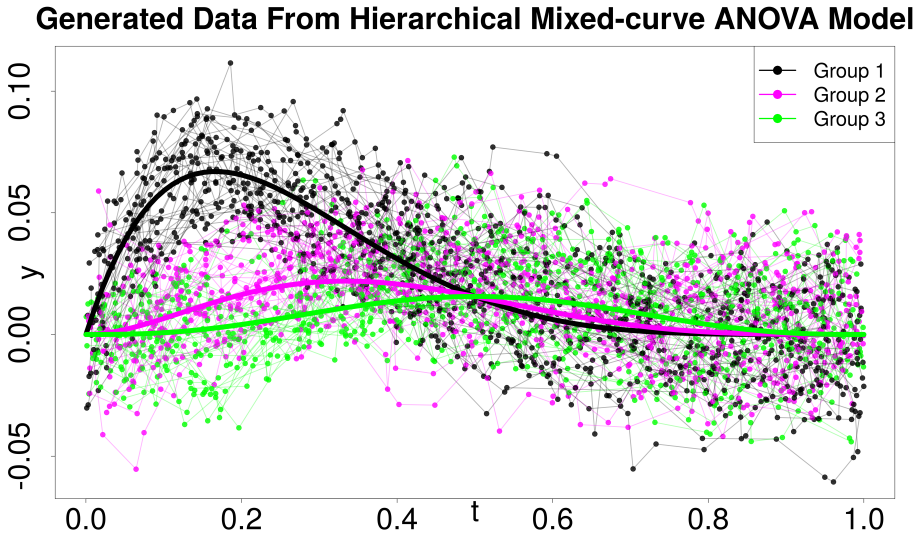


Figure 3.5: Simulated data from the example mixed-curve model. We have 10 individuals with 10 replicates, resulting in 100 total curves. Which categorical level each curve is associated to (*i.e.* the group label) is determined by the group label G_{ij}, which is sampled uniformly across the 3 groups for each replicate curve. Each curve is observed at 30 uniformly distributed time points in $[0, 1]$. The true underlying population curves are shown in bold and raw curves for each replicate curve colored by group label G_{ij}.

well-known that likelihood ratio tests can be anti-conservative in mixed-effect models.

While there is some evidence that there may be some benefit of using Kenward-Roger approximations in the GLMM setting [283], there are no anyaltical results that we are aware of that show that this is appropriate. Thus, we use the Kenward-Roger approximation for the fixed-effect tests in the LMM setting, but we just use likelihood ratio tests for the GLMM setting. For the random-effect tests, we use a 50:50 mixture of chi-squared distributions, which was shown by Self and Liang [266] to be a more appropriate null distribution for testing variance components. Note that parametric bootstrap approaches are also commonly used, but we use the simpler adjustments here for computational efficiency.

Repeating the procedure over a large number of subset pivotal permutations (we used 200 here) of the group labels allows us to compute adjusted p-values for the fixed-effect curves

using the Westfall-Young procedure. For this case, we permute the group label corresponding to each replicate curve across all individuals and replicates while preserving the individual and replicate structure of the data (*i.e.*, we permute labels within individuals). Then, for the individual random-effect tests the subset pivotal permutations involve permuting which individual each replicate curve belongs to across all individuals and replicates. Finally, for the replicate random-effect, a subset pivotal permutation involves permuting the replicate labels within each individual across all *samples* within an individual, while preserving the replicate group labels and the spatial index. That is, we shuffle which samples are associated to which replicate within each individual and time point, but only across replicates of the same label. These subset pivotal permutations ensure that the distribution of the test statistic under the null hypothesis is preserved across permutations.

For illustrative purposes, we also provide adjusted p-value curves according to the Holm, Hochberg, Hommel, Benjamini-Hochberg, Benjamini-Yekutieli, and Romano-Wolf procedures. Note however, that we can only recommend the Westfall-Young procedure for controlling the FWER in this setting based on the analytical results of Cox and Lee [62]. In this example (see Figure 3.6 (c), (e), and (f)), it does appear though that all of the adjusted p-value curves are fairly similar. However, to the best of our knowledge, the Westfall-Young procedure is the only one that has been shown to have theoretical guarantees for controlling the FWER in the local-test setting. We do suspect that the Romano-Wolf procedure may also control the FWER in this setting, as it is also a test based on resampling and step-down procedures similar to Westfall-Young. There are analytical results in the literature that show that the Benjamini-Hochberg procedure over a dense sequence across the domain approximates a FDR in the functional data setting, thus implying that local testing should be appropriate with Benjamini-Hochberg adjustments (see [220]). However, for the global test (*i.e.*, by taking the minimum of the adjusted p-value curves), we can only recommend the FWER controlling procedures as we would expect FDR methods to have inflated Type I error rates for the

global test.

Comparing Figures 3.6 (a) to 3.5, we see that the mixed-curve model is able to recover the underlying population curves quite well and that the Westfall-Young adjusted p-value surface is able to identify the regions of the domain accurately where there are clear differences between the population mean curves (Figure 3.6 (c)) and where there are not (*i.e.*, where the curves cross and the edges of the domain), and where there are individual and replicate-level variations (Figures 3.6 (e) and (f)). The estimated variance curves (Figure 3.6 (b)) show the increasing variability of the individual effects which stems from the Brownian motion model used to simulate the individual effects. In contrast, the random error and replicate variation are relatively constant across the domain. This has the effect of increasing intraclass correlation coefficient curves (Figure 3.6 (d)) as a function of t since the individual variation is in the numerator of the calculation for both ICC curves.

3.5 Discussion

In this work, we made use of local hierarchical mixed-effect modelling to model collections of persistence diagrams in a hierarchical design setting, treating persistence diagrams as Poisson point processes. This approach allows us to model and conduct inference for persistence diagrams under more complicated sampling and experimental designs than previously considered in the literature. Notably, the methods could be applied beyond the persistence diagram data and are more broadly applicable to repeated measures designs for functional data and point pattern data. Additionally, we have constructed a hypothesis testing procedure based on pointwise likelihood ratio tests and multiple comparison adjustment procedures to globally test for differences in the population-level intensity function across different levels of a categorical covariate after accounting for subject-level and replicate-level variability. Because this is based off of local-testing, this also allows us to detect regions of the domain where there are differences in the persistence diagrams between different levels of

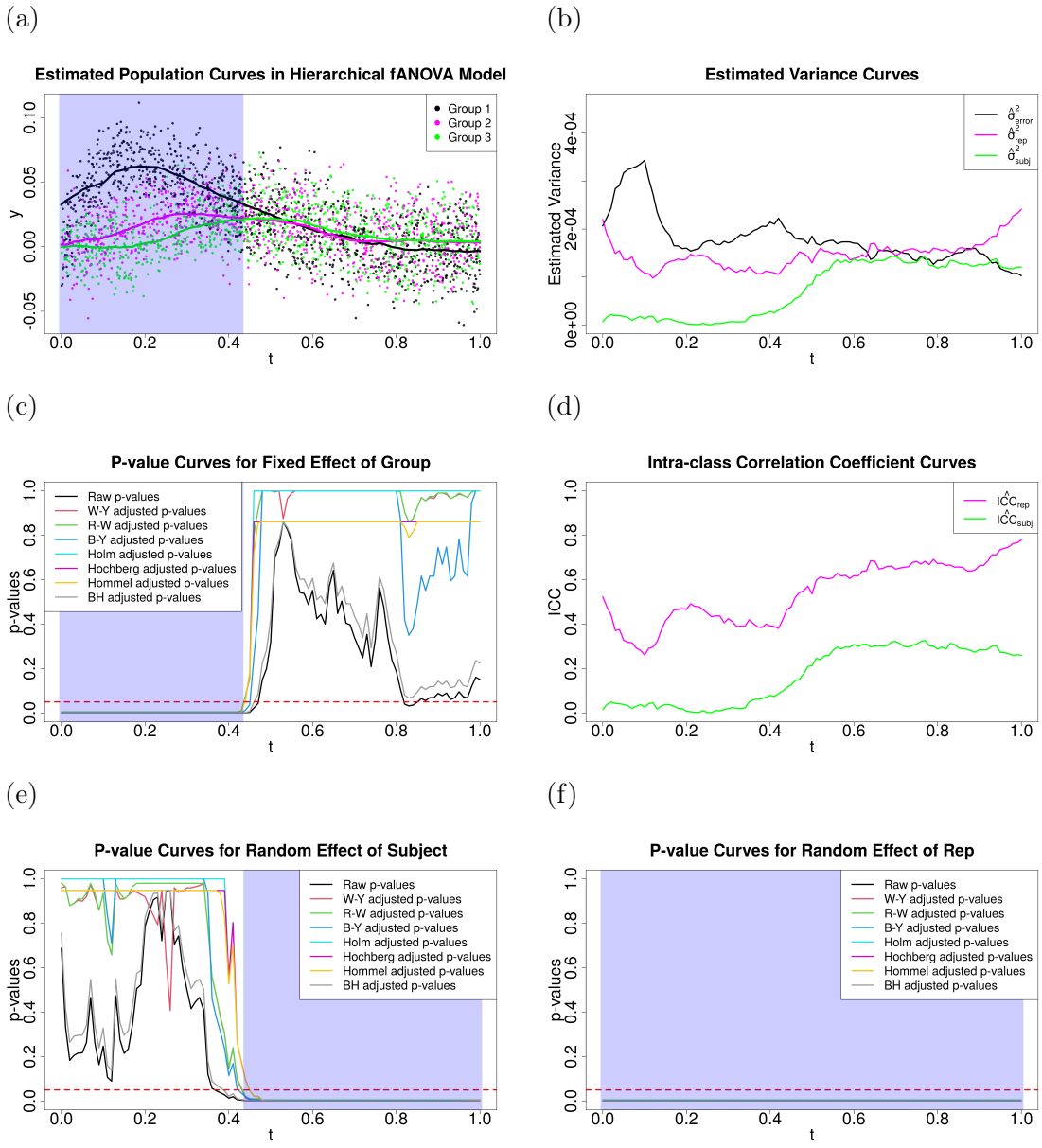


Figure 3.6: Estimated population intensity curves, variance curves, ICC curves and p-value curves for a simulated data set with 10 subjects, 10 curves per subject, observed with noise over 30 uniformly sampled points along the curve, from three groups of true mean curves. (a) Estimated population curves for each group and estimated subject curves. (b) Estimated variance curves for errors, rep and subject random effects. (c) p-value curves for the fixed-effect test. (d) ICC curves for the subject and replicate random effects. (e) p-value curves for the subject random-effect test. (f) p-value curves for the replicate random-effect test.

the covariate.

3.6 Future Work

There are several avenues for future work. The first involves the use of other dependence structures in the random effects. For example, we may wish to consider spatial or temporal dependence structures in the random effects to model persistence diagrams changing across space or time. In the Gleason grade application, this would allow us to model disease progression over time or spatially over a tissue sample. In general, this would allow us to model data collected over time or space and be able to account for the lack of independence between observations due to these more complex dependence structures.

The methods could incorporate more than one predictor or continuous covariates in the fixed-effects portion of the model, which would allow us to model the population-level intensity function as a function of continuous covariates such as age or biomarker levels, and explore group differences controlled for demographics of subjects. This would allow us to better understand the relationship between the persistence diagrams and these types of covariates.

In this work, we have only considered the case of a single global hypothesis test for differences in the population-level intensity function across two levels of a categorical covariate, though our method handles more than two levels of a categorical covariate as well. However, because the fANOVA test considers multiple groups, it is important to be able to carry out follow-up pairwise comparisons between multiple groups to better characterize which groups differ from each other (and where in t this occurs). As stated here, our method does not directly allow for pairwise comparisons between multiple groups, though it could be adapted to do so by carrying out pairwise comparisons between groups and applying a multiple comparison adjustment procedure to control for the FWER. Future work could involve developing these methods for carrying out pairwise comparisons between multiple

groups while controlling for the FWER over t.

Another avenue for future work involves the use of Bayesian methods for estimating the population-level intensity function and conducting inference. This would allow us to incorporate prior information about the intensity function to obtain more accurate estimates of the intensity function in settings with limited data and obtain model estimates when the pointwise MLEs are not well-defined.

Finally, it would be worth investigating more complex point pattern models for modelling the persistence diagrams. For example, we could consider Cox processes to model more complex sources of variability or Gibbs processes to model dependence between points within a persistence diagram. Gibbs processes were considered by Adler *et al.* [5] in the context of modelling persistence diagrams, but not in the context of mixed-effect models or hypothesis testing. This would allow us, for example, to model interactions between points in the persistence diagrams and better understand the dependencies between topological features within a persistence diagram. Additionally, we could consider marked point processes to model the collection of persistence diagrams (*i.e.,* multiple homological dimensions) in a single model. In our modelling approach, we have treated each homological dimension independently, but there is most likely a strong dependence between homological dimensions that may be possible to model using marked point processes.

CHAPTER FOUR

MIXED CURVE MODEL EXPERIMENTS AND GLEASON ANALYSIS

Now that we have introduced the mixed-curve modelling framework for estimating and testing for differences in functional data, we carry out several experiments to evaluate the Type I error rates and power of this method in various settings. We conduct several simulation studies to evaluate the performance of the method in functional ANOVA settings with different error models and noise levels, as well as in a point process intensity function setting and a repeated measures functional ANOVA setting. Finally, we apply the method to a dataset of 2D histopathology images of prostate cancer tissue graded according to the Gleason grading system to identify differences in the topological features of the tissue between the two grades.

4.1 Simulation Experiments

We carry out several experiments on simulated data inspired by a power analysis for the functional ANOVA test originally carried out by Cuevas *et al.* [66] and expanded on by Mrkvicka *et al.* [209]. Their experiments are conducted on a collection of 1D functions, which we add an additional benchmark (M5) to, based on a scaled beta distribution, where the local effect-size is greatest in the center of the domain where the edge-effect bias of the mixed-curve estimator is smallest. The collection of benchmark functions for the means of each of the 3 groups are as follows:

- $M1 : m_{(l)}(t) = t(1-t)$ for $l = 1, 2, 3, \;\; t \in [0, 1]$,
- $M2 : m_{(l)}(t) = t^l(1-t)^{6-l}$ for $l = 1, 2, 3, \;\; t \in [0, 1]$,
- $M3 : m_{(l)}(t) = t^{l/5}(1-t)^{6-l/5}$ for $l = 1, 2, 3, \;\; t \in [0, 1]$,

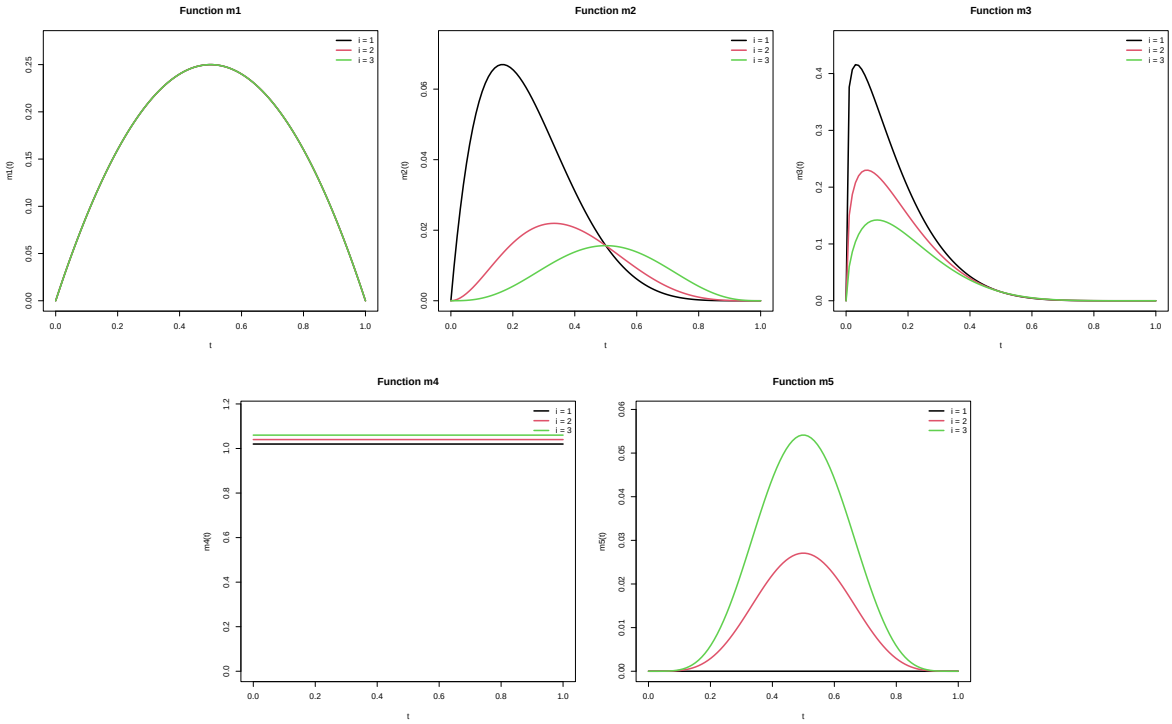


Figure 4.1: 1D functions used in in our power analysis.

- $M4 : m_{(l)}(t) = 1 + l/50$ for $l = 1, 2, 3, \;\; t \in [0, 1]$, and
- $M5 : m_{(l)}(t) = (l-1)\frac{\Gamma(12)}{\Gamma(6)\Gamma(6)} t^{6-1}(1-t)^{6-1}$ for $l = 1, 2, 3, \;\; t \in [0, 1]$.

We provide illustrations of each function in Figure 4.1.

In the original Cuevas *et al.* [66] experiments, they considered experimental factors of noise level and error model (white noise and Brownian motion). Levels of noise considered were $\sigma^{BM} = \{0.2, 1.0, 1.8, 2.6, 3.4, 4.2, 5.0\}$ and $\sigma^{WN} = \sigma^{BM}/25$ for the Brownian motion and white noise cases respectively. We replicate these two experiments using an implementation in the `fdANOVA` R package [209] and expand noise levels to also include $\sigma \in \{10, 15, 20, 25, 50, 100\}$. For each replicate curve in the 1D fANOVA experiments, we sample 25 points $(t_1, \ldots, t_{25})$ over a fixed grid in the domain $[0, 1]$ and sample $n = 10$ replicates (curves) for each function $m_i(t)$. We compare the homogeneous and inhomogeneous versions of the Cuevas *et al.* overall

F-test to our method of pointwise hypothesis testing with a Westfall-Young adjustment and taking the minimum of the adjusted p-values. We also include other FWER adjustment methods including Romano-Wolf, Hochberg, Hommel, Holm, and also the FDR adjustment methods Benjamini-Hochberg and Benjamini-Yekutieli. With the functional ANOVA design, we fit a reduced version of the model introduced in Equation 3.4:

$$m_{ij(l)}(t_{ijk}|G_{ij} = l) = \alpha_A(t_{ijk}) + \alpha_{\Delta_B}(t_{ijk})I(G_{ij}^{grp} = 2) + \alpha_{\Delta_C}(t_{ijk})I(G_{ij}^{grp} = 3) + \nu_i(t_{ijk}) + \epsilon_{ijk}.$$

Notice the removal of the replicate-level random-effect term, $\zeta_{ij}(\cdot)$. For an additional baseline comparison, we also include a likelihood ratio test using B-spline parameterizations of the curves of order 2 with five equally spaced knot points across the interval $[0, 1]$. Note that, for each experiment, we replicate the experiment 500 times and average the results across replicates to estimate power and Type I error rates as a function of noise level and error model.

Because we are interested in using the mixed-curve model for 1D and 2D point process intensity functions, we consider other simulations not present in Cuevas *et al.* [66] that are more directly relevant to this setting. In particular, we expand upon their experiments to also include a 1D point process intensity function setting, a 2D functional ANOVA setting, and a 1D hierarchical mixed-curve (repeated measures functional ANOVA) setting.

For the point process intensity function setting, we consider a scaled and shifted version of the 1D intensity function defined. We first consider the max and min values of the three functions ($m_i(t, 1), m_i(t, 2)$, and $m_i(t, 3)$), denoted as $d = \max_{l,t} m_i(t, l)$ and $c = \min_{l,t} m_i(t, l)$. Then, we scale all of them together by that range, $(d - c)$, and transform the new min and max to be between 0.3 and 0.6 respectively. Finally, the transformed functions are scaled by 10000 and then 2000, and Gaussian process noise with variance $\sigma^{BM} * 1M$ and scale 1 are added. This ensures that the intensity functions are positive and have a reasonably large

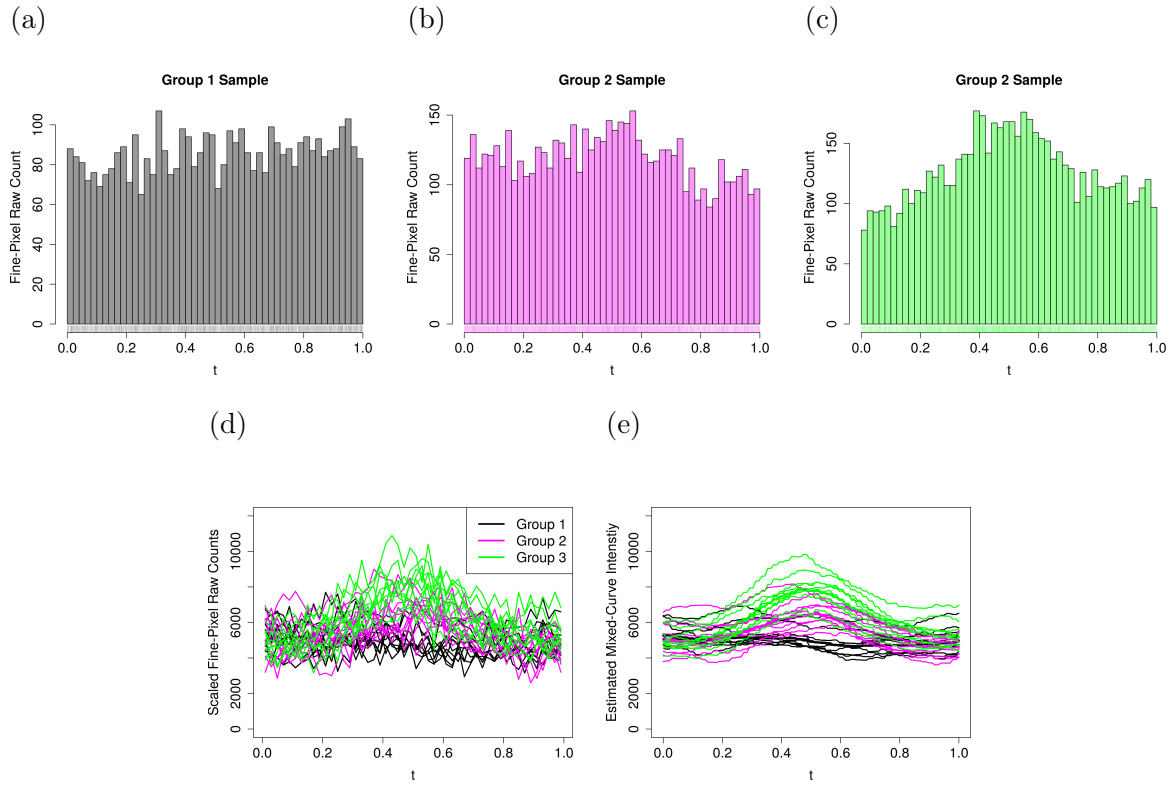


Figure 4.2: Example dataset of point patterns sampled from the inhomogeneous Poisson process. Figure (a)-(c) give fine-pixel counts via a histogram with 50 cells for a single replicate of the point pattern for each of the 3 groups with a rug plot below giving the raw data points. Figure (d) gives all fine-pixel counts (scaled by inverse of volume, *i.e.* 50) for all replicates and groups. Figure (e) gives the estimated intensity curves for each group using the mixed-curve model.

number of points sampled from the inhomogeneous Poisson process to detect differences with small sample sizes. We then sample a point pattern from the inhomogeneous Poisson process with intensity function, $\lambda_{(l)}(t)$, using the Lewis-Shelder thinning based approach (see Lewis and Shedler [175] and Sec. 5.4.2 of Baddeley *et al.* [16]). An example dataset of point patterns sampled from the inhomogeneous Poisson process is provided in Figure 4.2 along with estimated curves from the associated mixed-curve model.

For the 2D case, we construct a rotation of the origin of the 1D functions about the point $(0.5, 0.5)$. We do this by first composing the L_2 norm function between a particular

point, $\mathbf{s}$ and the new origin point $(0.5, 0.5)$ and scaling to fit the function across the diagonal from $(0.5, 0.5)$ to $(1.0, 1.0)$. That is, for $\mathbf{s} \in [0,1]^2$, we define the 2D functions as

$$m_{(l)}^{(2d)}(\mathbf{s}) = m_{(l)}\left(\sqrt{(\mathbf{s}_1 - 0.5)^2 + (\mathbf{s}_2 - 0.5)^2} \times \sqrt{2}/2\right),$$

for each of the 1d functions ($M1$ thru $M5$). Heatmaps for the 2D variants of the functions are provided in Figure 4.3. For the 2D cases we sample $K = 100$ points uniformly from the domain $[0,1]^2$.

With the repeated measures functional ANOVA design, we fit the hierarchical mixed-curve introduced in Equation 3.4,

$$m_{ij(l)}(t|G_{ij} = l) = \alpha_1(t_{ijk}) + \alpha_{\Delta_2}(t_{ijk})I(G_{ij} = 2) + \alpha_{\Delta_3}(t_{ijk})I(G_{ij} = 3) + \nu_i(t_{ijk}) + \zeta_{ij}(t_{ijk}) + \epsilon_{ijk}.$$

In this case, we simulate $I = 30$ individuals with $n_i = 30$ replicates each and observe data at $K = 30$ uniformly distributed points over the domain, t. We consider a white-noise error model here as well. We vary the noise level, σ, at each of the levels of the hierarchical model relative to a baseline model where $\nu_i(t_{ijk}) = b_i \sim N(0, 0.008)$, $\zeta_{ij}(t_{ijk}) = b_{ij} \sim N(0, 0.008)$, and $\epsilon_{ijk} \sim N(0, 0.008)$. We then vary noise levels in σ^{WN}.

4.2 Simulation Results

We now summarize the results of our simulation experiments, beginning with the functional ANOVA setting. It is notable that all of the methods considered here typically maintain Type I error rate control fairly well across all noise levels and are able to detect differences when there is a true effect for some range of noise levels.

Results for the white-noise error model case with $B = 500$ simulations are summarized in Figure 4.4. Surprisingly, we do not see the minimum FDR-adjusted p-values (Benjamini-

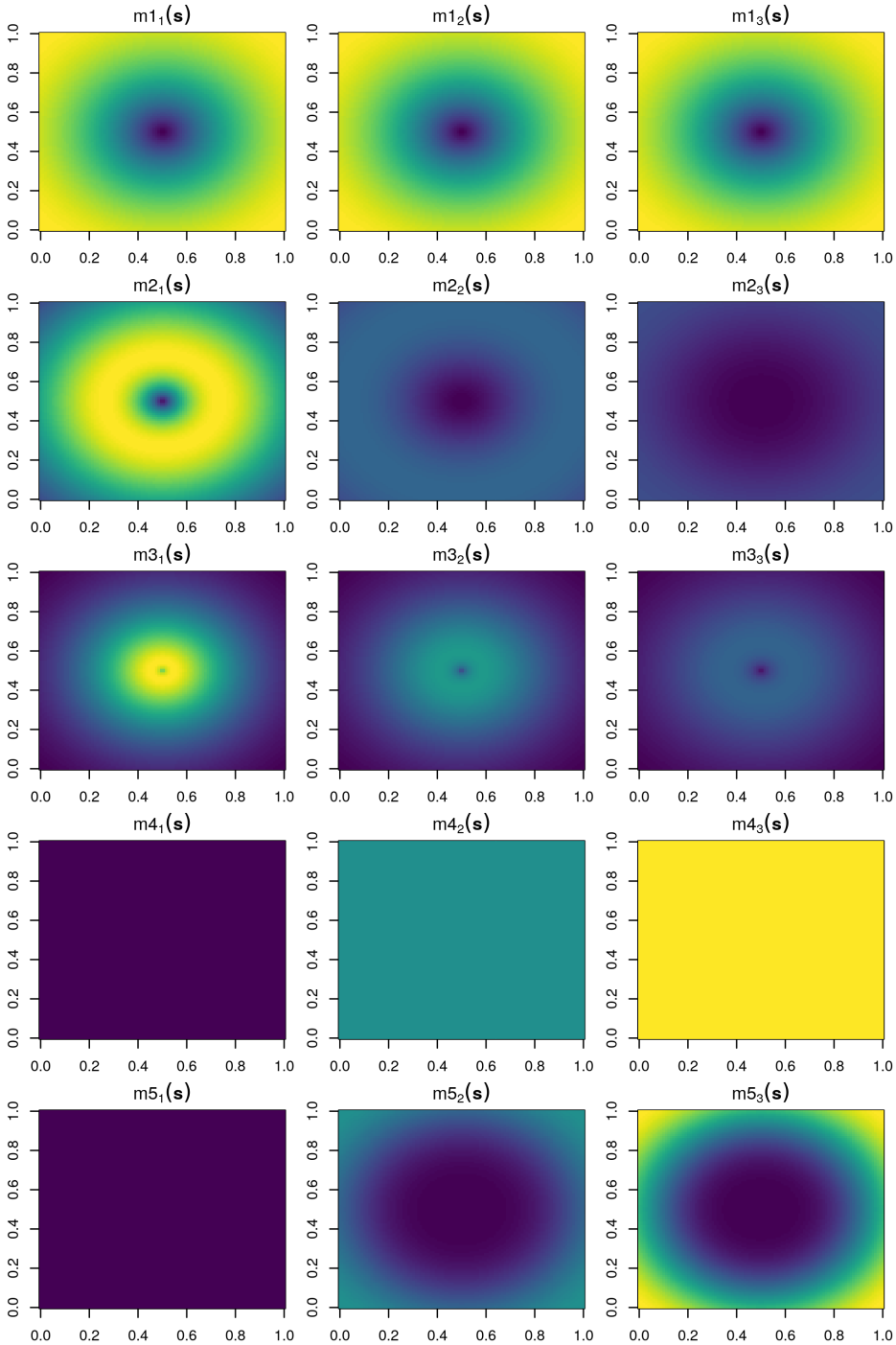


Figure 4.3: Heatmaps of Cuevas *et al.* functions modified to be used our 2d fANOVA power analysis study. Colors are scaled to be between min and max of function values across each row.

Hochberg and Benjamini-Yekutieli) performing poorly on M1, as we would expect them to have inflated Type I error rates in this global hypothesis testing setting. This may be due to the fact that the number of pointwise tests being carried out here (100) is not large enough for the FDR methods to have inflated error rates. In general, we see that most methods are conservative for the case where the null hypothesis is true (M1), except for the likelihood ratio test of B-splines and the Romano-Wolf adjustment method, both of which are only slightly conservative. For the other benchmarks (M2-M5) where there is a true effect, we see that likelihood-ratio, Romano-Wolf, and Benjamini-Hochberg methods consistently had the highest power across all noise levels and functions. The Westfall-Young, Hommel and Benjamini-Yekutieli methods group together in the middle with more moderate power, while the Cuevas *et al.* methods have the lowest power overall for assessing the global hypothesis across time.

For the Brownian motion error model case, results are summarized in Figure 4.5. Here, we see somewhat similar trends as in the white-noise case, except that the likelihood ratio test of B-splines now is anti-conservative along with the inhomogeneous Cuevas *et al.* method. Benjamini-Hochberg and Cuevas homogeneous methods were only slightly conservative (with estimated rejection rates at about 0.03) while the remaining were all even more conservative (with estimated rejection rates near 0.01). For the benchmarks with true effects (M2-M5), we again see that the likelihood-ratio test of B-splines was the most powerful and Benjamini-Hochberg and Romano-Wolf methods also having high power. The Cuevas methods usually had lower power for these benchmarks, and the remaining methods usually clustering together in the middle with more moderate power.

In the point pattern case, results are summarized in Figure 4.6. We see the Likelihood ratio method is anti-conservative, meaning the nominal Type I error rate is larger than the specified level (*i.e.*, 0.05), while all other methods are conservative for Type I error control across the noise levels. Romano-Wolf and Benjamini-Hochberg methods had the highest

power across the other benchmarks, while the remaining methods all grouped together with slightly lower power across the noise levels. Overall, we see that all methods except for the Likelihood ratio test of B-splines had good Type I error control and power across the noise levels for the point process case, which is encouraging for the use of these methods in this setting.

Results for the 2D functional ANOVA case are summarized in Figure 4.7. Here, we see that Romano-Wolf was anti-conservative for Type I error control which is then offset by maintaining higher power. The remaining methods all seem to mostly group together, with Westfall-Young maintaining the closest to nominal Type I error rate control and having the highest power among those methods. Benjamini-Yekutieli notably had very conservative Type I error rates and the lowest power overall.

Next, consider the case of the repeated measures functional ANOVA setting. Results for the case where we increase the lowest level of noise (*i.e.*, the white-noise error term) are summarized in Figure 4.8. Here, we see that all methods typically maintain good Type I error rate control but are typically conservative with the exception of one or two noise levels. All of the methods have similar power curves, except for Benjamini-Yekutieli which has notably lower power than the other methods, which also coincides with being more conservative for Type I error control. For the case where we increase the middle level of noise (*i.e.*, the individual-level random-effect term), results are summarized in Figure 4.9. Here, we see Type I error control is only maintained well for the permutation based methods (Westfall-Young and Romano-Wolf), while the other methods are anti-conservative at all noise levels. Power is maintained at 1 for all methods in the cases where there is a true effect. We believe that this is likely due to the fact that rep-to-rep variablity is being captured well by the model even at high levels of individual-level variability. Finally, for the case where we increase the highest level of noise (*i.e.*, subject-level random effect), results are summarized in Figure 4.10. Here, we see a similar pattern for the lowest level of noise case. The permutation based

methods maintain Type I error control well, while the other methods are anti-conservative. This lack of Type I error control also coincides with high power across the range of noise levels. In contrast, the Type I error rates and the rates for power for the permutation based methods (Westfall-Young and Romano-Wolf) appear to be reasonable.

Overall, we see that the Westfall-Young and Romano-Wolf methods generally perform well in these experimental settings with a few exceptions. One thing to keep in mind with these experiments is that the number of pointwise tests being carried out here is relatively small (100 for 1D and 400 for 2D) and the sample sizes are also small ($n = 10$ for each group in 1D and 2D and n_i,n_j,$n_k = 30$ for the hierarchical setting). We may expect that the results to improve for these methods with larger sample sizes and potentially get worse for some methods as the number of pointwise tests increases.

4.3 Gleason Data Analysis

In this section, we apply the local mixed-curve modelling approach to a dataset of 2D images of histopathology images of prostate cancer data graded according to the Gleason grading system [91] as described in Chapter 2. For this analysis, we have a dataset consisting of 200 512×512 pixel images of prostate cancer tissue, with each image graded according to the Gleason grading system and being of one of two grades: 3 or 4 ($n_{G3} = 147$, $n_{G4} = 53$). It is important to distinguish between Gleason grades 3 and 4 as they have very different prognostic implications for patients, with grade 4 being much more aggressive and associated with worse outcomes than grade 3. Additionally, it is often difficult for pathologists to distinguish between these two grades, resulting in high inter- and intra-observer variability in grading from pathologists.

Images are obtained from $i = 1, \ldots, 10$ individuals, with $j = 1, \ldots, 10$ replicate images per individual. Nuclei locations were extracted from these images using a U-net convolutional neural network [259] obtained from the HistomicsTK python library [238] so that the raw pixel

images are converted into 2D point clouds of nuclei locations. See Figure 4.11 for examples of nuclei extracted from images of different Gleason grades. We then apply a Vietoris-Rips filtration (Section 2.3.1) to each image and compute persistence for each filtration, resulting in both 0 and 1-dimensional persistence diagrams for each image. We then fit the mixed-effect PLPQL model (described above) to the persistence diagrams with a fixed effect for Gleason grade and nested random effects of individual and replicate within individual pointwise over a grid of query points.

For the 0-dimensional diagrams (H_0)[1], we estimate a 1-dimensional persistence intensity surface, as all connected components are born at the same time ($t = 0$) and only have varying death times. Thus, the H_0 persistence diagrams for this filtration can be represented as a collection of points along the death axis, resulting in a 1D persistence intensity surface estimation problem. Here, we use a 1D Box kernel with bandwidth $h = 4$ and consider query points over a sequence of 100 points over the interval $[D_{min}, D_{max}]$, where D_{min}, D_{max} are the minimum and maximum death times across all H_0 persistence diagrams in the dataset. For the 1-dimensional diagrams (H_1), we estimate a 2-dimensional persistence intensity surface as both birth and death times vary for 1-dimensional features. Here, we use a 2D Box kernel with bandwidth $h = 15$ and consider query points over a sequence of 30×30 grid points over region $[15, 60]^2$ of the birth-death plane.

We provide results of the H_0 persistence diagram analysis in Figure 4.12. In panel (a), we see that the estimated population-level persistence intensity curves for Gleason grades 3 and 4 differ in that the intensity for grade 4 is higher for the entire range of death times in the window considered. This indicates that there are on average more connected components in the nuclei location point clouds for grade 4 images than for grade 3 images. This makes sense biologically, as higher Gleason grades are associated with more chaotic tissue structures,

[1]This is not to be confused with the null hypothesis, which we will state explicitly in this section when needed.

which often leads to smaller glands and a higher density of nuclei in the tissue, leading to more connected components in the Rips filtration. In fact, the average number of points in the 0-dimensional persistence diagrams across all images for grade 3 images is about 262 while for grade 4 images is about 375. Note that there are subtypes of Gleason grades 3 and 4 that can lead to larger and smaller, or more spread out gland structures, so this pattern may not hold for every case.

Carrying out the global hypothesis test for fixed-effect differences in the 1D intensity curves between the two grades, we obtain a p-value of < 0.01 after the Westfall-Young adjustment and taking the minimum adjusted p-value across the death time range. This indicates strong evidence against the null hypothesis of no difference in the persistence intensity surfaces between the two grades, after accounting for individual and replicate level variability. We also see in panel (c) from the pointwise adjusted p-value curve that there is strong evidence against the null hypothesis of no differences in the intensity surfaces across most of the range of chosen death times (between 10 and 40).

In panel (b), the standard deviation curves for the random effects are relatively high for the lower death times, indicating greater variability in the number of connected components with those death times across both individuals and replicates. Similarly, intraclass correlation coefficients (panel (d)) are relatively high for small death times and decreasing as death time increases. This corresponds to a maximum of about 0.4 for the replicate level ICC and about 0.2 for the individual level ICC, indicating moderate amounts of similarity at both levels of the model in the number of connected components for smaller death times. This helps justify the use of the mixed-effect model for these data, as there is also a moderate proportion of total variability being explained at both the individual and replicate levels.

Next, we provide results of the H_1 persistence diagram analysis in Figure 4.13. In panels (a)-(c), we also see that the estimated fixed-effect persistence intensity surfaces for Gleason grades 3 and 4 differ in that the intensity for grade 4 is higher in the region of average lifetime

and average persistence (similar to the H_0 case). This again makes sense biologically, as higher Gleason grades are often (though not always) associated with smaller glands, which would lead to more 2-dimensional holes in the nuclei location point clouds for grade 4 images than for grade 3 images.

In panels (i)-(f), we see that there is strong evidence against the null hypothesis of no differences in the persistence intensity surfaces between the two grades for both 0 and 1-dimensional persistence diagrams after accounting for individual and replicate level variability (p-values < 0.01) and strong evidence against the null hypothesis of no variability at both the individual and replicate levels (p-values < 0.01 for both tests). We also see from the pointwise population-level hypothesis test surface that there is strong evidence against the null of no differences in the intensity surfaces across portions of the domain where differences in the intensity surfaces are greatest. This is also generally true for each of the random-effect hypothesis tests. However, the domain of significant differences is large for the individual random effects as compared to the replicate random effects, indicating that there may be some spatial clustering of points within individuals across the domain. From panels (d)-(f) we also see estimated standard deviation surfaces for the individual and replicate level random effects are roughly similar in structure, resulting in intraclass correlation coefficient surfaces (panels (g) and (h)) that are also similar in shape but with lower magnitude for the replicate level ICC surface.

4.4 Discussion

In this work, we carried out simulation experiments to demonstrate the effectiveness of our hypothesis testing approach in detecting differences in the population-level intensity function across different levels of a categorical covariate. We found that the Westfall-Young and Romano-Wolf methods generally perform well in controlling the Type I error rate and have good power properties in this setting, while other methods such as the Holm and

Benjamini-Hochberg adjustments can be anti-conservative in some settings. However, the only method that consistently controlled the Type I error rate across all of our simulation experiments was the Westfall-Young method, which is also the only method with known theoretical guarantees for controlling the Type I error rate in this setting.

We have also applied our method to a real-world dataset of histopathology images of prostate cancer tissue, demonstrating its practical utility in analyzing complex biological data with repeated measurements and multiple sources of variability using topological descriptors. In our Gleason data analysis, the design involves repeated measurements on the same subject, which is a common sampling design in many applications. Thus, we considered a hierarchical design with replicates nested within subjects, where each replicate is also associated with a categorical covariate (Gleason grade). This model allows us to model the variability in the persistence diagrams due to both subject-level and replicate-level effects, while also allowing us to model the population-level intensity function as a function of covariates, accounting for these sources of variability. As a consequence, we are able to obtain more accurate estimates of and inferences for the population-level intensity function and better understand the sources of variability in the persistence diagrams. This is particularly important in applications where there is large variability between subjects and replicates, and unbalanced numbers of replicates per subject, which is common in biological and medical settings.

Additionally, this kind of analysis can be particularly important when conducting statistical inference in randomized controlled trials with repeated measurements on subjects, as is common in clinical settings. It is in this kind of setting where we are most often interested in hypothesis testing for differences across different treatment groups while accounting for the variability between subjects and replicates. The hierarchical mixed-effect model allows us to account for the lack of independence between repeated measurements on subjects and replicates on subjects, which is important for obtaining valid statistical inference. Added information from intra-class correlations provide insight into the process being studied and

relative sources of variation.

4.5 Future Work

The simulation experiments we have considered suggest that the minimum of the Westfall-Young adjusted p-values generally works well in controlling the Type I error rate and has good power properties in this setting. However, the properties of this estimator for the global hypothesis tests are still not well understood, and so the theoretical properties of these estimators need further investigation. Additionally, the Romano-Wolf adjustment method appears to have comparable results to the Westfall-Young method in many of our simulation experiments. For this reason, it would be worthwhile to explore the theoretical properties of the Romano-Wolf adjustment further in this setting, as the Romano-Wolf method is more flexible than the Westfall-Young method because it applies to bootstrap or resampling-based hypothesis tests. Thus, it could potentially be applied in a wider range of applications including pairwise comparisons between multiple groups (or more general contrasts) and one-sample tests against a known baseline function.

We also note that our experimental results are likely the only power-analysis results that exist in the literature that are intended for power analysis of persistence diagram hypothesis tests. However, our experiments did not use persistence diagrams as the data source, but rather functional data and point pattern data. Thus, it would be worthwhile to carry out further simulation experiments where the data source is persistence diagrams themselves. This would allow us to understand the performance of our hypothesis testing procedure better in settings that more closely resemble real-world applications. It would also be worthwhile to compare our method to other existing methods for hypothesis testing with persistence diagrams, such as the distance-based methods on persistence-diagrams of Robinson and Turner [256] or the permutation based methods on persistence intensity functions of Chen *et al.* [56]. This would allow us to better understand the relative performance of our method in

comparison to these existing methods.

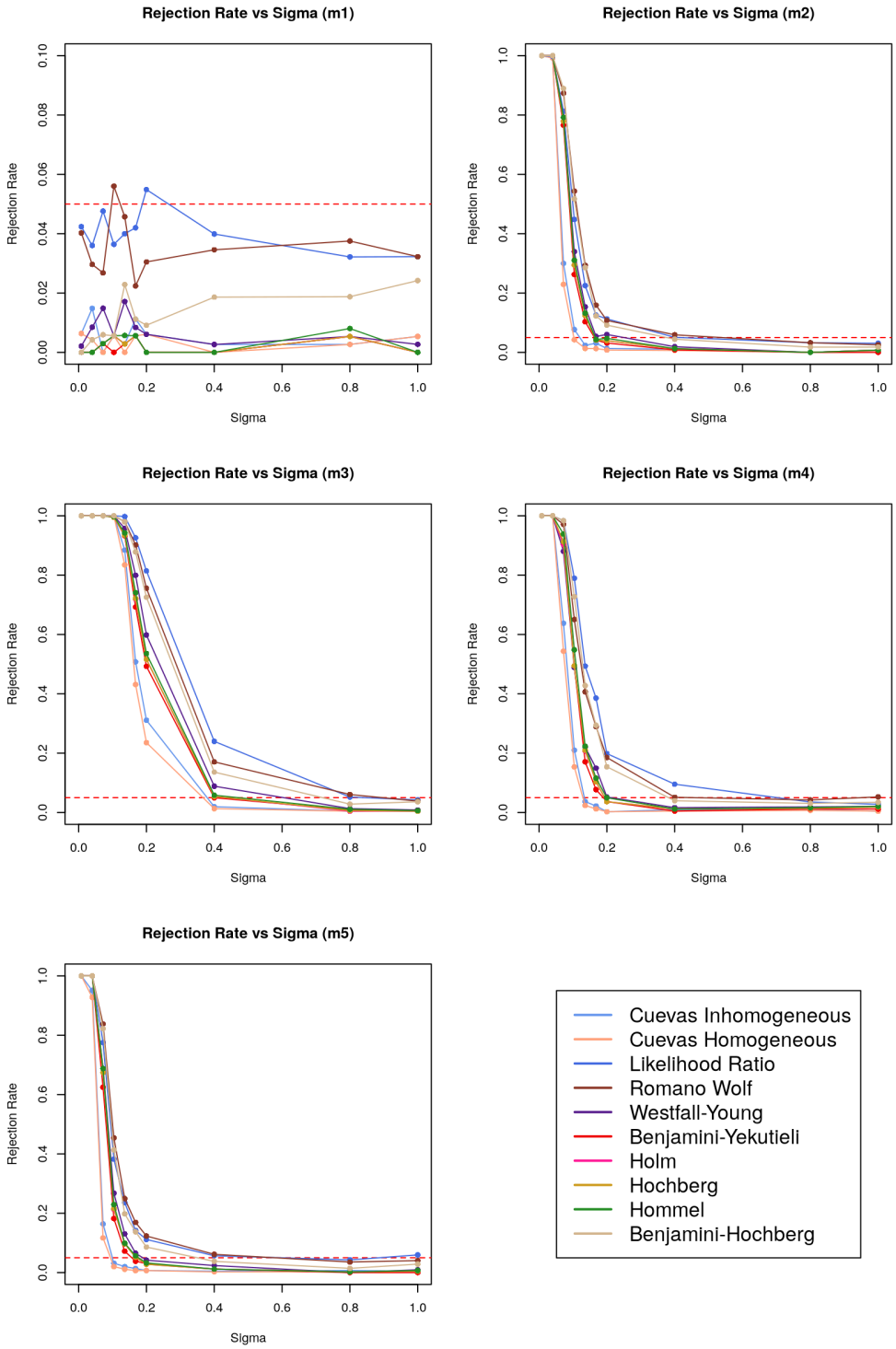


Figure 4.4: Rejection rates for 1D fANOVA under white-noise error model.

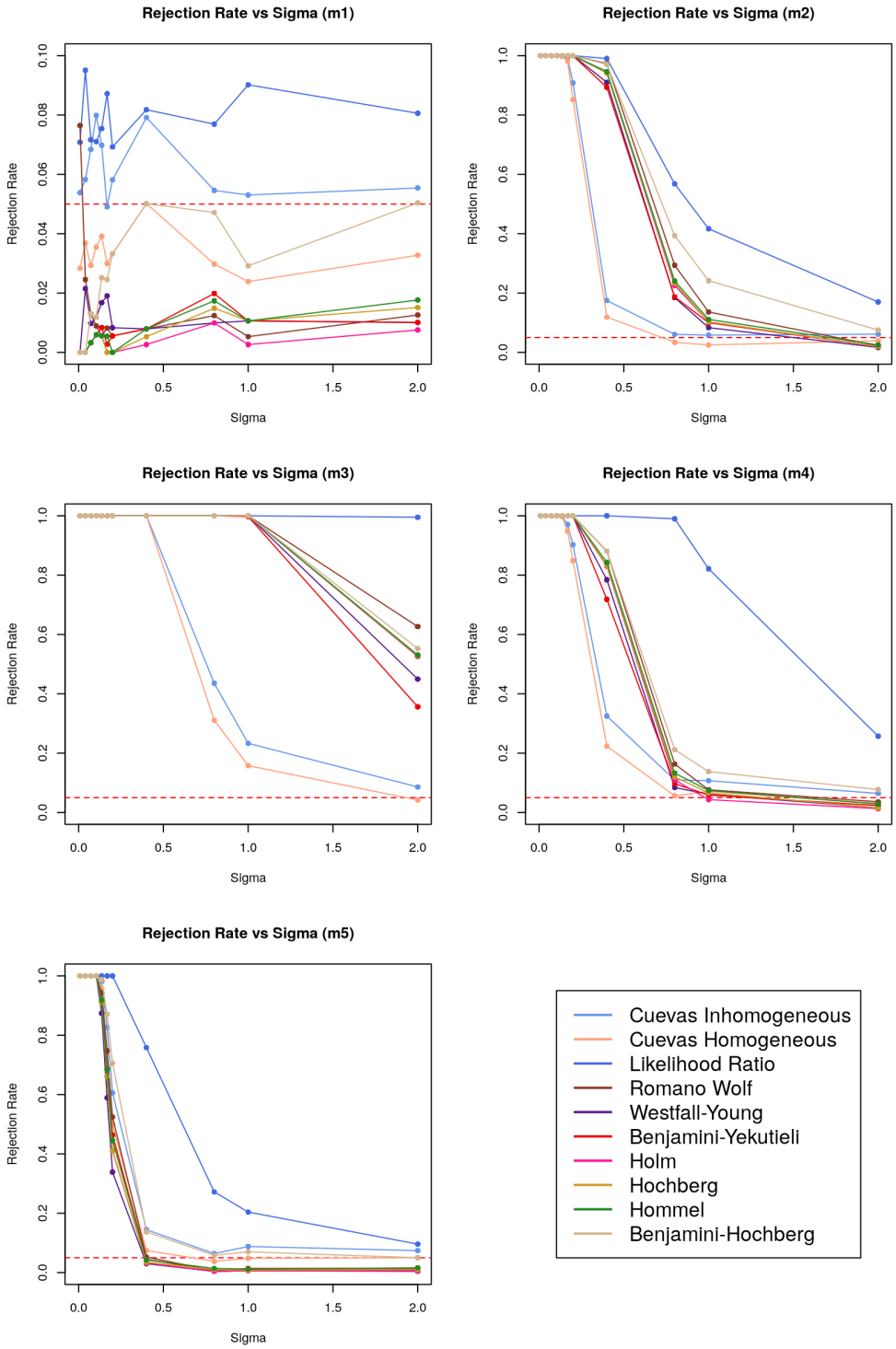


Figure 4.5: Rejection rates for 1D fANOVA under Brownian motion error model.

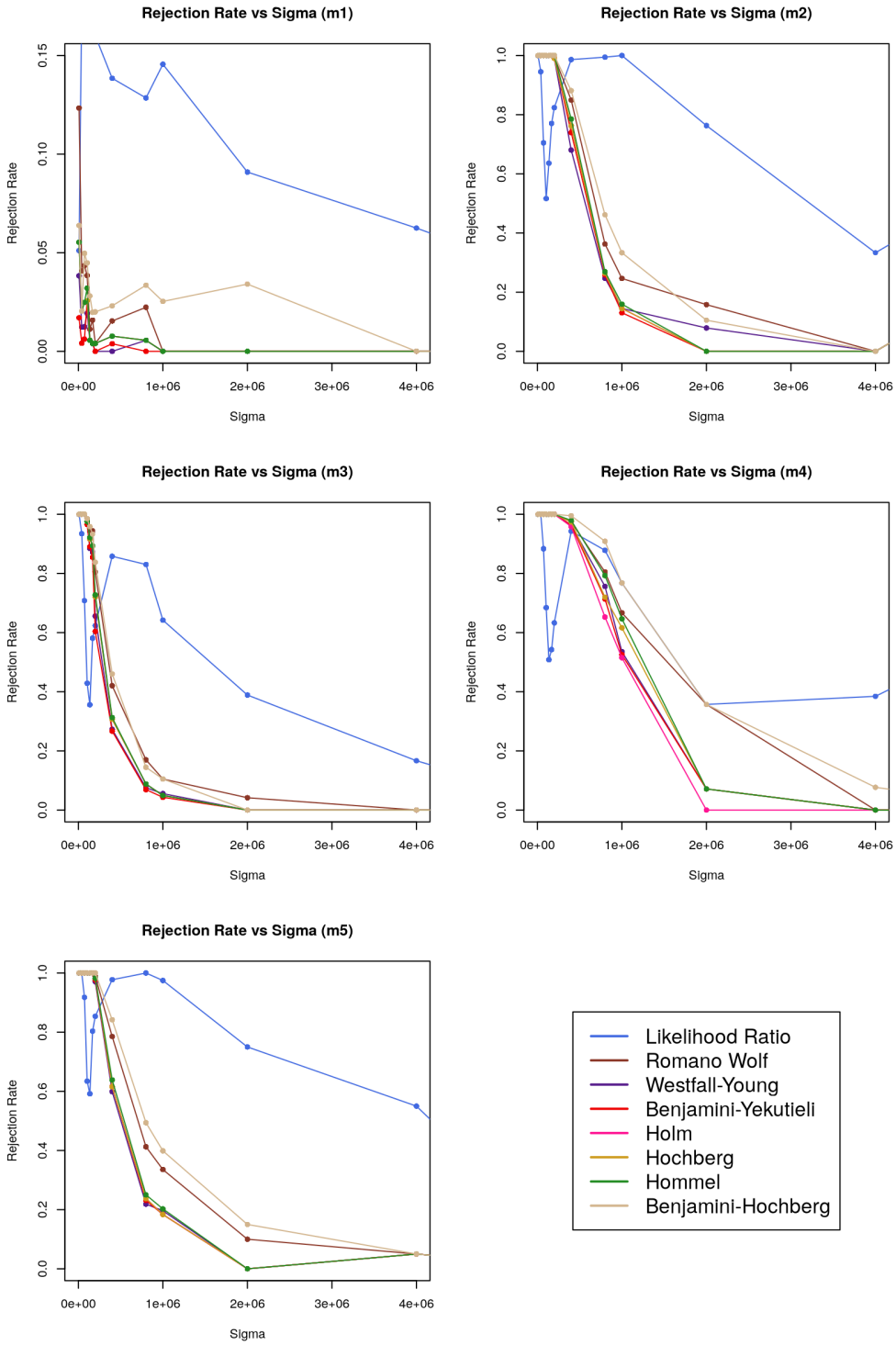


Figure 4.6: Rejection rates for 1D fANOVA point process data with Gaussian process error.

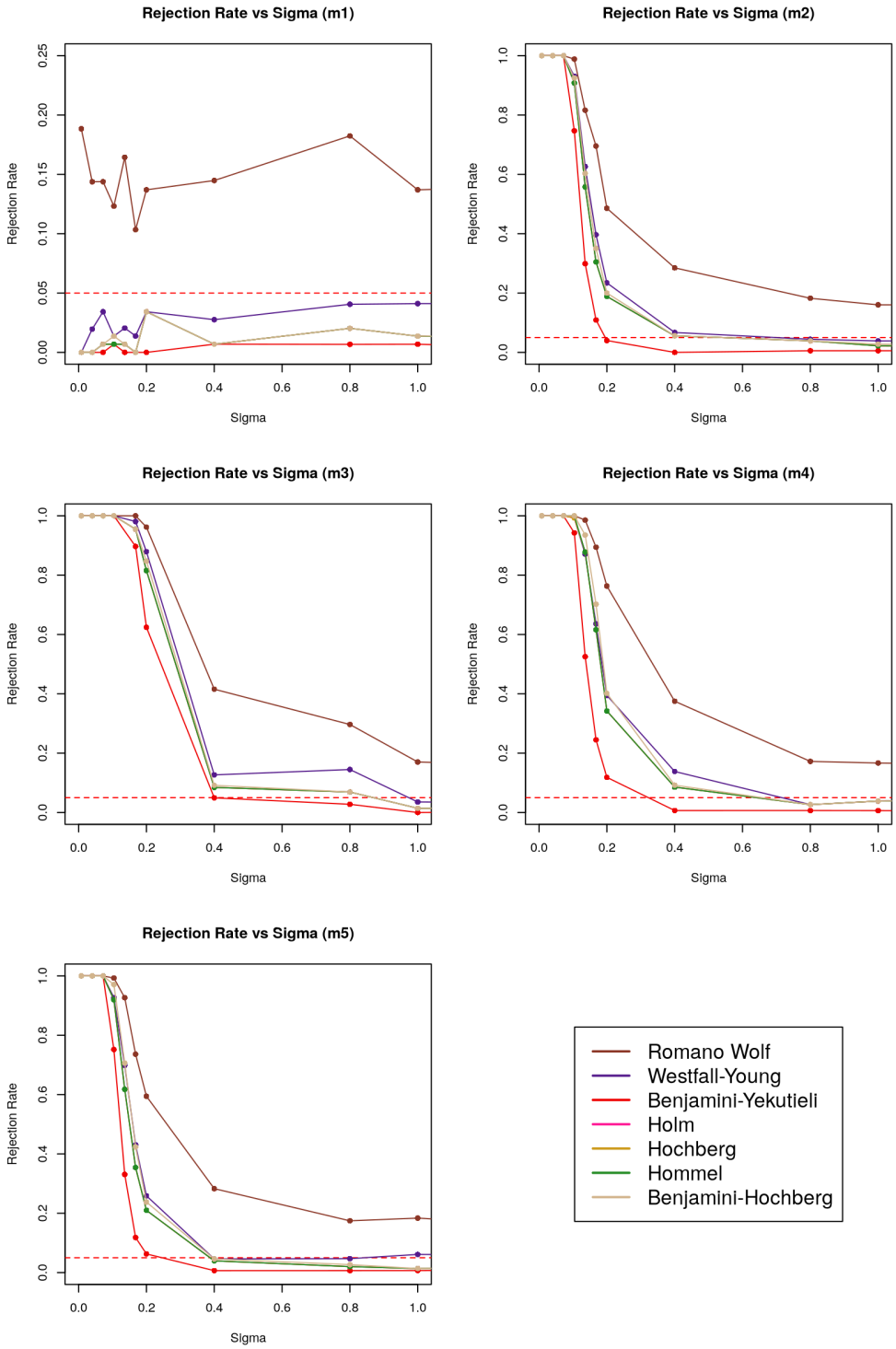


Figure 4.7: Rejection rates 2D fANOVA with white-noise error model.

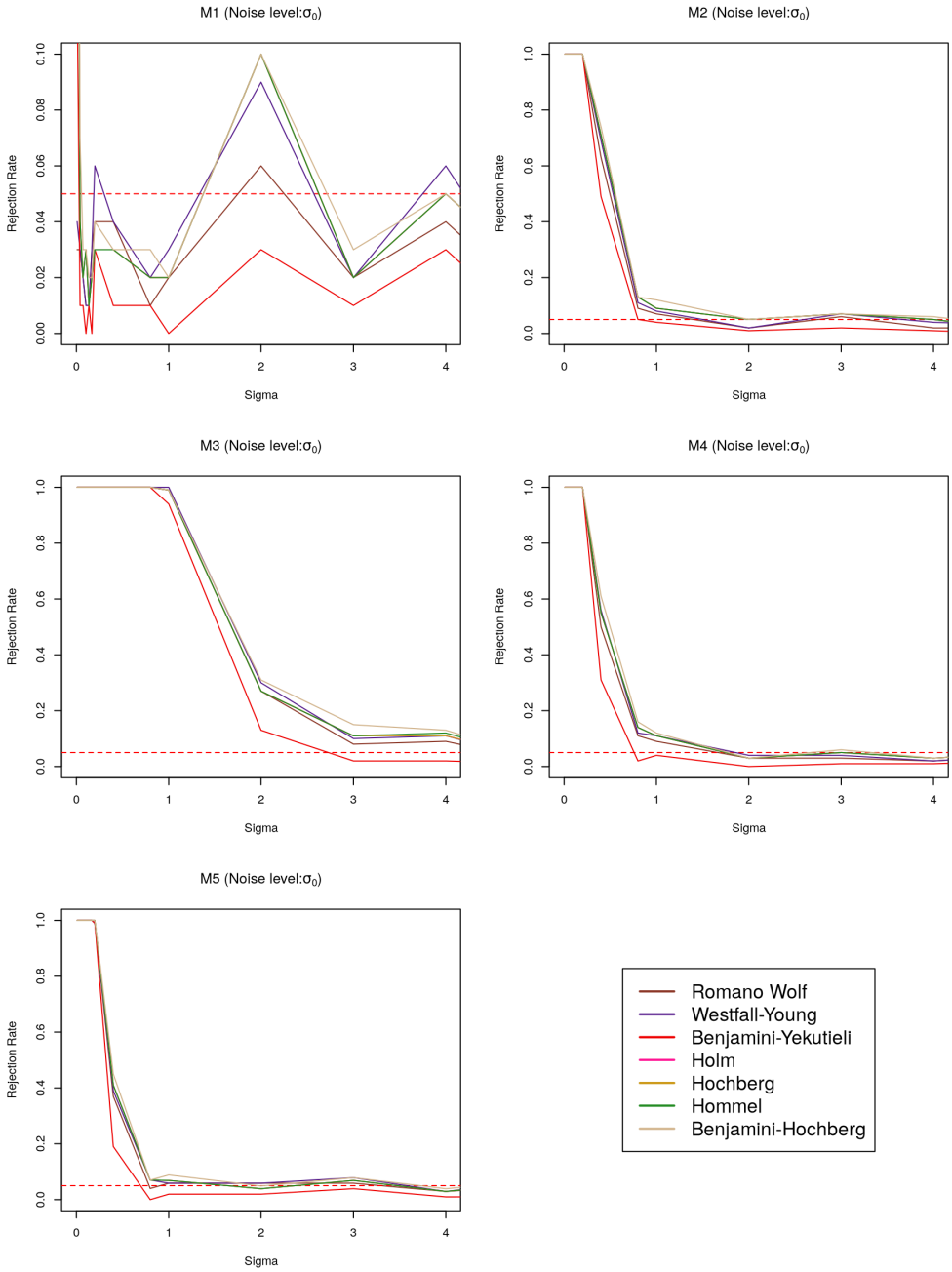


Figure 4.8: Rejection rates for 1D repeated measures fANOVA under white-noise error model, where lowest level of variance (σ_0) is varied holding higher-level variances (σ_1, σ_2) constant at 0.008.

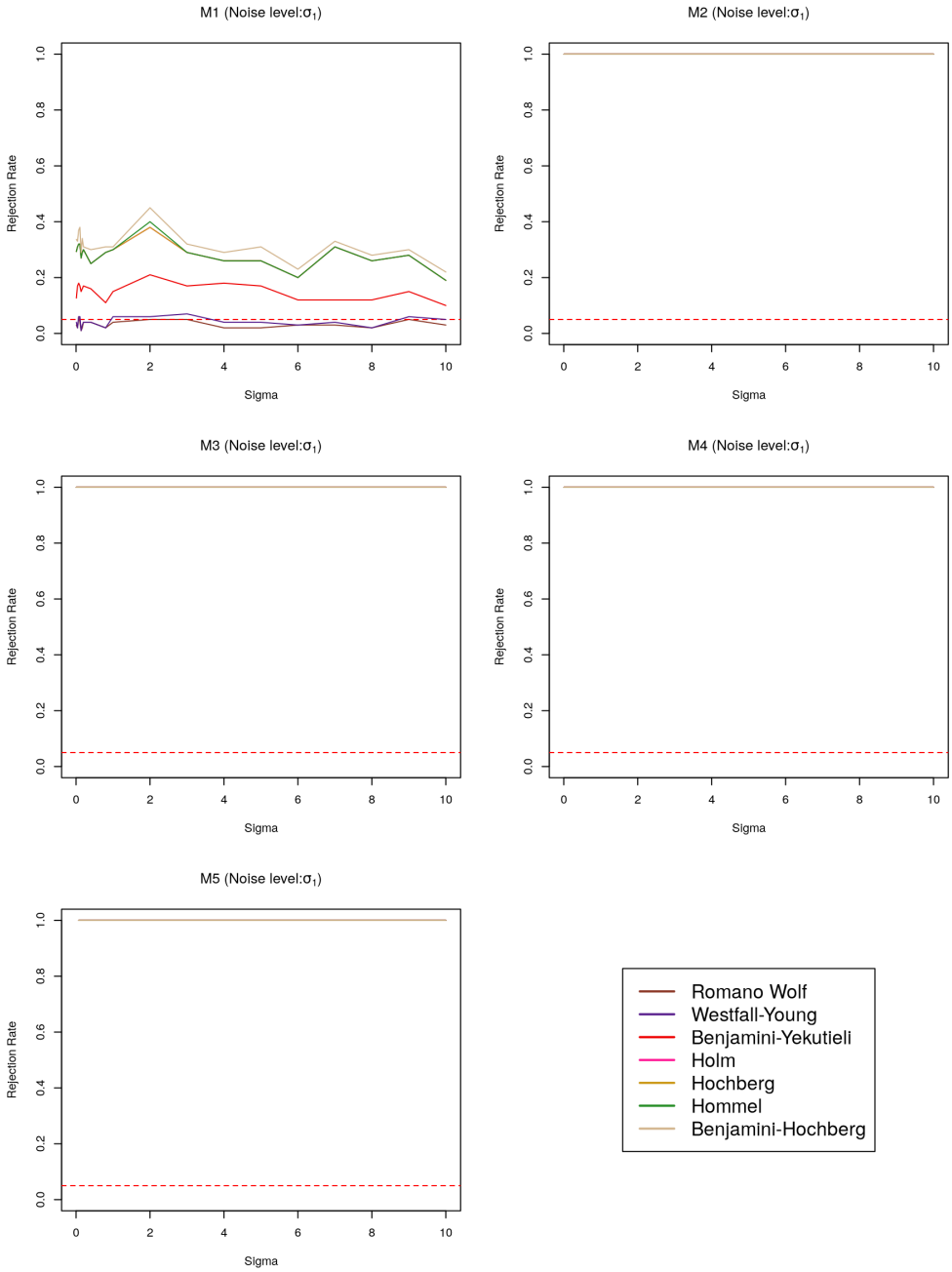


Figure 4.9: Rejection rates for 1D repeated measures fANOVA under white-noise error model, where lowest level of variance (σ_1) is varied holding other variances (σ_0, σ_2) constant at 0.008.

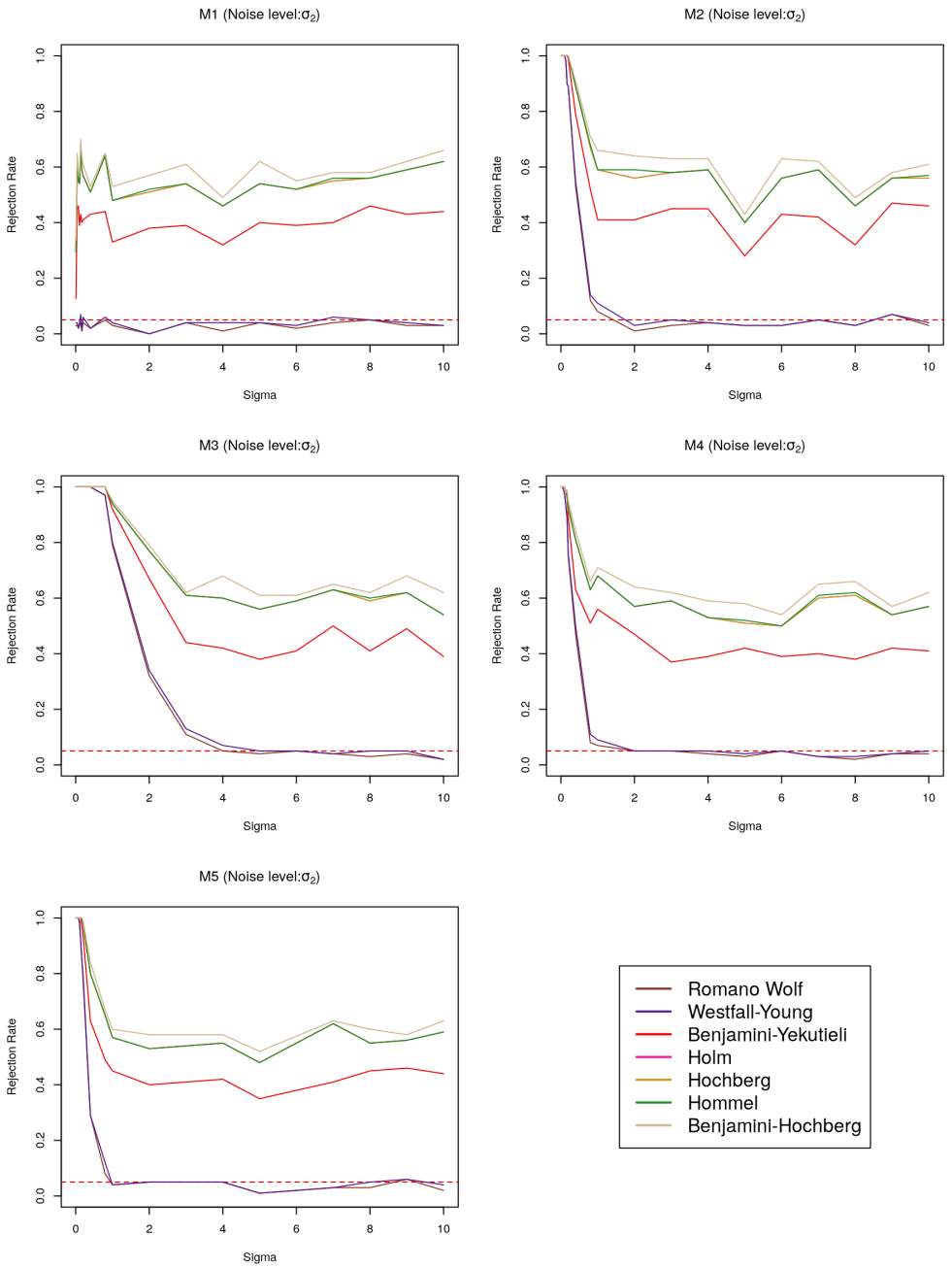


Figure 4.10: Rejection rates for 1D repeated measures fANOVA under white-noise error model, where lowest level of variance (σ_2) is varied holding other variances (σ_0, σ_1) constant at 0.008.

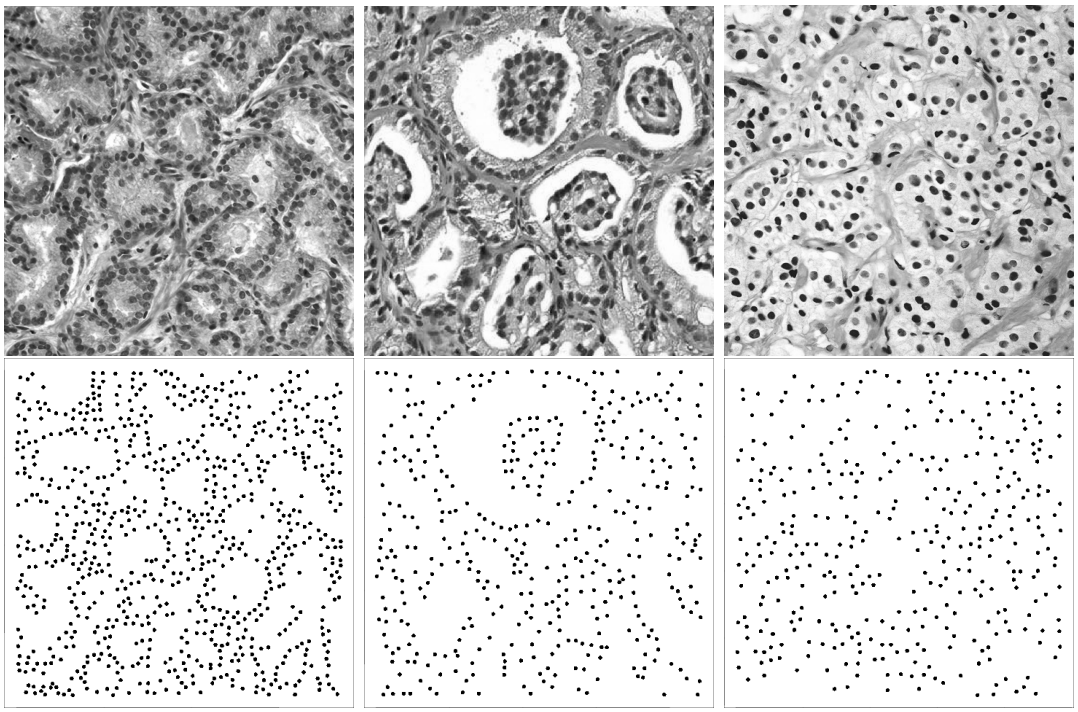


Figure 4.11: Top row gives examples of 512×512 grayscale images for (from left to right) Gleason grades 3, 4, and 5. Bottom row gives corresponding nuclei location point clouds extracted from each image using a U-net convolutional neural network.

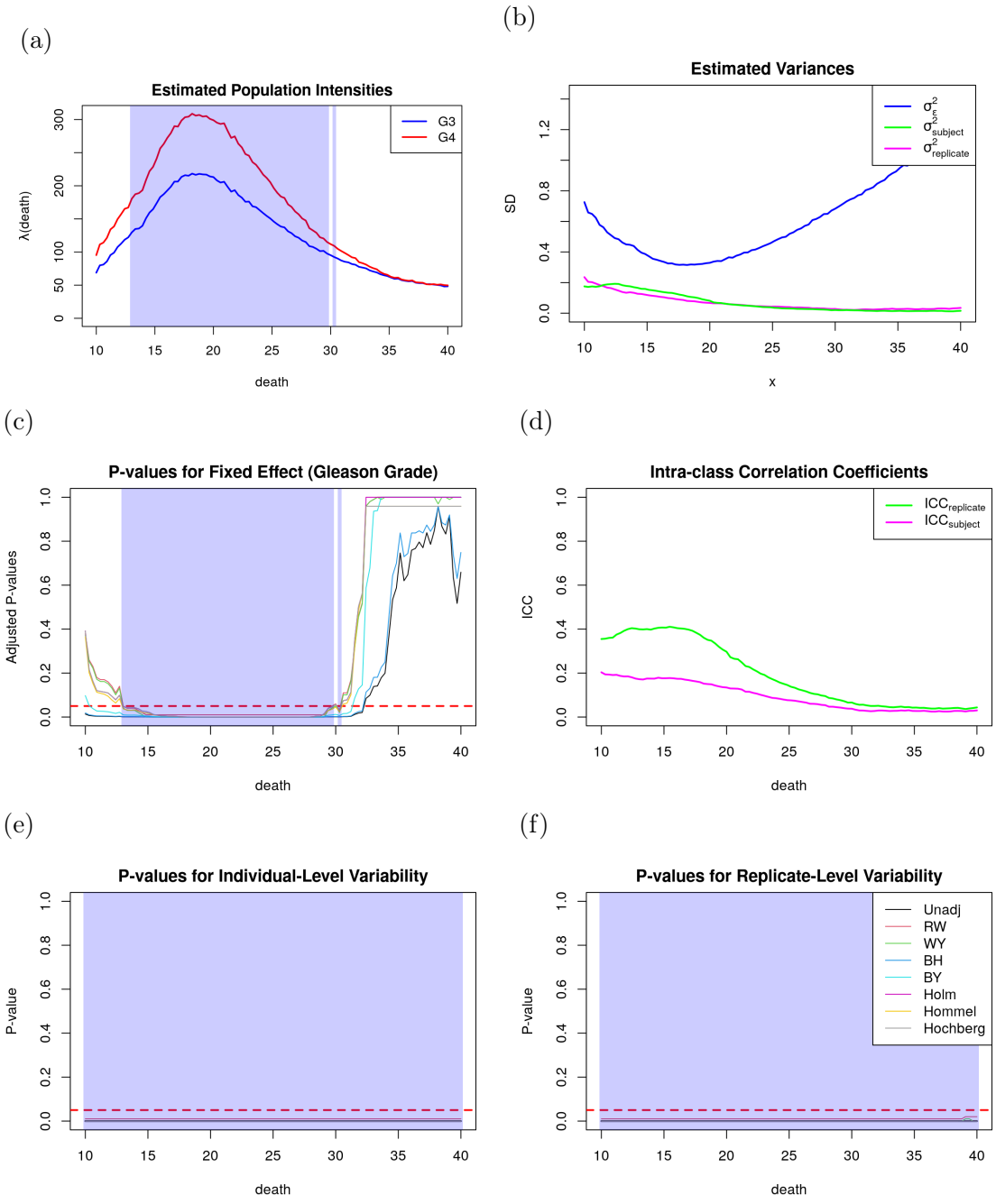


Figure 4.12: Results for 1D persistence intensity surface analysis for H_0 persistence diagrams. Panel (a) give the estimated population-level intensity curves for each grade and panel (b) gives the estimated variance curves for each random effect. Panel (c) gives the p-value curve for the fixed-effect test for differences in the population-level intensity curves between grades. Panel (d) gives the intraclass correlation coefficient curve for the subject and replicate random effects. Panels (e) and (f) give the p-value curves for the subject and replicate random-effect tests.

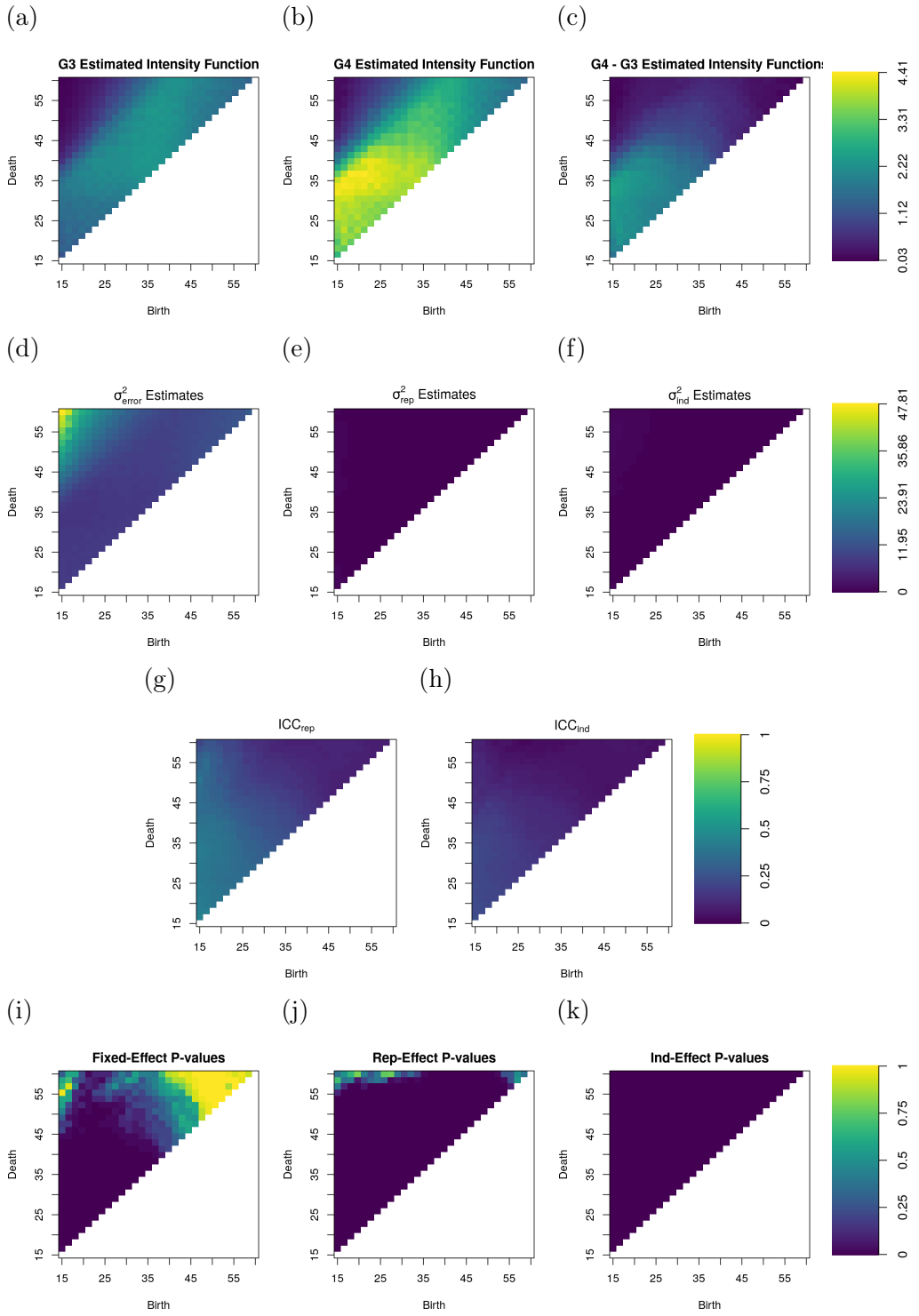


Figure 4.13: Estimated population intensity surfaces (first row), variance surfaces (second row), ICC surfaces (third row), and p-value surfaces (fourth row) for the H_1 homological features of the Gleason data.

CHAPTER FIVE

KNOT-SELECTION AND ADAPTIVITY

As stated in previous chapters, estimation of persistence intensity functions is nonparametric in nature. The space of functions that we are trying to estimate is large and complex [256, 285], and we have no prior knowledge of the underlying functional form. Thus, we need a flexible method to estimate these functions. B-splines are a popular choice for nonparametric function estimation due to their efficiency, ability to adapt to different functional forms, and interpretability (i.e. the ability to understand the model's internal mechanics). However, the quality of fit for a particular dataset is highly dependent on the choice of knot points. Thus, selecting knot points is an important problem when using B-splines for function estimation.

Stochastic search algorithms are an effective way to solve the B-spline knot selection problem. However, stochastic search algorithms often require a large number of (in this case, expensive) fitness function evaluations to find good solutions. In an attempt to mitigate this issue, we apply a modern cooperative coevolutionary algorithm – factored evolutionary algorithms (FEA) [282] – to the B-spline knot selection problem. This work was published in proceedings of the IEEE Symposium Series on Computational Intelligence [112]. In this work, we demonstrate FEA's performance on a variety of benchmark functions and compare FEA to traditional stochastic search methods. We also propose an FEA-specific method to evaluate Mean Squared Error (MSE) more efficiently that providing performance improvements for the method.

5.1 Introduction

Fitting a curve to data (*i.e.*, nonlinear regression) is an important and foundational problem for many fields, as we often obtain data which follow some unknown functional form. In Chapter 3, we described how we can use both B-splines and kernel regression estimators to estimate persistence intensity functions. Our focus in this chapter is on the B-spline estimator, and in particular, the problem of selecting knot points for the B-spline basis functions. B-spline basis functions are a popular choice to form a basis for approximating unknown functions due to their efficiency (with respect to number of parameters) and ability to adapt to different functional forms. Adaptation is enabled by the selection of knot points, which determine the underlying space of functions allowed to fit the data and hence the quality of fit for a particular dataset/knot-vector combination.

While estimating a B-spline curve with a fixed knot vector is a basic linear optimization, determining the best knot vector is notably more challenging. Finding the optimal knot vector to minimize mean squared error (*i.e.*, the full functional problem [117]) for a specific data set is a nonlinear optimization problem [152] with a potentially large number of stationary points [252].

Other issues with knot selection have been found that make this optimization particularly challenging. First, Jupp [152] showed that the "lethargy" property [151] is intrinsic to the knot selection problem, which states that the normal component of the gradient field of the objective function is zero along the main faces of the simplex defined by the knot vector. This has the consequence that the objective function has many stationary points, is non-convex, and has poor convergence for gradient based algorithms when solutions are near the boundary, which is often the case.

Second, Zhou and Shen [316] described the knot confounding problem, which arises from the interdependence of knots on the resulting quality of fit. This problem manifests as a

problem for step-wise insertion/removal algorithms where knots cannot be added, removed, or moved without modifying nearby knots or severely degrading fit. This interdependence can also manifest as the "two steps forward, one step back problem" in stochastic search algorithms [292], where some knots move closer to the optimal solution while others move away, contributing to slower convergence rates and potentially suboptimal solutions.

Finally, the curse of dimensionality is inherent to the knot-selection problem. As we consider estimating more complex functionals, we require more knot points to capture the complexity of the function. This puts greater computational demands on the optimization algorithm to find the optimal knot locations as both the size of the search space and computation of the objective function grow exponentially.

In this chapter, we apply factored evolutionary algorithms (FEA) to the knot selection problem to mitigate some of these issues. FEA is a stochastic search algorithm that separates the objective function into smaller, more manageable, overlapping subproblems and optimizes the collection of these subproblems to iterate towards a global solution [282]. We find that this novel approach to the knot selection problem is both effective and potentially more computationally efficient than traditional stochastic search algorithms. While there are many heuristic based approaches to the knot-selection problem, we focus our comparisons on stochastic search algorithms as FEA can be thought of as a meta-algorithm that (traditionally) takes other stochastic search algorithms.

5.2 Literature Review

Determining how best to solve the knot selection problem is well-studied with many potential solutions. Each solution has some way of determining both the optimal number of knots and the optimal placement of those knots. Some of the earliest methods for solving the knot selection problem date back to the late 1960s and early 1970s. For example, de Boor and Rice proposed the `SWEEP` and `OPT` algorithm [73], which alternate between adding

knots and then repositioning one at a time. At a high level, `SWEEP` can be thought of as using cyclic coordinate descent to determine the location of knots in the vector.

Various step-wise forward and backward selection procedures have also been proposed that use a variety of selection criteria (*e.g.* MSE, AIC, BIC, GCV, C_p) to guide the insertion and removal of knots. This includes work on `TURBO` [102], `MARS` [101], `LOGSPLINE`, `POLYMARS` [278], and the insertion/removal/relocation algorithm of Zhou and Shen [316].

There are also various methods based on geometric heuristics. For example, Park and Li [225] proposed a method that uses a two-stage approach where in the first stage, "dominant points" are selected and in the second, knot positions are optimized to balance inter-segment shape index distance. Both Li *et al.* [177] and Michel and Zidna [196] use heuristics based on estimates of discrete local curvature for the addition of knots. Razdan uses arc length and curvature estimates to select points of interest [249]. Aguilar *et al.* also rely on estimates of curvature for a knot readjustment scheme [7]. Some methods place knots where estimates of the fourth derivative are largest [114]. Yeh *et al.* [312] also make use of derivatives as a heuristic for knot selection.

Finally, there is a fairly large body of work emerging that use and adapt various stochastic search methods. For example, Miyata and Shen [199] and Pittman [234] use Genetic Algorithms (GA). Iglesias and Galvez [125] and Mohanty and Fahnestock [201] use Particle Swarm Optimization (PSO). Luo *et al.* [185] use Differential Evolution (DE). Galvez and Iglesias use the Firefly algorithm [106] and Galvez et al. use the elitist clonal selection algorithm [105]. Finally, Miyata and Shen [199, 200] use evolutionary algorithms with simulated annealing. Stochastic search algorithms often have good theoretical guarantees for finding an optimal solution and work well in practice, but they can be expensive computationally.

5.3 Background

Here, we provide a brief introduction to B-splines but refer the reader to the classic text by De Boor [72] for a more thorough treatment of the topic. This is followed by a discussion of stochastic search and FEA.

5.3.1 B-splines and Knot Selection

B-spline basis functions are piecewise polynomial functions of a given degree, d, defined by a nondecreasing sequence called a knot vector: $\mathbf{k} = (k_0, k_1, ..., k_p)$. B-spline basis functions can then be defined recursively through the Cox-de Boor formulation [71]. The base case is:

$$B_{i,0}(t) = \begin{cases} 1 & \text{if } k_i \leq t < k_{i+1} \\ 0 & \text{otherwise} \end{cases}.$$

Then, the general case for the i-th basis function of degree, d, is:

$$B_{i,d}(t) = \frac{t - t_i}{t_{i+d} - t_i} B_{i,d-1}(t) + \frac{t_{i+d+1} - t}{t_{i+d+1} - t_{i+1}} B_{i+1,d-1}(t).$$

A B-spline curve, C, is then defined as a linear combination of the basis functions:

$$C(t) = \sum_{i=0}^{p} \theta_i B_{i,d}(t).$$

Regression splines use this functional form to fit a curve to data according to the following model:

$$y_i = f(x_i) + \epsilon_i = \sum_{i=0}^{p} \theta_i B_{i,d}(x_i) + \epsilon_i.$$

This model is estimated by solving the least squares problem:

$$\hat{\boldsymbol{\theta}} = (\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{y}, \tag{5.1}$$

where $\mathbf{B}_{i,j} = B_{i,d}(x_i)$.

Note that model matrix, $\mathbf{B}$, is sparse (depending on the order of B-spline) enabling the system to be solved efficiently with sparse iterative solvers. In the B-spline knot selection problem, we find the knot vector that minimizes MSE for some dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$:

$$\mathbf{k}^{opt} = \arg\min_{\mathbf{k},\theta} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2.$$

Note that other criteria can be used to determine the optimal knot vector (*e.g.* AIC, BIC, GCV, or Mallow's C_p).

When using stochastic search to solve the knot selection problem, the algorithm coordinates search by determining which next knot vector to evaluate by solving the least squares problem and computing MSE. However, an important property of the knot vector is that it is sorted, so the search space is constrained to the p-simplex:

$$S_p[a, b] = \{\mathbf{k} \in \mathbb{R}^d : a = \mathbf{k}_1 < \mathbf{k}_2 < \cdots < \mathbf{k}_p = b\}.$$

This can be achieved directly by re-instantiating the search domain at each step or more simply by sorting the knot vector before each evaluation, which is the approach we use in our implementations of stochastic search algorithms and FEA. Note, however, that resorting may change the behavior of the stochastic search algorithm because it requires re-association of the variables in the knot vector with the variables in the model matrix (and hence the members of the population in the case of population-based algorithms). We believe these effects to be minimal, but they are worth noting.

5.3.2 Stochastic Search

We explore three traditional stochastic search algorithms as base algorithms to FEA: PSO, GA, and DE. All three algorithms have been applied to the knot selection problem in prior research [125, 185, 313]. The novelty of our method is in the use of FEA wrapped around these three methods. Before we describe our use of FEA, we briefly recap each of these algorithms. There are many variants of each of these algorithms, but we describe the variants we use in our implementation of FEA.

Particle Swarm Optimization PSO is a nature-inspired stochastic search algorithm where a population of particles searches for the optimal solution by updating their positions and velocities based on their "individual" and "social" knowledge [157]. Many variants of the algorithm have been developed, but the general framework involves a population of particles that iteratively update their positions and velocities based on their own best position and the best position found by the entire swarm. Mathematically, the velocity update (using `gBest` updates) from iteration t to $t+1$ is given by

$$\mathbf{v}_i(t+1) = \omega \cdot \mathbf{v}_i(t) + c_1 r_1(\mathbf{p}_i - \mathbf{k}_i(t)) + c_2 r_2(\mathbf{g} - \mathbf{k}_i(t))$$

and the position update equation is

$$\mathbf{k}_i(t+1) = \mathbf{k}_i(t) + \mathbf{v}_i(t+1),$$

where $\mathbf{v}_i$ is the velocity of the ith particle, ω is the inertia weight, c_1 and c_2 are acceleration coefficients, r_1 and r_2 are random Uniform(0, 1) numbers, $\mathbf{p}_i$ is the personal best position of the ith particle, $\mathbf{g}$ is the global best position and $\mathbf{k}_i(t)$ is the position of the ith particle at iteration t. The personal and the global best positions are then updated at each step of the algorithm if the current position has better fitness.

Genetic Algorithms GAs are a class of evolutionary algorithms that imitate the process of natural selection to find optimal solutions [136]. The general framework of this algorithm involves a population of candidate solutions that evolve over time through selection, (the means by which individuals are selected for reproduction), crossover (the means by which genetic information from two parent individuals is combined), and mutation (the means by which random changes are introduced to the offspring). Mathematically, the uniform crossover operation (a version of binomial crossover where the crossover rate is 0.5) combines genetic information from two parent individuals to create new offspring:

$$\mathbf{k}_{child,d} = \begin{cases} \mathbf{k}_{i_1,d} & \text{if } r \leq 0.5 \\ \mathbf{k}_{i_2,d} & \text{otherwise} \end{cases},$$

where $\mathbf{k}_{i_1}$ and $\mathbf{k}_{i_2}$ are two individuals selected from the population, $\mathbf{k}_{child}$ is the new offspring, and r is a uniform random variable between 0 and 1. The mutation operation introduces random changes to the offspring:

$$\mathbf{k}_{mutatedChild} = \mathbf{k}_{child} + \boldsymbol{\epsilon},$$

where $\mathbf{k}_{mutatedChild}$ is the mutated offspring and $\boldsymbol{\epsilon}$ is a random mutation vector. In single-stage tournament selection with the parents (the method we use), the new child replaces the parent if the child has better fitness than the parent. Note that there are many variants of each of these operations, and the specific implementation can vary widely across different GA implementations.

Differential Evolution DE is a population-based evolutionary algorithm that iteratively improves candidate solutions through the mutation, crossover, and selection operators [280]. Several variants of the algorithm have been developed, but the traditional algorithm uses a

population of candidate solutions that evolve over time through mutation, crossover, and selection. Mathematically, the mutation operation creates a trial vector, $\mathbf{u}_i(t)$, as a particular linear combination of three randomly selected population members

$$\mathbf{u}_i(t) = \mathbf{k}_{i_1}(t) + F \cdot (\mathbf{k}_{i_2}(t) - \mathbf{k}_{i_3}(t)),$$

where $F \in (0, \infty)$ is the scaling factor and $\mathbf{k}_{i_1}$, $\mathbf{k}_{i_2}$, and $\mathbf{k}_{i_3}$ are three distinct randomly selected individuals from the current population. The binomial crossover operation then modifies each element d from the parent population member, $\mathbf{k}_i$, randomly from the trial vector:

$$\mathbf{k}'_{i,d}(t) = \begin{cases} \mathbf{u}_{i,d}(t) & \text{if } U(0,1) \leq CR \\ \mathbf{k}_{i,d}(t) & \text{otherwise} \end{cases},$$

where CR is the crossover rate. If deterministic selection is used, the new child vector, $\mathbf{k}'$, replaces the parent vector, $\mathbf{k}$, if the child vector has lower (in the case of minimization) fitness than the parent:

$$\mathbf{k}_i(t+1) = \begin{cases} \mathbf{k}'_i(t) & \text{if } f(\mathbf{k}'(t)) < f(\mathbf{k}(t)) \\ \mathbf{k}_i(t) & \text{otherwise} \end{cases}$$

5.3.3 Factored Evolutionary Algorithms

FEA is a relatively new class of stochastic search algorithms that subdivide the problem into overlapping "factors" [282]. Each factor is given its own optimization routine and the factors compete with and share information with one another to iterate towards an optimal solution by maintaining a global solution $\mathbf{G}$, also referred to as the context vector.

Given a parameter set, $\mathbf{k} = \{\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_d\}$, with index set $\mathcal{I}^d = \{1, \ldots, d\}$, the factor architecture, $\boldsymbol{\mathcal{F}}$, is a collection of subsets of $\mathbf{k}$ such that $\boldsymbol{\mathcal{F}}_i \subset \mathbf{k}$ for each $\boldsymbol{\mathcal{F}}_i \in \boldsymbol{\mathcal{F}}$ and

$\cup_{i=1}^{|\boldsymbol{\mathcal{F}}|} \boldsymbol{\mathcal{F}}_i = \mathbf{k}$. Associated with each factor is an index set, $\boldsymbol{\mathcal{I}}_i = \{a \in \mathcal{I}^d : \mathbf{k}_a \in \boldsymbol{\mathcal{F}}_i\}$, and an optimization algorithm that searches via a subpopulation, $\boldsymbol{\mathcal{S}}_i$. In the other direction, associated with each variable $\mathbf{k}_j$ is the set of factors containing that variable $\boldsymbol{\mathcal{O}}_j = \{\boldsymbol{\mathcal{F}}_k \in \boldsymbol{\mathcal{F}} : \mathbf{k}_j \in \boldsymbol{\mathcal{F}}_k\}$, called the "overlap." Additionally, there are the subpopulations associated with the factors in the overlap set, referred to as "overlapping subpopulations" and denoted $(\boldsymbol{\mathcal{OS}})_j = \{\boldsymbol{\mathcal{S}}_k \in \boldsymbol{\mathcal{S}} : \mathbf{k}_j \in \boldsymbol{\mathcal{F}}_k\}$. After initializing the global context $\mathbf{G}$ and each subpopulation $\boldsymbol{\mathcal{S}}_i$, the FEA algorithm repeats three steps until convergence: `update`, `compete`, and `share`.

During the `update` step, each subpopulation, $\boldsymbol{\mathcal{S}}_i$, is optimized relative to the objective function, $f(\mathbf{k})$, allowing only the variables associated to its factor, $\boldsymbol{\mathcal{F}}_i$, to vary while the remaining variables, $\mathbf{r}_i = \mathbf{k}/\boldsymbol{\mathcal{F}}_i$, are held to the values given by the global context, $\mathbf{G}$.

During the `compete` step, variables in $\mathbf{k}$ are iterated through in a permuted order, $\mathbf{p} = perm(\mathcal{I})$. The subpopulations associated to the factors in overlap set $\boldsymbol{\mathcal{O}}_{\mathbf{p}_j}$, namely $(\boldsymbol{\mathcal{OS}})_{\mathbf{p}_j}$ are considered for updating variable $\mathbf{k}_{\mathbf{p}_j}$ in $\mathbf{G}$. The value of $\mathbf{k}_{\mathbf{p}_j}$ in $\mathbf{G}$ is replaced by the $\mathbf{k}_{\mathbf{p}_j}$ from the best solution found among the subpopulations in $(\boldsymbol{\mathcal{OS}})_{\mathbf{p}_j}$, so long as that solution has higher fitness than $\mathbf{G}$.

Finally, during the `share` step. Each subpopulation, $\boldsymbol{\mathcal{S}}_i$, is updated so that the nonfree variables $\mathbf{r}_i$ coincide with $\mathbf{G}$.

5.4 Methods

We now describe our method of using FEA to solve the knot selection problem. We describe both our specific implementation of FEA for knot selection and how it enables a more efficient partial fitness function evaluation.

5.4.1 FEA for Knot Selection

Our FEA algorithm (Algorithm 5.7) is based on the algorithm presented in Strasser *et al.* [282]. However, FEA has never been employed for the B-spline knot selection problem

Algorithm 5.7 FEA

Input:

f Objective function
$\mathcal{A}$ Base optimization algorithm
$\mathcal{F}$ Factor architecture
$\mathcal{D}$ Domain

Output:

$\mathbf{G}$ Global context solution

1: $\mathbf{G} \leftarrow$ initializeGlobal($\mathcal{D}$)
2: Sort($\mathbf{G}$)
3: $\boldsymbol{\mathcal{S}} \leftarrow$ initializeSubpops($f, \mathcal{F}, \mathcal{A}$)
4: **repeat**
5: **for all** $\boldsymbol{\mathcal{S}}_i \in \boldsymbol{\mathcal{S}}$ **do** ▷ `Update` step
6: **repeat**
7: $\boldsymbol{\mathcal{S}}_i$.updateIndividuals()
8: **until** Termination criterion is met
9: $\mathbf{G} \leftarrow$ `Compete`($f, \boldsymbol{\mathcal{S}}$)
10: `Share`($\mathbf{G}, \boldsymbol{\mathcal{S}}$)
11: **until** Termination criterion is met
12: **return** $\mathbf{G}$

---

before, so we modified it to allow the algorithm to run on this problem. In particular, the general case of FEA does not require its context vector to be sorted. Several modifications to the `compete` and `share` steps (Algorithm 5.8 and Algorithm 5.9, respectively) and the base algorithms were made to accommodate this requirement. In particular, a copy of the context vector, $\mathbf{G}$, is maintained and sorted before each fitness function evaluation throughout the `compete` step, and sorting occurs before evaluations within the base algorithms. Working on a copy also ensures a consistent ordering of variables in $\mathbf{G}$ throughout the competition phase.

It is natural to consider "linear" factor architectures with some amount of overlap for the B-spline knot selection problem. For example, a knot vector, $\mathbf{k} = \{\mathbf{k}_1, \ldots, \mathbf{k}_d\}$, may be broken into a linear factor architecture, $\boldsymbol{\mathcal{F}} = \{\{\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3\}, \{\mathbf{k}_2, \mathbf{k}_3, \mathbf{k}_4\}, \ldots, \{\mathbf{k}_{d-2}, \mathbf{k}_{d-1}, \mathbf{k}_d\}\}$, with a factor size of 3 and maximum overlap size of 2. This kind of architecture may help mitigate

Algorithm 5.8 FEA `Compete`

Input:

f Objective function
$\mathcal{A}$ Optimization algorithm
$\mathcal{F}$ Factor architecture
$\mathcal{D}$ Domain
$\mathbf{G}$ Context
$\mathcal{S}$ Subpopulations

Output:

$\mathbf{G}$ Updated Context

1: $\mathbf{p} \leftarrow perm(\mathcal{I}^d)$
2: **for** j = 1 **to** d **do**
3: $(\boldsymbol{\mathcal{OS}})_{\mathbf{p}_j} \leftarrow \{\boldsymbol{\mathcal{S}}_k \in \boldsymbol{\mathcal{S}} : \mathbf{x}_j \in \boldsymbol{\mathcal{F}}_k\}$
4: bestVal $\leftarrow \mathbf{G}[\mathbf{p}_j]$
5: **GCopy** $\leftarrow$ **G**
6: sort(**GCopy**)
7: bestFit $\leftarrow f$(**GCopy**)
8: $\mathbf{q} \leftarrow perm(\mathcal{I}^{|(\boldsymbol{\mathcal{OS}})|})$
9: **for** $i = 1$ **to** $|(\boldsymbol{\mathcal{OS}})|$ **do**
10: $currVal \leftarrow (\boldsymbol{\mathcal{OS}})_{\mathbf{p}_j}[\mathbf{q}_i]$.getBestSolution()$[\mathbf{p}_j]$
11: $\mathbf{G}[\mathbf{p}_j] \leftarrow currVal$
12: **GCopy** $\leftarrow$ **G**
13: sort(**GCopy**)
14: **if** f(**GCopy**) $< bestFit$ **then**
15: $bestVal \leftarrow currVal$
16: $bestFit \leftarrow f$(**GCopy**)
17: $\mathbf{G}[\mathbf{p}_j] \leftarrow bestVal$
18: sort(**G**)
19: **return G**

the knot confounding and "two steps forward, one step back" problems previously described as it allows a subsequence of knots to be changed simultaneously within the subpopulations.

This architecture also enables performance improvements by allowing "partial" fitness function evaluations. In particular, we can restrict the subpopulation domains to the knot locations neighboring the given factor. Essentially, each factor is only allowed to move its knot points within the bounds of the knot points on either side of that factor. See Figure

Algorithm 5.9 FEA `Share`

Input:
- f Objective function
- $\mathbf{G}$ Context
- $\mathcal{S}$ Subpopulations

Output:
- $\mathbf{S}$ Updated Subpopulations

---

for all $\boldsymbol{\mathcal{S}}_i \in \boldsymbol{\mathcal{S}}$ **do**
  $\boldsymbol{\mathcal{S}}_i$.contextVector $\leftarrow \mathbf{G}$
  $\boldsymbol{\mathcal{S}}_i$.updateDomain()
  $\boldsymbol{\mathcal{S}}_i$.updateFitness($f$)
return $\boldsymbol{\mathcal{S}}$

5.1 for an example of this, where knots in a factor are indicated by bold tick marks and are allowed to move within the bounds defined by its neighboring knots (indicated by the highlighted region). This allows for a partial update of the model matrix, $\mathbf{B}$, as we only need to update the columns of $\mathbf{B}$ that correspond to a neighborhood around the knots in the factor. This changes the complexity of updating $\mathbf{B}$ from $\mathcal{O}(nd)$ to $\mathcal{O}(ns)$ where $s < d$ is the factor size. It also reduces the complexity of the problem for subpopulations as the search space is reduced to a subinterval of the original search space. This in turn means there are fewer local minima within the subpopulation search space, which may lead to faster convergence and better solutions.

Because we know the knot locations not associated with a given factor do not change but stay fixed at the values given by $\mathbf{G}$ and because the model matrix $\mathbf{B}$ is sparse, we know those coefficients will remain similar to those computed on $\mathbf{G}$ from the previous FEA iteration. Thus, we can benefit from using that coefficient vector as a starting point for iterative solvers within the subpopulations. We give anecdotal evidence of this in the experiments section.

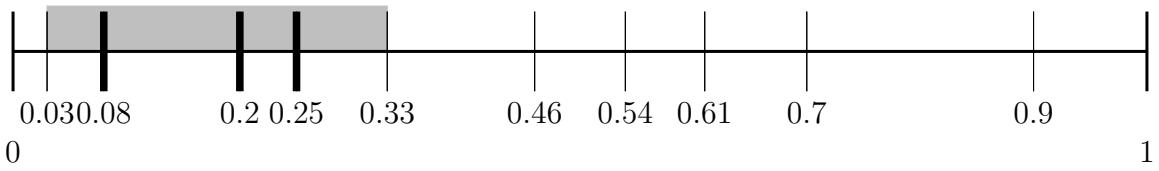


Figure 5.1: The bolded lines represent the three knots within a factor. The domain for the factor is 0.03–0.33 (knot 0 to knot 4).

5.5 Experiments

We conducted experiments to compare the performance of FEA to traditional stochastic search algorithms on a variety of benchmark functions. We did this by running the B-spline knot selection problem on each base algorithm (PSO, DE, and GA) and the FEA version of each. We used the Mean Squared Error (MSE) of the B-spline approximation as our metric for these comparisons. We considered six benchmark functions, two levels of sample size, and three levels of noise.

5.5.1 Benchmark Functions

Five of the six benchmark functions were taken from previous literature [81, 313]. Illustrations of these benchmark functions are provided in Figure 5.2. In particular, we chose their “Blocks", “Bumps", “HeaviSine", and “Big Spike" functions exactly as described in these papers. We also modified the doppler function; instead using the equation $f(t) = \sin\left(\frac{20}{t+0.3}\right)$. For the sixth benchmark, we created our own functions by producing randomly generated B-spline curves. We generated our random functions using either a Beta or Uniform distribution of 30, 40, and 50 knot points and a normal distribution of $\boldsymbol{\theta}$ to produce the random B-spline

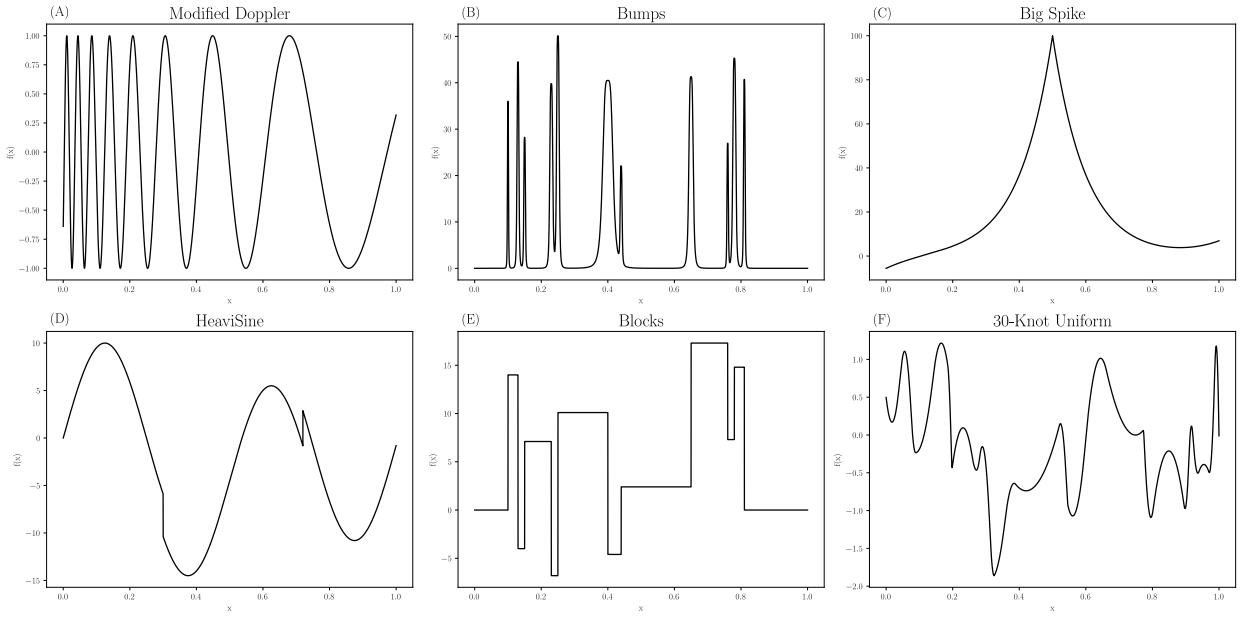


Figure 5.2: Benchmark functions used in experiments. Bottom-right (F) is the random benchmark function with 30-knots, drawn from a uniform distribution.

curves, that is:

$$\begin{aligned} \mathbf{k} &\sim \textsc{Beta}(a, b) \textit{ or } \mathbf{k} \sim \textsc{Uniform}(0, 1), \\ \boldsymbol{\theta} &\sim \textsc{Normal}(\mu, \sigma^2), \text{ and} \\ \mathbf{y}(t) &= \sum_{i=0}^{p} \theta_i \beta_{i,d}(t), \end{aligned}$$

with parameters set to $a = 1$, $b = 3$, $\mu = 0$, and $\sigma^2 = 1$.

5.5.2 Experimental Setup

We generated problems from each benchmark function by sampling points uniformly along the function, adding normally distributed noise to those points, and then running all of our stochastic search algorithms on the generated dataset for a given budget of fitness

function evaluations. Each dataset was generated according to

$$\begin{aligned} x &\sim \textsc{Uniform}(0,1), \\ y &= f(x) + \epsilon\ , \ \ \epsilon \overset{\text{iid}}{\sim} N(0,\sigma^2), \text{ and} \\ \mathcal{D} &= \{x_i, y_i\}_{i=0}^{n}. \end{aligned}$$

We used a sample size of 5000. The noise was sampled from a normal distribution with $\mu = 0$ and a σ^2 that depended on the range of the given benchmark function. For a function with range $r = range(f)$, we used standard deviation $\sigma = \frac{r}{20}$. Finally, we allowed each algorithm to run for 150,000 fitness function evaluations to allow the same fitness function evaluation budget for each method.

5.5.3 Hyperparameter Tuning

We gave each of the traditional algorithms a population size of 1000 and tuned the base algorithms' other parameters, including the number of knots, through Bayesian optimization [272]. By using Bayesian optimization to tune the number of knots, we avoided some of the issues with incremental addition and removal of knot points.

FEA had a few more parameters to tune, including the factor size and overlap allowed for the factor architecture, which were also tuned via Bayesian optimization. The additional number of parameters to tune in FEA means that tuning the algorithm is more difficult than tuning the base stochastic search algorithms alone. That said, we allowed FEA's tuning to run for the same amount of iterations as the base stochastic search algorithms.

5.5.4 Partial Fitness Function Evaluation

Recall that Equation 5.1 can be solved efficiently using sparse iterative solvers. These solvers can be provided a starting guess for the solution. As a simple illustrative experiment to show potential performance benefits of a partial update, we benchmarked all of the iterative

sparse linear solvers in the scipy v1.14.1 Python library [228] using coefficients found on a prior solution and a small subset initialized to zero. In particular, we generated a dataset with $n = 5000$ data points on the modifed doppler benchmark function (see Figure 2a) with a noise level of $\sigma = \frac{r}{20}$ and estimated a model using $K = 80$ knot points equally spaced across the domain. We estimated the model, and saved its coefficient vector $\boldsymbol{\theta}$ with knot points 40 through 50 zeroed out (corresponding to a factor size of 10). Then over 10 averaged replicates, we compute the ratio of time to solve the least squares problem when using this vector as a starting point to the time of not using a starting point. We stress that this experiment is merely illustrative, as timing benchmarks are heavily biased by implementation. Nonetheless, showing a performance increase in the iterative sparse solvers illustrates how the potential benefits of this "partial" update may be achieved. The results of this illustration are given in Table 5.1. In every case, we see a sizeable improvement in performance, with some solvers running nearly 7× faster when provided a good starting point. A similar strategy can be employed when using SGD type solvers to solve Equation 5.1.

5.6 Results

We provide results of our experiments comparing FEA to the base algorithms alone in Table 5.2 with the smallest average MSE across 30 simulations for each function in bold. The results generally show that FEA either outperforms or is competitive with the best performing algorithm on all of the benchmark functions. Additionally, the DE and GA variants of FEA generally outperformed the PSO variant. It is interesting to note that DE generally performed worst on the benchmark functions while the FEA variant of DE generally performed the best. Alternatively, the base PSO algorithm was fairly competitive with all of the other algorithms while the FEA variant appears to have degraded its performance in a number of cases. Interestingly, as the dimensionality of the problem increased in the random function experiments, the relative performance of the FEA algorithms appeared to improve

Table 5.1: Performance ratios for scipy iterative sparse solvers

	Performance Ratio
BICG	0.131455
BICGSTAB	0.37907
CG	0.133664
CGS	0.4305751
GMRES	0.431430
LGMRES	0.60955
MINRES	0.151452
QMR	0.1404274
GCROTMK	0.648662

compared with the base algorithms.

5.7 Discussion

The B-spline knot selection problem is an inherently difficult optimization problem due to the large number of local optima in the objective function and the large search space. Traditional stochastic search algorithms have been used to solve this problem effectively, but they are often slow and can get stuck in local optima. The results of our experiments show that FEA is competitive with or outperforms traditional stochastic search algorithms on the B-spline knot selection problem for the benchmark functions considered. Notably, this was achieved with each algorithm in the experiments being given the same budget of number fitness function evaluations. We have also shown that FEA can be modified to allow for partial fitness function evaluations, which can lead to potential performance gains.

We believe that FEA is a promising approach to the B-spline knot selection problem and

Table 5.2: Average MSE for experimental results of benchmark functions with $n = 5000$, $\sigma = \frac{r}{5}$, over 30 simulations. Results in bold are smallest average MSE for each function.

Function	DE	GA	PSO	DE FEA	GA FEA	PSO FEA
BigSpike	27.9663	27.0731	**26.5264**	27.5446	26.8739	26.6234
Blocks	3.0990	1.6143	1.5496	**1.4277**	1.4696	1.5751
Bumps	19.5134	6.6819	6.8302	**6.2533**	6.2911	8.0621
HeaviSine	1.4825	1.5047	1.4710	**1.4393**	1.4933	1.4869
MDoppler	0.0130	0.0102	**0.0097**	0.0100	0.0107	0.0108
Unif30	0.0470	0.0348	0.2384	0.0348	0.0344	**0.0339**
Unif40	0.0417	0.0506	0.0330	**0.0310**	0.0311	0.0321
Unif50	0.0738	0.1397	0.0556	0.0486	0.0490	**0.0481**
Beta30	0.0426	0.0482	**0.0341**	**0.0341**	0.0344	0.0345
Beta40	0.0901	0.1267	0.0699	0.0515	**0.0502**	0.0534
Beta50	0.0860	0.1170	0.0600	**0.0459**	0.0463	0.0545

that further research in this area is warranted. Future work includes scaling our experiments to more complex (and hence more needed knot points) functionals and work that can help explain how or when FEA may perform better than the base algorithms. We also wish to develop more optimized implementations of FEA for the B-spline knot selection problem, as our current implementation is not optimized for performance. This can help us to better understand the potential performance benefits of FEA and to make it more practical for use in real-world applications.

In this work, we have focused on the B-spline knot selection problem in the context of regression. As we discussed in the previous chapter, B-splines can also be used for intensity measure estimation and density estimation using fine-pixel approximations and

estimating a generalized linear (mixed) model. In particular, we are interested in estimation of persistence intensity functions for use in topological data analysis. The traditional treatment of this problem is to use a kernel density estimator with a fixed bandwidth. However, this approach can lead to oversmoothing or undersmoothing of the data as it does not allow for local adaptivity. This is also a well studied problem in the density estimation and kernel regression literature with many solutions that allow for variable bandwidths across the domain [2, 40, 42, 127]. Local adaptivity here is directly analogous to the knot selection problem in regression, as the bandwidth can be comparable to the knot spacing in regression. Thus, for future work, we are interested in using FEA to solve the knot selection problem in this context as well and to compare its performance to these other adaptive bandwidth selection methods, in particular in the context of persistence intensity estimation. We wish to explore this problem in the context of persistence intensity estimation as it is a problem that has not been well studied in the literature. We hypothesize that not allowing for local adaptivity in persistence intensity estimation leads to biased estimates of the true underlying persistence intensity function which can lead to suboptimal performance in downstream tasks such as modelling, clustering, and classification.

Finally, we are interested in using B-splines for surfaces in higher dimensions. FEA is a promising approach to this problem as it can be used to break the problem into smaller subproblems that can be solved more efficiently. This approach can be adapted to the multivariate B-spline knot selection problem by using a tensor-product B-spline basis and defining the factors in terms of the knot points in each dimension. However, as we will discuss in the next chapter, the tensor-product B-spline basis used in the multivariate case can lead to an unnecessarily large number of basis functions and hence a very large search space. For this reason, we are also interested in exploring alternative hierarchical B-spline bases that can help to mitigate this issue and make adaptivity in higher dimensions more computationally tractable.

CHAPTER SIX

TRUNCATED HIERARCHICAL B-SPLINE REGRESSION

In the previous chapters, we discussed the importance of choosing a flexible function representation for estimating persistence intensity functions and provided a method for solving the knot selection problem to give adaptivity in the function representation for B-spline regression. However, this approach is limited when modelling higher dimensional surfaces, and persistence intensity functions are typically defined in 2D. The main issue that arises with tensor-product B-spline representations is that when adding knot points, we are forced to add an entire row or column of control points (*i.e.*, where the new control point intersects control points in the other directions). This can lead to extra control points being added in locations where we do not need them (*i.e.*, where the true function is smooth and can be well represented with fewer basis). As a consequence, we incur a greater computational burden in the estimation process and have potential for over-fitting in some parts of the surface and under-fitting in other parts of the surface (via the bias-variance tradeoff). In an extreme case, we may be unable to (without an overall reduction in fit) add enough control points in some local region of the surface where the true function is more complex because we are forced to add too many control points in other parts of the surface where the true function is smooth.

In this way, tensor-product B-spline representations are fundamentally limited for data-adaptive function estimation. Hierarchical splines [100, 111] are a class of refinable splines that allow for local refinement of the surface without the need to add an entire row or column of control points, which leads to a more efficient and flexible function representation for surface estimation. In this chapter, we consider the use of truncated hierarchical B-splines, introduced by Giannelli *et al.* [111], for a more efficient, flexible, and data-adaptive function representation for non-parametric regression estimation.

6.1 Introduction

Regression splines are an important class of regression methods that use spline basis functions to model the relationship between a response variable and one or more predictor variables. Incorporation of adaptivity into the representation typically involves some data-driven approach that allows smoothness parameters to vary across the domain of the function. As we discussed in the previous chapter, for B-spline representations, this can be achieved by solving the knot selection problem, which is the problem of selecting the number and location of knot points used to construct the B-spline basis. This adaptivity leads to improved estimation accuracy and reduced computational burden (once the knot-selection problem is solved) compared to traditional fixed knot width spline regression methods. However, solving the knot-selection problem to achieve adaptivity in the tensor-product B-spline setting carries the issue of potentially adding more control points than are necessary across the domain of the function when extra control is needed within some local region on the surface. This issue is problematic when modelling surfaces in more than one dimension as the extra number of parameters to be estimated can be fairly large in practice.

To illustrate this problem, consider Figure 6.1, which gives an example of the issue with adding knot points in the tensor-product B-spline setting. Panels (a) and (b) show the original knot points in gray and the addition of two knot points in each direction in black. Notice that an entire row and column of control points are added in the tensor-product representation (panel (a)), while the addition of control points are limited to the top left quadrant of the surface in the THB-spline representation (panel (b)). In this simple example, if we add extra control in the top left quadrant of the surface, then for the tensor-product B-spline we need an additional eight parameters for the two knot points added in each direction and in locations that we may not need them. This adds unnecessary computational burden and can lead to over-fitting in regions where fewer basis functions are needed to represent the true

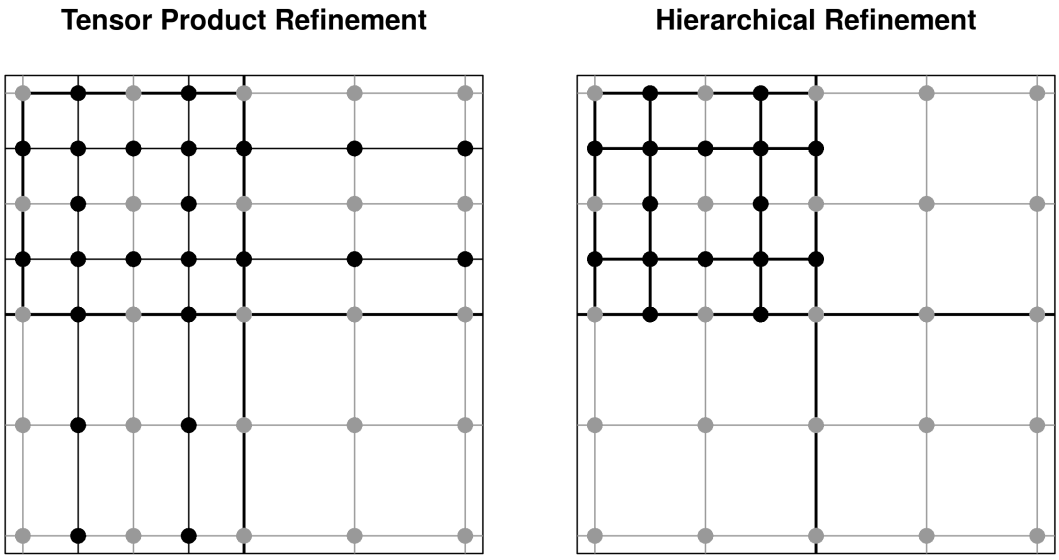


Figure 6.1: An example of the issue with adding knot points in the tensor-product B-spline setting. Each panel shows two knot points added in each direction, with the original knot points in gray and the new knot points in black. The left panel gives knot refinement in the tensor-product B-spline setting while the right panel in the THB-spline setting. Notice that an entire row and column of control points must be added in the tensor product setting, leading to an extra eight parameters in the tensor-product case.

function. The figure illustrates a fairly minimal example of the issue with adding knot points in the tensor-product B-spline. However, the issue compounds as the number of knot points in each dimension increases.

To address this issue, we use Truncated Hierarchical B-splines [111] (THB-splines), which allow for local refinement of the surface without the need to add an entire row or column of control points. This leads to a more efficient and flexible function representation for surface estimation, particularly in dimensions greater than one. In this work, we propose a model building strategy for searching the space of possible THB-spline parameterizations for estimating regression functions. In particular, we use a greedy local forward step-wise

selection strategy based on a quad-tree partition of the domain to refine the THB-spline basis according to a local drop in sums of squared-errors (SSE). This strategy allows us to efficiently search the space of possible THB-spline parameterizations and to find a good model for the data. We compare this approach to a tensor-product B-spline regression approach with fixed knot-width parameterizations and show that our approach can lead to improved estimation accuracy. In what follows, we give a brief review of the literature on adaptive regression splines and refinable splines. We then give a detailed description of the Hierarchical and Truncated Hierarchical B-spline bases. Next, we describe our model building strategy for estimating functions using THB-splines, which is the main contribution of this chapter. Finally, we present some experimental results on the performance of our approach and compare it to a tensor-product B-spline regression approach. We conclude with a discussion of our findings and discussion of future directions for research in this area.

6.2 Literature Review

Several approaches over the years have been developed for adaptive regression splines. However, as with the B-spline regression case, methods are typically developed for the 1D case. There are, however, some notable exceptions to this. As stated previously, all adaptive approaches have some mechanism for allowing the smoothness parameters to vary across the domain of the function.

One important example of adaptive regression includes multivariate adaptive regression splines (MARS) [101]. In this work, Friedman developed a method for building regression models using a linear combination of basis functions that are either constant, hinge functions, or products of hinge functions. This simple form allows for a flexible and adaptive function representation that scales well to higher dimensions. `MARS` builds the model in two stages, a forward stepwise stage where basis functions are greedily added to the model according to a lack-of-fit criterion, and a backward stepwise stage where basis functions are removed from

the model according to a generalized cross-validation criterion. MARS can be considered a generalization of recursive partitioning but replaces the piecewise constant function representation with a more flexible truncated power spline basis function. The THB-spline basis approach we use can also be considered a generalization of recursive partitioning, but replaces the piecewise constant function representation with a more flexible B-spline basis surface. Further, our approach maintains a local likelihood-ratio test, but instead uses it for a stopping criterion for model building. Finally, MARS takes a build and prune approach to model building, while our approach just takes a forward stepwise approach to model building. Then we rely on tuning the selecting number of parameters in the final model. However, in the future we plan to consider a penalized regression approach to handle model complexity.

Rogers [257] extended the approach of MARS to instead use genetic algorithms for finding a full model instead of the forward stepwise approach used by Friedman. This approach is more computationally intensive than the forward stepwise approach, but it allows for a more thorough search of the model space and may potentially lead to better models. Similarly, a stochastic search algorithm approach to searching the space of THB-spline refinements instead of the greedy forward stepwise approach we currently use would likely lead to better models, but it would also be more computationally intensive.

Luo and Wahba defined an adaptive approach to regression splines called Hybrid Adaptive Splines (HAS) [186], which can be thought of as a combination of MARS and a penalized regression approach. Similar to the MARS approach, the method uses a forward stepwise procedure to build a model, placing basis functions sequentially to (greedily) minimize the residual SSE in the model. Finally, instead of taking a pruning approach as with MARS, the method applies a penalized regression approach to provide the final model. The method extends to the multivariate case and was applied to a estimating global average winter temperatures over the surface of the globe. Similar to this method, we also are interested in applying a penalized regression approach after our forward stepwise procedure to provide

a smoother final model instead of our current approach of using cross-validation to select the number of parameters in the final model. However, we are considering a P-spline based penalty [88] applied to the THB-spline representation instead of the second derivative penalty used in HAS, as use of the P-spline penalty carries additional computational benefits [89].

Crainiceanu *et al.* [64] developed an approach for spatially adaptive Bayesian P-splines that is applicable to the multivariate case. The method uses the mixed model form of P-splines to allow for a Bayesian formulation of the model and incorporates a spatially adaptive penalty parameter that allows for different levels of smoothness across the domain of the function. MCMC sampling is used for estimation, which can be computationally intensive, but we believe other estimation strategies could be used to make the method more computationally efficient using variational methods (e.g., using INLA [260] or TMB [164]). This approach is of particular interest to us as taking a Bayesian approach to estimation, we believe, would allow us to naturally consider more complicated sampling designs such as the repeated measures setting that was the focus of Chapter 3.

Donoho and Johnstone [81] developed an approach called `SureShrink`, which is a method that adaptively thresholds empirical wavelet coefficients. This approach is based on the idea of using a discrete wavelet transform of noisy data and then applying a soft thresholding to the noisy wavelet coefficients. The amount of thresholding is level-dependent and estimated using Stein's unbiased risk estimate (SURE). This method is computationally efficient and can be applied to higher-dimensional data.

Goepp *et al.* [113] use an Adaptive Ridge procedure to approximate an L_0 penalty for knot selection in the B-spline regression setting. The method results in a sparse B-spline representation that is adaptive to the data and can be applied to higher-dimensional data. However, for higher dimensions, the method still uses a tensor-product B-spline basis and so is still at best only able to approximate the optimal knot selection in the tensor-product setting. Also, because it sparsely selects knot points, it still suffers from the computational

burden of starting with a potentially large number of parameters in the multi-dimensional setting.

Hierarhical B-splines were originally developed by Forsey and Bartels in the computer graphics community for the purpose of modeling surfaces in 3D [100]. However, more recently, refinable spline methods, including THB-splines that were developed by Giannelli *et al.* [111], have been developed and studied in the isogeometric analysis (IGA) community. IGA is primarily concerned with integrating methods for analysis of PDEs and Computer Aided Geometric Design (CAGD) into a single methodological framework [142]. In this field, they are interested in using refinable splines for efficient solving of PDEs and the related problem of scattered data approximation and interpolation (e.g. see [174, 289]) that is more concerned with surface reconstruction. Other refinable spline methods have also developed in this community and in the CAGD community, namely T-splines [265] by Sederberg *et al.* and LR-splines [80] by Dokken *et al.* While the goals of these communities are different than our goals for non-parametric regression, the spline methods developed in them are also of interest to us as they may provide us with other refinable splines with different advantages and disadvantages compared to THB-splines for non-parametric regression estimation in terms of computational efficiency, interpretability, and flexibility. However, our focus in this chapter is on the use of THB-splines.

Adaptivity in the IGA community also occurs via refinable spline methods and is also achieved by finding a data-driven way to refine the spline basis according to some criterion. In these cases, the adaptivity is typically for the purpose of efficiently solving PDEs or for scattered data approximation. In the original THB-spline paper by Giannelli *et al.* [111], they give three different refinement strategies for constructing the hierarchical domains and tested them out on the scattered data approximation problem. The three strategies were a simple union of all cells with a local error above a given threshold, the same strategy but with an offset ring around the marked cells, and taking all of the cells for any cell marked in

lower hierarchical levels. These varying refinement strategies highlight how the mechanics of refinement can lead to different amounts of error in the approximation. Similarly, Skytt and Dokken [270] considered various refinement strategies for scattered data approximation with LR-splines. They found that the choice of refinement strategy can have a dramatic impact on the approximation error and computational efficiency of the method. Finally, Bracco *et al.* [36] considered an adaptive scattered data fitting method based on thresholding the smallest singular value of the local collocation matrix associated with a local least squared problem. We similarly base our refinement on estimating local models but instead use a local drop in SSE as a criterion for refinement.

Buffa *et al.* [46] give a modern review of adaptive methods for isogeometric analysis. They discuss various methods for adaptivity and refinement strategies in the context of IGA and the finite element methods and boundary element methods. Coradello *et al.* considered adaptive methods for various PDE problems using THB-splines [60]. They used an *a posteriori* error estimator for the Finite Element Method of the PDE to guide the refinement of the THB-spline basis. They found that their adaptive method led to improved efficiency with respect to degrees of freedom compared to a uniform refinement method.

Finally, within the statistics community, refinable splines have not been widely studied. However, one notable exception to this is the work of Kohler in 1999 [162] who showed conditions for strong universal consistency of a regression estimators based on hierarchical B-splines. He provided a data-dependent choice of knots so that universal consistency can be achieved for regression estimation. Since Hierarchical B-splines and THB-spline bases have the same span for the same refinement mesh, these results should also apply to regression estimation with THB-splines. However, the refinement strategy used in our work is based heuristically on the drop in SSE, which is different than the data-dependent knot selection strategy used by Kohler. Hence, our approach does not necessarily satisfy the conditions for strong universal consistency given by Kohler, but we believe it is a reasonable heuristic for

refinement in practice as it is a common approach used for solving the knot-selection problem in the 1D setting. Proving consistency of our approach would be an interesting direction for future work.

6.3 Background

We now provide necessary background on Hierarchical B-splines and Truncated Hierarchical B-splines. We give a detailed description of the construction of these bases and their properties. We also provide some examples to illustrate the differences between these bases and the traditional tensor-product B-spline basis.

6.3.1 Hierarchical B-Splines

Tensor product B-splines provide a flexible function representation in dimensions greater than one. As stated previously, a major drawback presents itself when considering the addition of knot points for added flexibility. The problem is the possibility of adding more control points than are necessary when extra control is needed within some locality on the surface. A solution to this issue came with Hierarchical B-splines (HB-Splines), developed by Forsey and Bartels [100] in 1988, which constructs a spline basis through a hierarchical refinement of B-splines.

First, consider an initial tensor-product B-spline basis, $\mathcal{B}^0$, induced by knot vectors, U_i^0, where i gives an index for dimension d (corresponding to the standard basis of $\mathbb{R}^d$). The B-spline basis induces an associated function space, $\mathcal{V}^0$, over domain Ω^0 whose element is given by the choice of a control grid, $\mathbf{P}^0$.

Now, consider a refinement of the knot vectors, U_i^1, over domain, $\Omega^1 \subset \Omega^0$, which induces a new function space, $\mathcal{V}^1 \supset \mathcal{V}^0$ over Ω^1, and is controlled by $\mathbf{P}^1$. This refinement process is then repeated for N levels in this hierarchy to construct the hierarchical B-spline basis resulting in hierarchical domains $\Omega^N \subset \cdots \subset \Omega^1 \subset \Omega^0$, B-spline bases $\{\mathcal{B}^i\}_{i=1}^N$, and induced

function spaces $\mathcal{V}^N \supset \ldots \mathcal{V}^1 \supset \mathcal{V}^0$. As an example, see the hierarchical domain constructed in Figure 6.2. The Hierarchical B-spline basis is constructed recursively by the following.

1. Initialization: $\mathcal{H}^0 = \{\beta \in \mathcal{B}^0\}$.

2. Recursive case: $\mathcal{H}^{\ell+1} = \mathcal{H}_A^{\ell+1} + \mathcal{H}_B^{\ell+1}$ for $\ell = 0, 1, \ldots, N-2$, where

$$\mathcal{H}_A^{\ell+1} = \{\beta \in \mathcal{H}^\ell : \text{supp}(\beta) \not\subset \Omega^{\ell+1}\}$$

   and

$$\mathcal{H}_B^{\ell+1} = \{\beta \in \mathcal{B}^{\ell+1} : \text{supp}(\beta) \subset \Omega^{\ell+1}\},$$

   where supp (β) is non-zero domain of basis function β (*i.e.* the support).

3. $\mathcal{H} = \mathcal{H}^{N-1}$.

Note that when we construct our basis set in this way, we lose the partition-of-unity property that B-splines possess as a sort of "double counting" occurs at the boundaries of the hierarchical domains. This issue has been addressed with the development of Truncated Hierarchical B-splines [111] by Giannelli *et al.* in 2012.

6.3.2 Truncated Hierarchical B-Splines

Truncated hierarchical B-splines (THB-Splines) are extensions of HB-Splines that possess the partition-of-unity property, which is achieved by the truncation of coarser level spline basis functions that overlap into higher-level domains. This truncation removes the precise amount of "double counting" of basis functions present in the hierarchical B-spline representation, and can algorithmically be achieved by removing parts of the basis function given in the two-scale relation when doubling the amount of knot points at each level of the hierarchy.

The ***two-scale relation*** refers to the property that B-spline basis functions may be represented as a linear combination of scaled and translated copies of itself. In particular, it

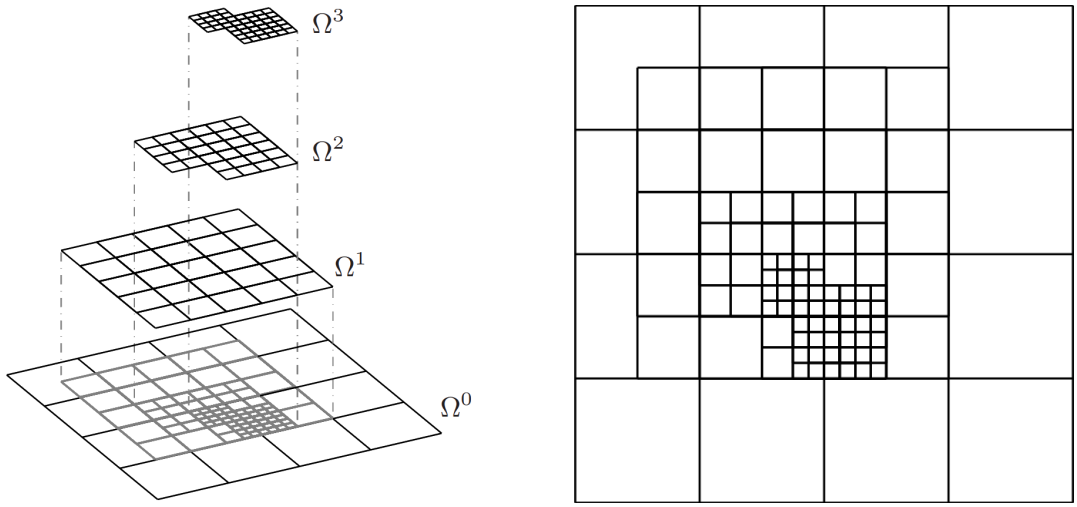


Figure 6.2: An example of a hierarchical domain constructed for a hierarchical B-spline.

states that:

$$N_k(t) = \sum_{i=0}^{m} p_i N_k(2t - i),$$

where

$$p_i = \frac{1}{2^{k-1}} \binom{k}{i}.$$

See Figure 6.3 for a visual depiction of this relation. This two-scale relation gives a relationship between refinements in the hierarchical B-spline representation. In particular, we write a basis function as a linear combination of scaled basis functions at a next level of refinement since the domain of each basis function is reduced by a factor of two in this refinement process.

Let $\tau \in \mathcal{V}^\ell$ be an element of the function space induced by B-spline bases at the ℓ level of the hierarchy, and let

$$\tau = \sum_{\beta \in \mathcal{B}^{\ell+1}} c_\beta^{\ell+1}(\tau)\beta, \quad c_\beta^{\ell+1} \in \mathbb{R}$$

be its representation with respect to the finer basis of $\mathcal{V}^{\ell+1}$ (which is achieved through the

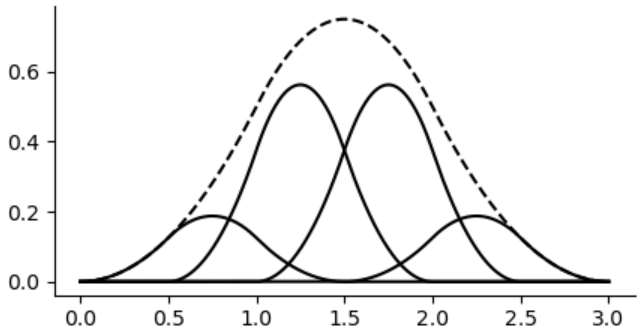


Figure 6.3: A visual depiction of the two-scale relation. Note that the dashed line, $N_k(t)$, is a linear combination of scaled, shifted, and translated copies of itself (*i.e.*, the sum of the solid lines).

two-scale relation). The ***truncation*** of τ with respect to $\mathcal{B}^{\ell+1}$ and $\Omega^{\ell+1}$ is defined as

$$trunc^{\ell+1}(\tau) = \sum_{\beta \in \mathcal{B}^{\ell+1}, supp(\beta) \not\subset \Omega^{\ell+1}} c_\beta^{\ell+1}(\tau)\beta.$$

where $supp(\beta)$ is the support (*i.e.* the portion of the domain where β is non-zero) of basis function β. That is, we view the truncation as the parts of the representation of the basis function defined in the two scale relation that do not overlap into the next finer level of the hierarchical domain. Applying this truncation to coarser levels of the HB-Spline, we construct the THB-Spline in a recursive manner as done with the HB-Spline. That is, the truncated hierarchical B-spline basis, $\mathcal{T}$, is recursively defined as follows:

1. Initialization: $\mathcal{T}^0 = \mathcal{H}^0$.

2. Recursive case: $\mathcal{T}^{\ell+1} = \mathcal{T}_A^{\ell+1} \cup \mathcal{T}_B^{\ell+1}$, for $\ell = 0, \ldots, N-2$ where

$$\mathcal{T}_A^{\ell+1} = \{trunc^{\ell+1}\tau : \tau \in \mathcal{T} \wedge supp\ \tau \not\subset \Omega^{\ell+1}\}$$

and $\mathcal{T}_B^{\ell+1} = \mathcal{H}_B^{\ell+1}$.

3. $\mathcal{T} = \mathcal{T}^{N-1}$.

At each recursion, we truncate the basis functions of the current set of basis functions with respect to the finer basis set. That is, any basis function with support in the finer domain has part or all of the basis function removed. We give a description of this process and compare it to the refinement process for HB-splines in Figure 6.4. Notice that the difference between the HB-spline and the THB-spline are in the basis functions that overlap between levels in the hierarchical domain. That is, the truncation operation removes the basis functions that are supported fully in the finer domain. Finally, note that there is a one-to-one correspondence between the hierarchical mesh and the THB-spline basis once refinement rules are in place. That is, each cell in the hierarchical mesh corresponds to a collection of basis functions in the THB-spline basis. Hence, the refinement of the hierarchical mesh according to a quad-tree partition of the domain induces a refinement of the THB-spline basis according to the same quad-tree partition.

6.4 THB-Splines for Nonparametric Regression

Now that we have given a detailed description of the construction of hierarchical and truncated hierarchical B-splines, we are ready to describe our approach for building THB-spline bases for non-parametric regression estimation. At a high level, our strategy is to subdivide the domain of the d-dimensional surface into hierarchical domains according to a d-dimensional hyperoctree with 2^d equal volume orthants (or hyperoctants) split at each cell. Hyperoctrees are a particular generalization of binary trees to higher dimensions and are commonly used in computer science for various applications, such as data storage and searching algorithms (see *e.g.*, [98, 160, 311]). For example, $k = 1$ results in a binary tree, $k = 2$ results in a quad-tree, $k = 3$ results in an oct-tree, and so on.

We will describe a quad-tree in depth as it corresponds to the 2D setting of our persistence

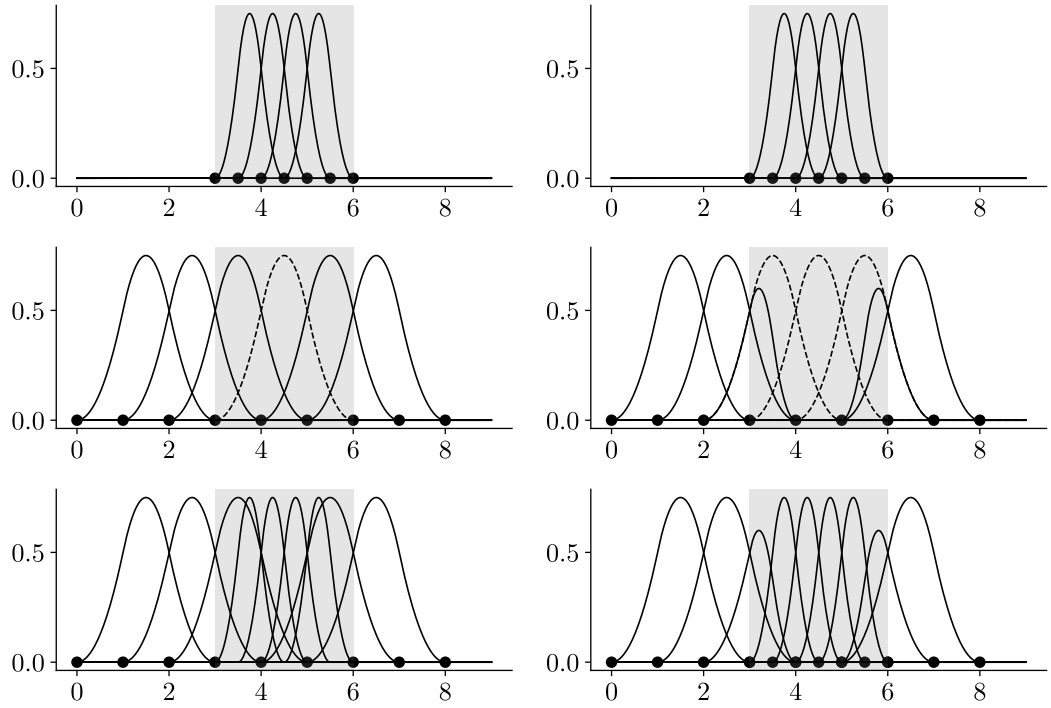


Figure 6.4: HB and THB spline bases constructed by refining the interval $[3, 6]$. The left side gives the HB-spline construction and the right side gives THB-spline construction with the refinement region highlighted in gray. The top row gives the basis set from the higher (finer) level in the hierarchy. The second row gives the lower (coarse) level basis set with the dotted lines representing the portions of the basis set removed to form the HB or THB basis set for the given level and the third row gives the combined basis set.

intensity functions, but the same approach can be applied to trees for higher dimensions. A quad-tree is an ordered tree with data attached to each node and where each node has either zero or four children. They are constructed by starting with a root node and recursively dividing each node into four according to a splitting rule. Quad-trees are used to generate subdivisions of planar spatial regions, where a local quadrant corresponds to a node in the quad-tree. Each time a node is divided, a quadrant is divided into four equal volume subquadrants. See Figure 6.5 for an example of a quad-tree that splits up to 3 levels of the hierarchy. In the context of our work, we do not necessarily want to split until exactly one data point falls as is often done when using quad-trees for data storage and searching

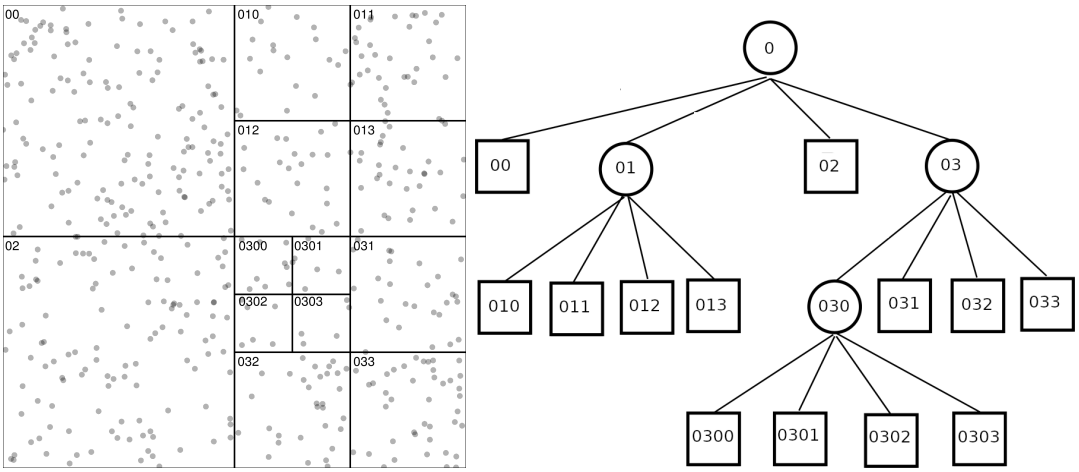


Figure 6.5: Quad-tree built over domain $[0, 1]^2$ with 500 data points uniformly distributed over the domain.

algorithms. Instead, we need to have enough data within each cell to ensure we are able to fit the global THB-spline model.

At a high level, the quad-tree gives us a hierarchical partition of the domain of the surface, and this hierarchical partition then induces a THB-spline basis. Thus, we construct the quad-tree by recursively splitting the domain according to a splitting rule and stop once no cells satisfy the splitting rule. We then use this quad-tree to construct the THB-spline basis used for regression estimation. This is summarized in Algorithm 6.10. In this algorithm, we build the quad-tree by first filtering out the leaves that do not satisfy a split predicate. We then greedily select the leaf that has the largest value for a split criteria to split in the next iteration of the algorithm. This process continues until no leaves satisfy the split predicate. Note that quad-tree data structures are typically used in representing THB-spline hierarchical meshes as they allow for efficient construction and evaluation of THB-spline bases [160]. Thus, their use in our approach is natural for building the THB-spline basis as there are often constraints on the hierarchical mesh for THB-splines used in practice because of the implemented algorithms. Our implementation relies on the G+Smo c++ library [153] used for isogeometric analysis which carries these constraints.

Algorithm 6.10 `buildTHBBases`($\mathcal{D}$, Ω, $\mathcal{S}$,)

Input:

$\mathcal{D}$ Data
Ω Domain of the surface
$\mathcal{S}$ Splitting predicates

Output:

$\mathcal{T}$ THB-spline Basis

1: $Q \leftarrow initializeHyperOctTree(\Omega)$
2: $\ell \leftarrow getRoot(Q)$
3: **while** $\ell \neq \emptyset$ **do**
4: $\quad \mathcal{L} \leftarrow getLeaves(\mathrm{Q})$
5: $\quad \mathcal{L} \leftarrow filter(\mathcal{L}, \lambda(l) : \texttt{splittableP}(l, \mathcal{D}, \mathcal{S}))$
6: $\quad$ **if** $\mathcal{L} = \emptyset$ **then**
7: $\quad\quad \ell \leftarrow \emptyset$
8: $\quad$ **else**
9: $\quad\quad \ell \leftarrow \arg\max_{l} [\texttt{leafScore}(l, \mathcal{D}, \mathcal{S})]$
10: $\quad Q \leftarrow splitCell(Q, \ell)$
11: $\mathcal{T} \leftarrow thbBases(Q)$
12: **return** $\mathcal{T}$

We can see from this general approach that the choice of splitting rule and splitting criteria can lead to different quad-trees and hence different THB-spline bases. For example, if we use a splitting rule based on sample size, then we will end up with a quad-tree that has more splits in regions of the domain where there are more data points. This will lead to a THB-spline basis that has more control points in these regions, which may allow for better estimation of the regression function in these regions. On the other hand, if we use a splitting rule based on curvature, then we will end up with a quad-tree that has more splits in regions of the domain where there is more curvature in the regression function. This will lead to a THB-spline basis that has more control points in these regions, which would make sense as it coincides with knot insertion techniques used in the 1D setting where knot insertion occurs where estimates of curvature are largest (*e.g.* [7, 59]). Our approach to

Algorithm 6.11 `leafScore`(ℓ, Q, $\mathcal{D}$, $\mathcal{S}$)

Input:

Q HyperOctree

$\mathcal{D}$ Data

ℓ Leaf considered for splitting

$\mathcal{S}$ Splitting parameters

Output:

s Score associated with leaf ℓ for splitting

`scores` $\leftarrow \emptyset$
$\mathcal{B}_0 \leftarrow$ tensorProductBSplineBasis$(\mathcal{D}, \ell, \mathcal{S}_{minKnots})$
$LS_0 \leftarrow$ leastSquaresFit$(\mathcal{B}_0, \mathcal{D})$
$SSE_0 \leftarrow$ `SSE`(LS_0)
for $nKnots$ in $\mathcal{S}_{minKnots} + 1, \ldots, \mathcal{S}_{nLookForward}$ **do**
    $\mathcal{B} \leftarrow$ tensorProductBSplineBasis$(\mathcal{D}, \ell, nKnots)$
    $LS \leftarrow$ leastSquaresFit$(\mathcal{B}, \mathcal{D})$
    `scores`$\leftarrow$ `scores` $\cup$ (`SSE`$_0$ $-$ `SSE`(LS))
$s \leftarrow max($`scores`$)$
return s

---

splitting is based on a combination of these two ideas, where we use a splitting rule based on a drop in sums of squared errors (SSE) to determine where to split the quad-tree. In particular, we split a cell according to the largest drop in SSE from fitting a local model to a small tensor-product B-spline basis to increasingly larger tensor-product B-spline bases, in a "look-forward" manner. This allows us to split the quad-tree in regions where there is more curvature in the regression function, while also having enough data to make such determination. This leaf scoring procedure is described in Algorithm 6.11. In summary, we locally subset our data to the cell of interest and then fit local tensor-product B-spline models with an increasingly greater number of parameters to the data. We then take the maximum drop in SSE from fitting these local models as our score for splitting the cell.

Finally, we may also wish to consider only a subset of the leaves for splitting at any given iteration of the algorithm for various reasons. For example, we may want to prioritize

Algorithm 6.12 `splittableP`(Q, $\mathcal{D}$, $\mathcal{S}$)

Input:
Q 2^k-tree
$\mathcal{D}$ Data
$\mathcal{S}$ Splitting parameters
ℓ leaf to test for splittability
Output:
p Whether or not the leaf is splittable

1: $p \leftarrow true$
2: $p \leftarrow p \wedge$ `isLeafAtMaxDepthRange`$(\ell, Q, \mathcal{S}_{maxDR})$
3: $p \leftarrow p \wedge$ `treeAtMaxParms`$(Q, \mathcal{S}_{maxP})$
4: $p \leftarrow p \wedge$ `isGramMatrixNonSingular`(ℓ, Q)
5: **return** p

splitting leaves that have more data points in them, or leaves that have larger areas. This can allow for certain stopping criteria on leaves to be met at lower depths (to prevent overfitting) and can also lead to more efficient search (through heuristic-based criteria) of the space of possible THB-spline parameterizations. We consider three different predicates for leaf splitting. The first is if the leaf is a distance greater than some threshold from the coarsest level of the hierarchical domain. The second is if the tree has more than some prespecified number of parameters and the last is if splitting the leaf would result in a singular design matrix for the global model fit. These predicates determine the split predicate given in Algorithm 6.12. Essentially, the first predicate allows for a tree to be built in a more balanced fashion, the second predicate bounds the number of parameters to prevent overfitting (tuned via cross-validation), and the third predicate allows us to maintain viable models at each iteration of the tree building process.

Many variations of the general approach described in Algorithm 6.10 are possible. However, we have found that a simple split based off of a drop in SSE from fitting local models with increasingly more parameters is a reasonable heuristic for testing if more flexibility

is needed in a given region of the domain. To determine the final model, we can tune the parameters of the splitting rule (*i.e.* max depth range and max number of parameters, etc.) by minimizing cross-validated MSE and using grid or Bayesian search on the tuning space.

6.5 Simulation Experiments

To illustrate the benefits of using THB-Spline bases with our local forward selection procedure, we carry out analyses over collections of simulated data in 1D and 2D. In our experiments, we compare the performance of our approach to the performance of a tensor-product B-spline approach using averaged mean squared-error as a function of the number of parameters in the model. In the 1D case, we use the same benchmarks considered in Chapter 4, namely `bigspike`, `blocks`, `bumps`, `heavisin`, and `mdoppler`, but modify them to have the range of the function to be between 0 and 1. In particular, with each of these benchmarks, we simulate data according to the model

$$y_i = (f(x_i) - min(f))/range(f) + \epsilon_i, \quad \epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma), \quad i = 1, \ldots, n$$

where f is the benchmark function, x_i are uniformly sampled points over the domain $[0, 1]$, and $range(f)$ and $min(f)$ are the range and minimum of the function f respectively estimated over the finite domain considered for each function. For the 1D case, we sample $n = 100,000$ points from this model and then fit our THB-spline over an increasing number of parameters, $p \in \{10, 20, 30, \ldots, 400\}$ with $\sigma = 0.1$, and also fit a tensor-product B-spline with the same number of parameters for comparison. We choose very large values for the parameter controlling maximum depth range so that it does not serve as a stopping criteria for building the quad-tree and instead only use the parameter controlling the maximum number of parameters in the model. We then compute the MSE for each fitted model. We conduct this experiment for each of the benchmark functions and each number of parameters, p, and

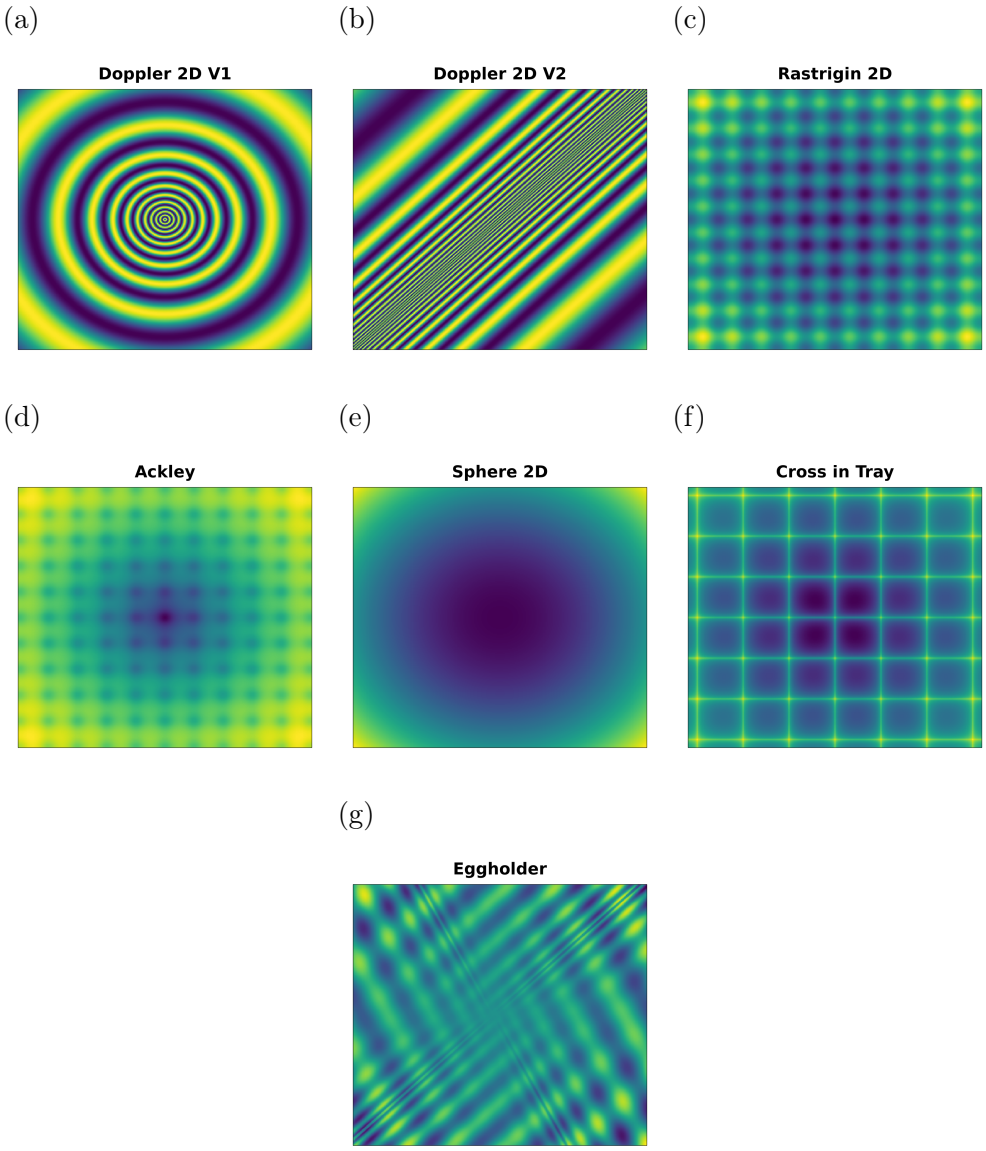


Figure 6.6: Heatmaps of the 2D benchmark functions used in our experiments.

average the MSE across $B = 30$ experimental runs.

We take the same approach for the 2D case, but instead we use some common 2D optimization benchmark functions [178, 215] as our regression functions. We also consider two 2D versions of the `mdoppler` function, one that is rotated around the origin and another that is reflected across the $y = x$ line. The second version (reflected across $y = x$ line) of the

`mdoppler` function serves as an extreme case where much of the local true curvature is along the diagonal of the domain, which should be a case where the THB-spline approach should have a clear advantage over the tensor-product B-spline approach, because the tensor-product B-spline approach will have to go to full refinement to capture the local curvature along the diagonal, while the THB-spline approach can adaptively refine along the diagonal as needed. Heatmaps of these functions are provided in Figure 6.6. We simulate data according to the same model as in the 1D case, but instead use a larger sample size of 1M points and consider increasing numbers of parameters $p \in \{100, 200, 300, \ldots, 4000\}$ and $\sigma = 0.05$. We again compare the MSE of our approach to the MSE of a tensor-product B-spline with the same number of parameters.

6.6 Simulation Results

Results from our 1D and 2D simulation experiments are given in Figures 6.7 and 6.8 respectively. In these figures, we plot the MSE of our approach (with dashed line) and the MSE of the tensor-product B-spline approach (with solid line) as a function of the number of parameters in the model for each of the benchmark functions considered. We find that our approach uniformly outperforms the tensor-product B-spline approach across all of the 1D and most of the 2D benchmark functions (all except the `sphere` and `eggholder` benchmark, in panels (e) and (g)). In particular, we find that our approach generally achieves a lower MSE than the tensor-product B-spline approach for the same number of parameters in the model, which suggests that our approach is able to use the parameters in the model to capture the underlying structure of the regression function more efficiently. We also find that the advantage of our approach over the tensor-product B-spline approach is more pronounced for functions with more (as a proportion of the domain) local curvature, such as the `mdoppler` functions, which is consistent with our intuition that the THB-spline approach should be able to capture local curvature in the regression function better compared to the tensor-product

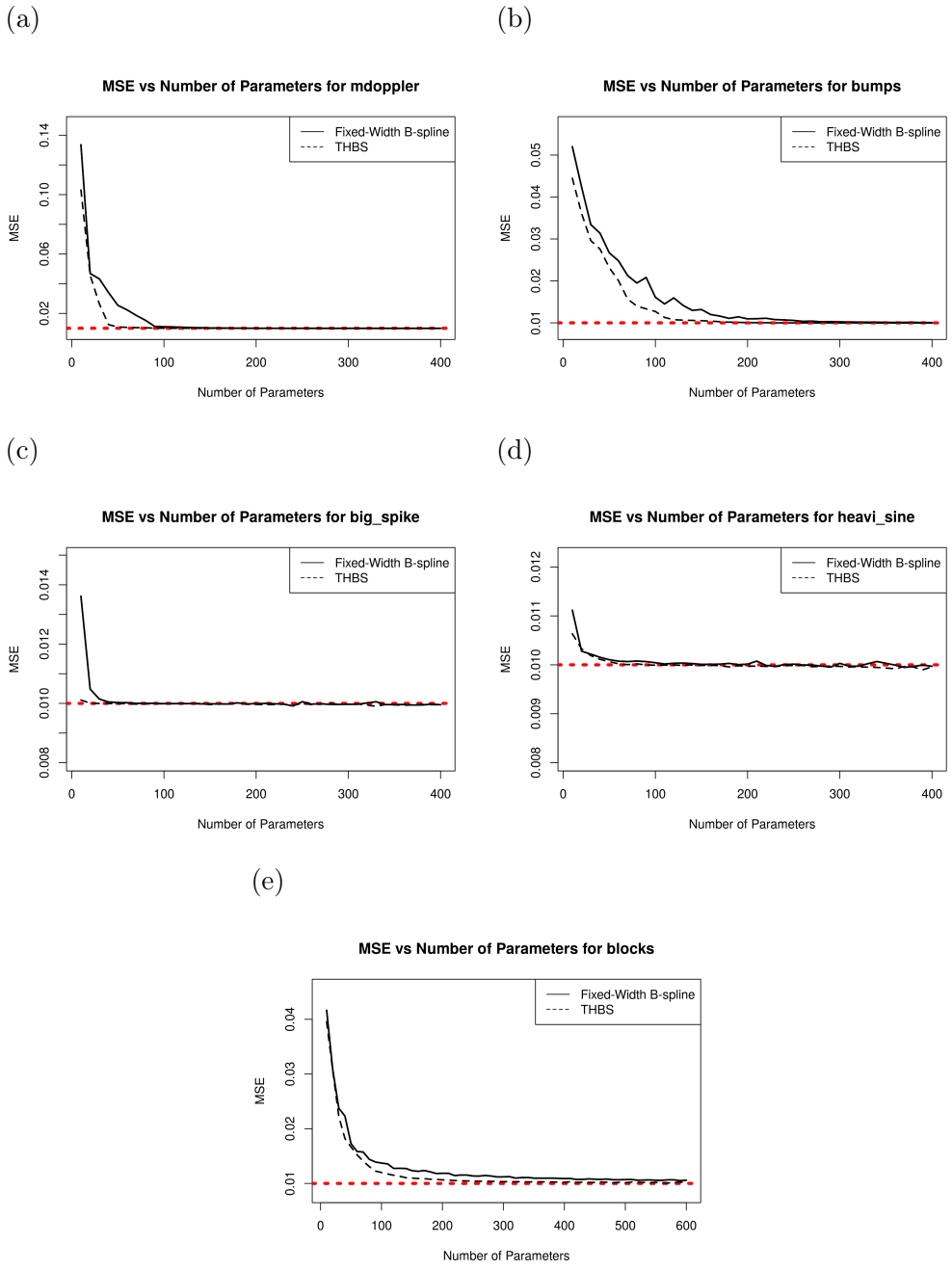


Figure 6.7: Results of our approach on the 1D Doppler function. The results show that the THB-Spline approach is uniformly more accurate to the tensor-product B-spline approach as a function of the number of parameters. The red line corresponds to the true MSE of the generating process ($\sigma_\epsilon = 0.1^2$).

B-spline approach, as we are essentially solving the knot-selection problem with our local forward selection procedure.

To illustrate why we see differences in performance between our approach and the tensor-product B-spline approach, we provide visualizations of the THB-spline basis and the fitted model in 1D (using $p = 150$ parameters) and 2D (using $p = 5000$ parameters). In 1D, we provide visualizations of the true function, the THB-spline basis, and the fitted model for each of the benchmark functions considered in our experiments. These visualizations are given in Table 6.1. We see that, for each benchmark, the THB-spline basis set has clear structure and with it we are able to adapt to the local curvature of the regression function, which allows for more efficient use of the parameters in the model to capture the underlying structure of the regression function.

In 2D, we provide visualizations of the true function, the hierarchical mesh and the THB-spline basis induced by this hierarchical mesh, and the fitted model for each of the benchmark functions considered in our experiments. These visualizations are given in Table 6.2. We see that, for functions that need it, the hierarchical mesh and THB-spline basis have clear structure and are able to adapt to the local curvature of the regression function, which allows for more efficient use of the parameters in the model to capture the underlying structure of the regression function. Alternatively, for functions that have roughly constant curvature across the function domain, the hierarchical mesh and THB-spline basis are more uniform across the domain, and more random in structure as there is not much local curvature to adapt to. We also see that the fitted model is able to capture the underlying structure of the regression function, which is consistent with the lower MSE achieved by our approach compared to the tensor-product B-spline approach.

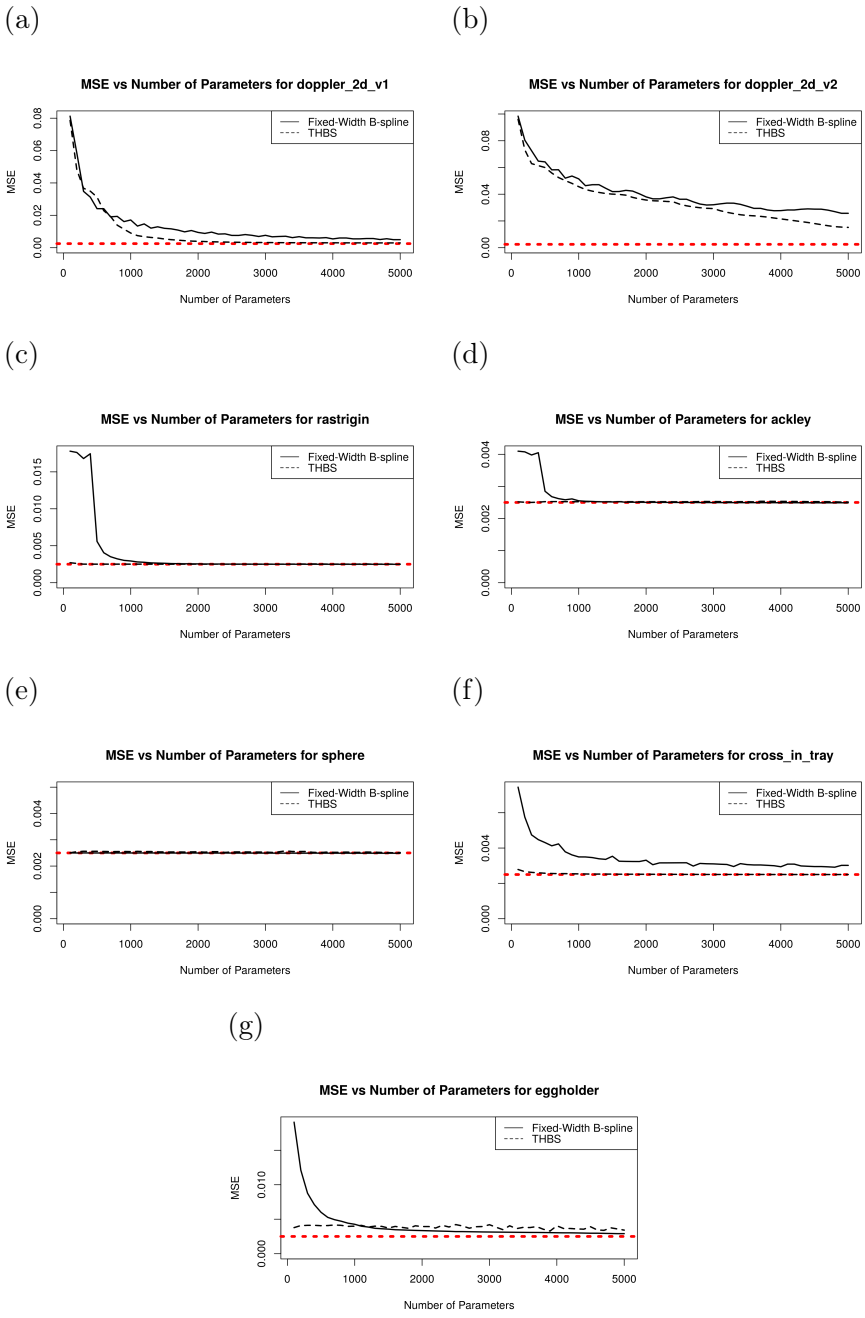


Figure 6.8: Experiment results for the 2D benchmark functions. Results plot MSE as a function of the number of parameters in the model for our THB-spline approach and the tensor-product B-spline approach. The red line corresponds to the true MSE of the generating process ($\sigma_\epsilon = 0.05^2$).

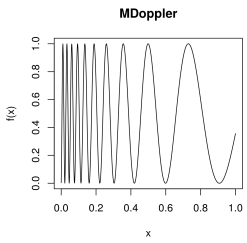
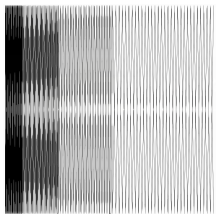
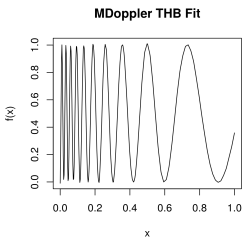
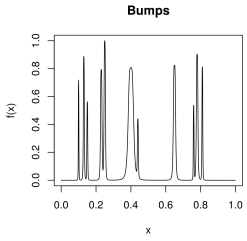
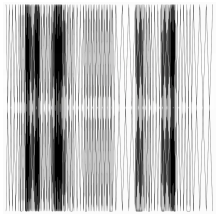
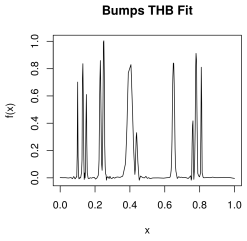
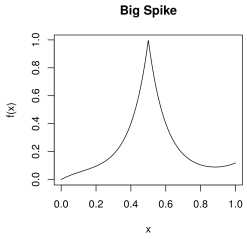
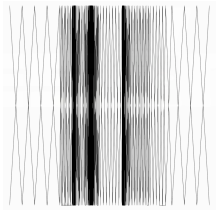
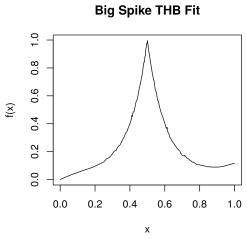
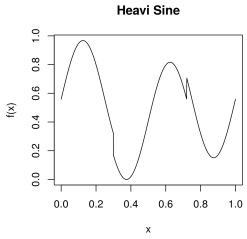
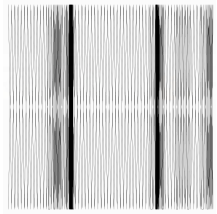
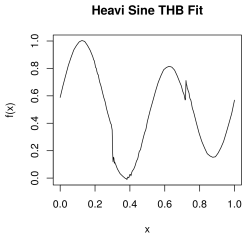
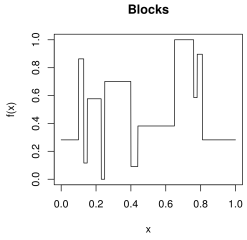
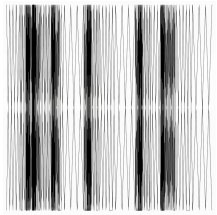
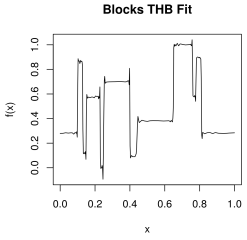
Function	True Function	THB-Spline Basis	Estimated Fit
`mdoppler`			
`bumps`			
`bigspike`			
`heavisine`			
`blocks`			

Table 6.1: Examples of the true function, THB-spline basis, and estimated fits using 150 basis functions for the 1D benchmark functions using our forward selection procedure.

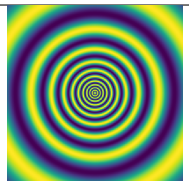
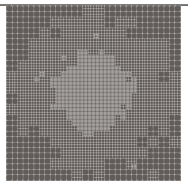
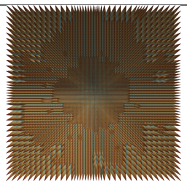
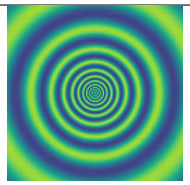
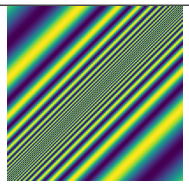
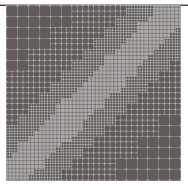
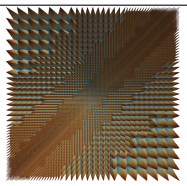
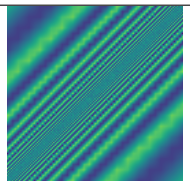
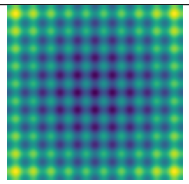
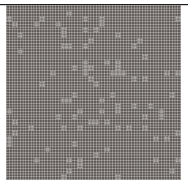
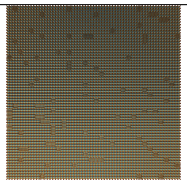
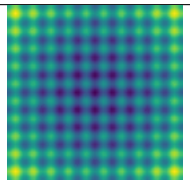
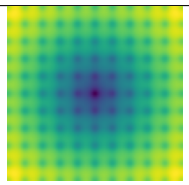
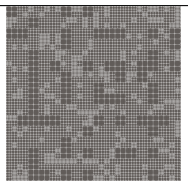
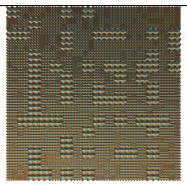
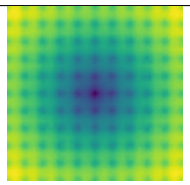
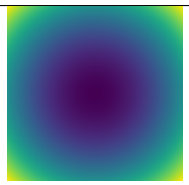
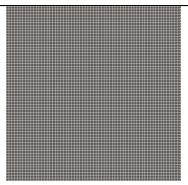
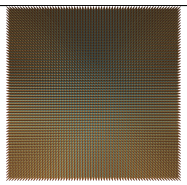
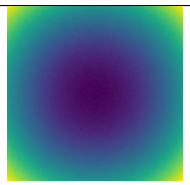
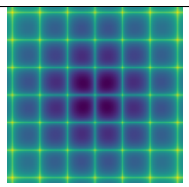
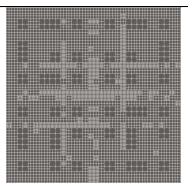
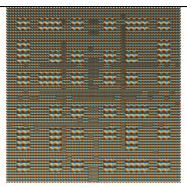
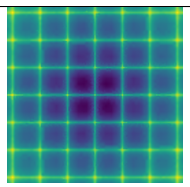
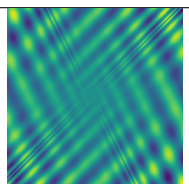
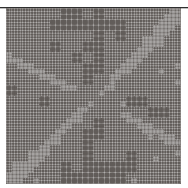
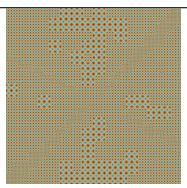
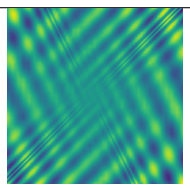
Function	True Function	THB-Mesh	THB-Spline Basis	Estimated Fit
Doppler_v1				
Doppler_v2				
Rastrigin				
Ackley				
Sphere				
Cross_in_Tray				
Eggholder				

Table 6.2: Examples of the true function, THB-Mesh, THB-spline basis, and estimated fits using 5000 basis functions for a subset of the 2D benchmark functions.

6.7 Discussion and Future Work

In this chapter, we have described our approach for building THB-spline bases for non-parametric regression estimation using a greedy forward step-wise selection procedure based on local model fits and a drop in SSE. We have developed this approach in the context of non-parametric regression estimation, with the purpose of estimating 2D persistence intensity functions when curvature in the intensity function varies spatially across the domain. Essentially, this allows us to solve the knot-selection problem in the 2D (and higher degree) setting, which is an important problem in spline-based regression estimation. By using a local forward selection procedure to build a THB-spline basis, we are able to construct THB-spline bases that are adaptive to the local curvature of the regression function, which can ultimately lead to more efficient use of the parameters in spline-based models.

We have also provided results from simulation experiments that illustrate the benefits of using THB-spline bases with our local forward selection procedure compared to using a standard tensor-product B-spline approach. We find that our approach generally outperforms the tensor-product B-spline approach across all the benchmark functions considered, except in some cases where the regression function is relatively smooth and local curvature does not vary over the domain. We also find that the advantage of our approach over the tensor-product B-spline approach is more pronounced for functions with more local curvature, which is consistent with our intuition that the THB-spline approach should be able to more efficiently capture local curvature in the regression function compared to the tensor-product B-spline approach. While we expect that the results presented in our simulation experiments should hold in the generalized linear model w/ spline components setting as well, our current experiments focus on the standard non-parametric regression setting. As such, future simulations should include the generalized linear model setting as well, specifically for our problem of estimating persistence intensity functions (*i.e.*, assuming an inhomogeneous Poisson process model for

the persistence diagram data and using a fine-pixel or Berman-Turner approximation of the likelihood for estimation with log-linear models).

Generally, our results are promising and there are several avenues for future work. One direction for future work is to explore the use of other types of hierarchical spline bases, such as hierarchical T-splines or hierarchical LR-splines, for non-parametric regression estimation. These types of hierarchical spline bases have different constraints when constructing the hierarchical mesh and different recursive splitting rules. Using these kinds of spline bases could allow for more flexible splitting rules that allow for splitting along a single axis, which could allow for even more efficient use of the parameters in the model to capture the underlying structure of the regression function.

Another direction for future work is to explore the use of different splitting rules and splitting criteria for building the quad-tree and THB-spline basis. For example, we could explore the use of splitting rules based on other measures of local curvature, such as estimates of the second derivative of the regression function or estimates of mean Gaussian curvature. We could also investigate splitting rules based on other model selection criteria, such as Akaike information criterion (AIC) or Bayesian information criterion (BIC).

We also believe that our approach can be used for creating an adaptive version of penalized splines (P-splines). P-splines are a type of spline-based regression model that uses a large number of basis functions and a penalty term to control the smoothness of the fitted model [88]. They are commonly used for non-parametric regression estimation and have been shown to perform well in a variety of settings. However, one potential limitation of P-splines is that they use a fixed-width set of basis functions, which consequently means that they are not adaptive. By using our approach to build a THB-spline basis, we could create an adaptive version of P-splines that allows for more efficient use of the parameters in the model to capture the underlying structure of the regression function. This could potentially lead to improved performance compared to traditional P-splines, especially for regression functions

with varying local curvature and for regression functions in higher dimensions.

We believe this can be achieved by decomposing the higher dimensional P-spline penalty into a sum of penalties over the different levels of the hierarchical mesh used to build the THB-spline basis and scaling the penalty by a factor of two[1] at each level to ensure the penalty is on the same scale across different levels of the hierarchical mesh. We believe this could greatly improve the performance of our approach as smoothing would help account for suboptimal splits in the quad-tree building process and could carry performance benefits. Additionally, by using a penalty to control the smoothness of the fitted model, we could potentially allow for deeper trees and more parameters in the model without overfitting, which could lead to improved performance compared to using a stopping rule based on the number of parameters in the model. We believe this would an interesting direction for future work with hierarchical data adaptive spline based methods such as the one we have developed in this chapter.

We suspect that our approach also leads to universal function approximation, in the sense that it can approximate any continuous function on a compact domain to arbitrary precision given enough data and enough parameters in the model. This is because THB-splines are a generalization of HB-splines, which are also known to be universal function approximators (given a particular hierarchical mesh) [162]. Developing a formal proof of this would also be an interesting direction for future work, as it would provide theoretical guarantees for the performance of our approach and could also potentially lead to insights into the types of functions that our approach are particularly well-suited for approximating.

Finally, we use a greedy forward step-wise selection procedure to build the THB-spline basis, which is a simple and computationally efficient approach for building the basis. However, it is possible that other approaches for building the THB-spline basis, such as a backward elimination (*i.e.* build and prune) procedure or a more global optimization approach, could

[1] because of the doubling of the number of basis functions at each level of the hierarchical mesh

lead to improved performance compared to the greedy forward step-wise selection. One interesting direction for future work would be to adapt the factored evolutionary algorithm (FEA) we used in the previous chapter to search the space of possible THB-spline bases.

CHAPTER SEVEN

CLASSIFICATION

In previous chapters, we developed methods for improving the estimation of persistence intensity functions (PIFs) and methods for conducting hypothesis tests over collections of PIFs. In this chapter, we focus on the use of estimated PIFs for our problem of classification of prostate cancer histopathology image data. We present and compare two approaches to classify prostate cancer histopathology image data using estimates of persistence intensity functions (PIFs). This is an extension of prior work, where we used PIFs to predict prostate cancer aggressiveness on a whole slide image (WSI) basis [172]. In the first approach, we use k-nearest neighbor (kNN) classification using L_1 distance between persistence images (*i.e.*, the evaluation of the PIF over a finite grid), and in the second approach, we use a random forest classifier with features coming from a vectorized persistence image. This comparison demonstrates the value of persistence images as a feature representation for machine learning approaches, and provides practical details and comparisons of approaches for using persistence images in machine learning approaches to classify histopathology image data.

7.1 Introduction

The Gleason grading system is a powerful prognostic predictor of patient outcomes in prostate cancer, and the most powerful predictor in routine clinical use. However, Gleason grading requires the subjective evaluation of architectural patterns present in prostate histopathology images. As a consequence, Gleason scores exhibit high inter-observer variability [90]. Efforts to increase the reproducibility of Gleason grading has led to the development of computational approaches, especially deep learning approaches, to enable the automatic grading of WSIs. In this work, we present an approach to grade ROIs using

features derived from the field of topological data analysis (TDA). TDA provides us with an alternative approach to image classification compared to the typical approach of convolutional neural networks (CNNs).

With the rise of digital pathology, enabled by the increase in the digitization of whole slide histopathology images, computer vision approaches, and in particular, deep learning approaches, are becoming the standard for building computer aided diagnostic (CAD) models to identify and grade cancer automatically in histopathology images [9, 48, 163, 179, 181, 182, 210, 214]. Deep learning models, including CNNs, require training on large image datasets, often consisting of millions of images. These images need to be annotated manually by expert pathologists prior to training, a task that is both time-consuming and cost prohibitive. While these deep learning models may represent the state of art in digital pathology, curation of large, expertly annotated datasets serves as a major impediment to their development and adoption. In addition, adoption of deep learning approaches is impeded by their "black box" nature, which prevents an intuitive mapping of the features learned by the model to features in the histopathology image [187].

The overall goal of this work is to demonstrate the value of features derived from persistent homology for predicting cancer grade on a region of interest (ROI) basis, and to compare methods for distance-based and vector-based classification using features derived from persistence images for the automatic prediction of prostate cancer images. In particular, we compare a k-nearest neighbor (kNN) approach using L_1 distance between persistence images (*i.e.*, the integrated absolute difference between surfaces) and a random forest approach using features obtained from persistence images. We chose a random forest approach because they perform well with high dimensional data and provide variable importance measures and feature selection methods that can be used to identify which features are most important for classification. This in turn allows us to create useful visualizations of the most important features used for classification.

For each method, we combine features from three separate filtrations constructed on the same image. The first is a cubical filtration, where the function is the pixel intensity of the grayscale image. The second is a height filtration, where the function is the height of the pixel in the image. The third is a Rips filtration on the point cloud generated by the centroids of the nuclei in the image. We conduct an ablation study to determine the contribution of each filtration to the overall performance of the classifier for both the kNN and random forest approaches.

7.2 Related Work

Automatic prostate cancer grading is an area of intense interest in digital pathology [9, 122, 141, 216, 251]. Automatic grading can be classified broadly into two approaches, grading with handcrafted features and grading with more automated feature engineering approaches, such as deep learning approaches that learn features from the image data directly [187]. Handcrafted features are those that correspond to a measurable feature in the histopathology image, such as gland size, shape, and orientation, and offer more interpretability than other approaches, as they often correspond to features pathologists are already trained to identify. Classifiers trained on histomorphometric features, like nuclei and gland shape, are often impeded by the amount of effort necessary by pathologists to hand demarcate features of interest in order to effectively train the segmentation approach. Approaches applying handcrafted features leverage simple classifiers, such as support vector machines (SVMs), trained on textural and morphometric features including one such work that achieved 77% predictive accuracy classifying Gleason 3 vs Gleason 4 images [82].

Conversely, deep learning approaches utilize large histopathology annotated datasets (often millions of ROIs) to learn features by learning a mapping from the image data to the class label without the need for handcrafting features, by incorporating operations such as convolution and pooling. These methods perform well but have been criticized in

the pathology community for lacking interpretability, serving as a "black-box" where it is difficult to understand learned features, or map them back to traditional histopathological features. Deep learning approaches, in particular CNNs, have shown excellent performance in automated grading of prostate cancer histopathology images [9, 122, 149, 213]. Some approaches seek to combine the explanatory power of handcrafted features with predictive power of CNN approaches. One such methodology leverages a hybrid approach involving identifying handcrafted features (shearlet transforms), and applying CNNs to generate learned features from the handcrafted features [251]. This hybrid approach involved training a CNN with both the shearlet transformed images and original RGB images, and outperformed an SVM using shearlet transforms alone in classifying Gleason grade 2 to 5 images (88 % predictive accuracy).

TDA can be viewed as a hybrid approach, a middle ground between traditional handcrafted feature approaches that require an *a priori* understanding to compute morphometric features, and deep learning approaches that learn representations of features over extremely large training datasets. Prior work demonstrated the power of TDA, and persistent homology specifically, to capture prostate cancer architecure both within and between Gleason grades [173]. TDA has a long history in describing the shape of geometric objects and is progressing in the area of automation of shape analysis. The analysis of handcrafted features in the context of shape analysis often means calculating certain characteristics of a given shape (*e.g.*, volume or maximum width), which can be a difficult computation problem, sometimes requiring laborious hand-annotation. In the case of prostate cancer images, this might correspond to the volume of individual prostate nuclei, cells, and glands. Calculating this, however, becomes a more cumbersome problem as cancer cells progress, as the delineation of individual prostate micro-architecture becomes more difficult. Turner *et al.* [287] developed a topological summary descriptor for capturing the shape of objects. Crawford *et al.* [65] extended their method to map to functional inner-product space

that is more conducive to statistical analysis. Crawford *et al.* used their method to capture the topological features of *glioblastoma multiforme* tumors for use in predicting disease-free and overall survival. Niclau *et al.* [218] used TDA in the discovery of a new type of breast cancer associated with longer survival.

7.3 Background

In this chapter, we use tools from topological data analysis (TDA) to construct a classifier for Gleason score prediction (3 versus 4) using sampled ROI level grayscale images. TDA provides a means of measuring features that are invariant under continuous deformation of an underlying space. We use techniques from TDA to capture architectural information contained in the ROI level images and then use these derived features in machine learning techniques to classify cancer grade. In particular, we represent ROI level images as a persistence image and, using this representation, then construct both a kNN and a random forest classifier.

One tool of TDA is persistent homology. Persistent homology is a way we can encode topological features changing in resolution as a multiset, which can be visualized either by a barcode or diagram [87, 317]. These can then be summarized in a variety of ways that are more useful for constructing statistical and machine learning models. For instance, persistence landscapes developed by Bubenik [44], persistence images described by Adams *et al.* [3], and Smooth Euler Characteristic Transform (SECT) described by Crawford *et al.* [65] are all ways to summarize persistence in a way that results in a vectorized representation that is more amenable to machine learning approaches. The methods chosen for this work are with persistence diagrams as summarized by persistence images. We refer to Chapter 2 for background on persistent homology, filtrations using cubical, height, and Rips filtrations, persistence diagrams, persistence intensity functions, and persistence images. Here, we give the required background on kNN and random forest classifiers.

7.3.1 K-Nearest Neighbor Classification

k-Nearest Neighbors (kNN) classification [61, 99] is a non-parametric, instance-based learning algorithm widely used for classification tasks. The fundamental idea behind kNN is to predict the class of a given data point based on the classes of its nearest neighbors in the feature space. The distance metric is critical in kNN, with the most common being the Euclidean distance. When a new data point needs to be classified, the kNN algorithm calculates the distances from this point to all other training data points and selects the k points that are closest, where k is a user-defined parameter. Let $\mathcal{D} = \{(\mathbf{X}_1, y_i), (\mathbf{X}_2, y_i), \ldots, (\mathbf{X}_n, y)\}$ be a dataset, where $\mathbf{X}_i$ is the feature vector of the i^{th} training instance and $y_i \in \{1, \ldots, L\}$ is its corresponding class label. Assuming distance metric, $d(\mathbf{X}, \mathbf{X}')$, the classification of the new point is determined by a simple majority vote among the k nearest neighbors. If $C_i(\mathbf{X}^{\text{new}})$ is the class of the i^{th} neighbor to point $\mathbf{X}^{\text{new}}$, the predicted class C for new point $\mathbf{X}^{\text{new}}$ can be expressed as:

$$C(\mathbf{X}^{\text{new}}) = \arg\max_{C_i} \sum_{j=1}^{k} I(C_j(\mathbf{X}^{\text{new}}) = C_i(\mathbf{X}^{\text{new}})),$$

where $I(\cdot)$ is the indicator function that equals 1 when the class labels are equal to one-another and 0 otherwise. The choice of k affects the algorithm's performance and the optimal choice is data-dependent and is often determined through methods such as cross-validation [13, 206, 279], rules-of-thumb [104, 183], or methods based on Bayesian strength [109, 110]. Generally, small values of k can lead to a model that is sensitive to noise, while large values can smooth out class boundaries and potentially lead to underfitting.

kNN operates under the assumption that similar instances are located close to each other in the feature space, making it well-suited for cases where cluster distributions exist. However, the algorithm requires careful consideration regarding the scaling of features and the choice of distance metric, as these can greatly affect the results, especially in high dimensional

spaces [6]. Additionally, kNN is computationally intensive as it stores all the training data and requires calculating distances during prediction, making it less practical for large datasets without effective optimizations. This computational issue can be mitigated by using data structures like KD-trees [28] or Ball trees [221] to speed up the nearest neighbor search and/or by using approximate nearest neighbor algorithms [10] that can provide faster results at the cost of some accuracy.

7.3.2 Random Forests

The random forest, developed by Breiman [38], is an extension of bagging [37] (Bootstrap AGGregating) classification and regression trees [39] that performs well with little tuning of parameters required [129]. In the case of binary trees, classification and regression trees predict a response by traversing down a decision tree, T, where each split into child nodes, t_L and t_R, corresponds to a split on a covariate in the dataset. In a quantitative response (regression) setting, predicted responses for a set of covariates is the mean value of the root node corresponding to that set of covariates in regression. Alternatively, in a classification setting, the probability of class membership is estimated by the proportion of each class in the root node. Trees are built in a greedy fashion, where the optimal split, s^*, is determined at each node, t, of the tree by maximizing the decrease of node impurity (*e.g.*, Gini impurity, Shannon entropy, etc.).

Bagging is a general ensemble procedure in which predictors are generated for a large number of bootstrap subsamples of some dataset, and then predictions are averaged across the ensemble to produce a predictor with less variability [129]. In the case of classification, the predicted class is the one that receives the largest number of plurality votes from the ensemble of classifiers. That is, for an ensemble of tree classifiers, $\{T_b(\mathbf{x})\}_{b=1}^{B}$, where each tree, T_b, in the ensemble is built on a bootstrap subsample of the larger dataset, the bagging

estimate of the predicted response would be

$$\hat{\mathbf{y}} = majority\ vote\left\{\{T_b(\mathbf{x})\}_{b=1}^{B}\right\}.$$

Random forests improve on bagged classification and regression trees by taking $m \leq p$ random subsamples of the p predictors when building the trees in the ensemble. This has the effect of reducing the correlation among the trees built using the bootstrap samples. For instance, if a very strong predictor exists in the training set, we might expect that the first split will always use this variable, inducing a correlation in the trees in the ensemble. The number of random predictors to use at each split, m, is an important tuning parameter for random forests (usually tuned via cross validation or out-of-bag errors).

7.4 Methods

To predict the grade of prostate cancer histopathology ROI images, we use features derived from persistence images, which are the vectorized representation of persistence diagrams. We construct two types of classifiers, a kNN classifier using L_1 distance between persistence images, and a random forest classifier using features derived from persistence images. For both classifiers, we use features derived from three separate filtrations constructed on the same image: a cubical filtration, a height filtration, and a Rips filtration on the point cloud generated by the centroids of the nuclei in the image. We conduct an ablation study to determine the contribution of each filtration to the overall performance of the classifier for both the kNN and random forest approaches.

7.4.1 Image Acquisition

Whole slide image (WSIs) were obtained from the Prostate cANcer graDe Assessment (PANDA) challenge dataset [47], which contains a dataset (from the Radboud University

Medical Center) of WSIs containing prostate glands that are individually labeled. With this dataset, we construct a collection of "pure-grade" 512×512 pixel ROIs of Gleason Grade 3 and Gleason Grade 4 by thresholding the proportion of pixels in the ROI that are labeled as prostate glands of grade 3 or grade 4. The chosen threshold of 0.5 was chosen to maximize the number of ROIs in the dataset while also ensuring that the ROIs were predominantly of a single grade. Then, to simplify tuning and training of the machine learning models, a large subset of these ROIs was taken to result in a balanced number of grade 3 and grade 4 ROIs, ultimately resulting in 3800 ROIs for each grade spread over a total of 3069 WSIs. Finally, each ROI was converted to an 8-bit grayscale image by taking the average of the RGB color channels.

7.4.2 Persistent Homology Derived Features

Each grayscale ROI was subject to three separate filtrations, a cubical filtration, a height filtration, and a Rips filtration on the point cloud generated by the centroids of the nuclei in the image. A cubical complex representation was constructed from the grayscale representation of an ROI, where the function value at each pixel corresponds to the pixel intensity. A lower-star filtration, $\{K_\tau^{\text{cubical}}\}_{\tau=0}^{255}$, was then constructed from the 8-bit grayscale ROI using a cubical complex representation, and then the associated persistence diagram was computed. This was done using the Python API from `GUDHI` [190]. This procedure resulted in a collection of H_0 and H_1 diagrams defined over the support $\{x, y \in \mathbb{Z}_{255} : x \leq y\}$ (*i.e.*, the discretized upper-diagonal of $[0, 255]^2$).

For the height filtration, the grayscale image was triangulated and a lower-star filtration, $\{K_\tau^{\text{height}}\}_{\tau=0}^{255}$, was constructed on the triangulated image using the height of the pixel as the function values and then the associated persistence diagram was computed. This was done using the R package `TDA` [95] with interface to `Dionysus` [204] for computing diagrams. This procedure also resulted in a collection of H_0 and H_1 diagrams, also defined over the support

$\{x, y \in \mathbb{Z}_{255} : x \leq y\}$.

Finally, for the Rips filtration, nuclei were segmented on the color version of the ROI using `StarDist` [262, 299, 300], a CNN-based semantic segmentation approach to nuclei segmentation.[1] Once the nuclei were segmented, the centroids of the segmented nuclei were identified and a Vietoris-Rips filtration was constructed on the resulting point cloud, $\{K_\tau^{\text{rips}}\}_{\tau=0}^{512}$. Persistence diagrams were then computed again using the `TDA` R package. This procedure resulted in a collection of H_0 and H_1 diagrams defined over the support $\{x, y \in \mathbb{Z}_{512} : x \leq y\}$ (*i.e.*, the upper-diagonal of $[0, 512]^2$). However, the connected components (*i.e.*, H_0 diagrams), only vary in death coordinates, and so this results in a collection of H_0 diagrams defined over death space of $[0, 512]$ and a collection of H_1 diagrams defined over the upper-diagonal plane of $[0, 512]^2$.

PIFs were then estimated for each diagram using kernel density estimation and bandwidth parameters tuned via cross-validation on the training set for each classifier. To minimize edge-effect bias in the estimation of the PIFs, an edge correction was applied to the intensity estimate, by dividing intensity by the convolution of the Gaussian kernel with the window of observation, an approach originally described by Diggle [76]. This was carried out using the `spatstat` R package [18].

Finally, PIs were constructed by evaluating the PIFs over (the upper-diagonal of) a grid of points in the birth-death plane. We used a grid of 50×50, resulting in $50 \times 50/2 = 1250$ evaluation points for each 2D diagram. For the 1D diagram, we used a grid of 500 evaluation points. Thus, the full feature set consists of $1250 \times 5 + 500 = 6750$ features for each ROI, which is the combined vectorized representation of the persistence diagrams for the three filtrations.

[1] We used the author's associated Python library, located at https://github.com/stardist/stardist

7.4.3 Model Comparisons

We conduct an ablation study to determine the contribution of each filtration to the overall performance of the classifier for both the kNN and random forest approaches. For each approach, we train classifiers using features derived from each filtration separately, and then a classifier using all features combined. In particular, we consider models with H_0 and H_1 PIs separately and combined for each filtration, and then each pair of combined filtrations, and finally a model with all three filtrations combined.

For the full random forest model with all combined features, we also consider a model with feature selection. We then train a final random forest classifier on the selected features. In particular, prior to training the random forest classifier, we apply an all-relevant feature selection algorithm, called `Boruta`. All-relevant feature selection is a type of feature selection that aims to identify all features which are relevant to the outcome be selected, rather than just a minimal optimal subset of features. Instead of asking, "Which features are the best minimal set for prediction?", an all-relevant feature-selection algorithm asks "Which features carry information that is genuinely better than random noise?". Thus, we can expect that the features selected by an all-relevant feature selection algorithm will improve the performance of the classifier by removing features that are not relevant to the outcome, while retaining all features that are relevant to the outcome.

We carry this out using the algorithm from the `Boruta` R package [165], to the full set of features derived from all three filtrations. `Boruta` compares the importance of original attributes with that of randomly permuted copies (called "shadow features"). For each original attribute, the algorithm creates a shadow attribute by randomly shuffling the values of the original attribute across objects. Then classification is performed using all attributes (original and shadow), and the importance of each attribute is computed. The algorithm iteratively removes features that are deemed less important than the best of the shadow features by testing the maximum Z score (computed by dividing the average loss by its

standard deviation) among shadow attributes against the Z score of each original attributes. The algorithm continues until it reaches a point where it cannot reject the null hypothesis that a feature is less important than the best shadow feature.

7.4.4 Model Tuning

Data were split into a training and validation set on a WSI basis, such that all ROIs from a given WSI were contained in either the training or validation set. This was done to avoid issues with pseudoreplication that can occur in the classifier. For example, if there is very high similarity of topological features within a WSI, then the classifier may be able to learn WSI specific features that do not generalize to other WSIs, and thus the performance of the classifier may be artificially inflated. To avoid this issue, we split the data on a WSI basis, according to an 80/20 split, resulting in 6093 ROIs nested within 2455 WSIs and 1507 ROIs nested within 614 WSIs in the training and validation sets respectively.

The kNN classifier was tuned using 5-fold cross validation (CV) on the training set. For the kNN classifier, the number of neighbors, k, was tuned over the set $\{1, 3, 5, 7, 9, 10, 20, 40, 50\}$ and the distance metric was fixed to be the L_1 distance between persistence images. The best performing model was selected based on classification accuracy on the validation set, and then the final model was retrained on the full training set using the selected parameters. Tuning revealed smaller values of k performed better than larger values of k, with the best performing model, as measured by mean classification accuracy on the validation set, having $k = 7$. Note that L_1 was chosen so that combining features would be additive in the distance metric across filtrations, which is not automatically the case for L_2 distance. Initial testing also indicated that L_1 distance performed better than L_2 distance for the kNN classifier.

For the random forest classifier, the number of trees was set to be 1000. Initial testing indicated that the number of predictors to use at each split, m, had little impact

on performance of the classifier, and was thus set to the rule-of-thumb value of $\sqrt{p}$ for the experiments, where p is the number of features.

7.5 Results

Results of the model comparisons for the kNN appoach are shown in Table 7.1, and results of the model comparisons for the random forest approach are shown in Table 7.2. The performance of the models was evaluated using area under the receiver operating characteristic curve (AUC), Youden's J statistic ($J = Sensitivity + Specificity - 1$), classification accuracy (proportion correct), and $F1$-score ($F1 = \frac{2 \times precision \times recall}{precision + recall}$), on the validation set.

The highest performing model from the kNN approach was the model using the combined cubical filtration features, with an AUC of 0.8938 and accuracy of 0.8135, while the highest performing model from the random forest approach was the model using all features combined with feature selection, with an AUC of 0.9364 and accuracy of 0.8647. In general, it appears for this dataset, that the random forest approach using features derived from persistence images performs better than the kNN approach using L_1 distance between persistence images, with each model using the same features performing better in the random forest approach compared to the kNN approach. This may be because the random forest approach is able to learn a more complex decision rule in the feature space, while the kNN approach is limited to a more local decision boundary. The random forest approach also allows for feature selection, which can help to improve performance by reducing noise and overfitting.

The highest performing models in both the kNN and random forest approaches were those using features derived from the cubical filtration, with the model using all features combined with feature selection performing best in the random forest approach, and the model using combined H_0 and H_1 features performing best in the kNN approach. The models using features derived from the height and Rips filtrations performed worse than the models using features derived from the cubical filtration, which may indicate that the cubical filtration

Model	AUC	Youden	Accuracy	F1
H_0 Cubical	0.8729	0.5972	0.7989	0.7914
H_1 Cubical	0.8570	0.5744	0.7876	0.7771
$H_0 \cup H_1$ Cubical	**0.8938**	**0.6265**	**0.8135**	**0.8068**
H_0 Height	0.6579	0.2498	0.6250	0.6164
H_1 Height	0.6606	0.2291	0.6144	0.6159
$H_0 \cup H_1$ Height	0.6762	0.2750	0.6376	0.6300
H_0 Rips	0.6471	0.2452	0.6230	0.6044
H_1 Rips	0.6659	0.2797	0.6396	0.6429
$H_0 \cup H_1$ Rips	0.6479	0.2453	0.6230	0.6066
Cubical/Height	0.8642	0.5747	0.7883	0.7666
Cubical/Rips	0.8435	0.5351	0.7684	0.7472
Rips/Height	0.6950	0.3287	0.6648	0.6466
Full	0.8548	0.5360	0.7690	0.7422

Table 7.1: Model characteristics for kNN models on the validation set. The best performing model is the combined $H_0 \cup H_1$ cubical model and performance appears to degrade when adding more features across filtrations.

captures more important architectural features of the prostate cancer ROIs, or image data more generally, than the height and Rips filtrations.

The random forest model using all features combined with feature selection performed better than the model using just H_0 cubical filtration features, with an AUC of 0.9364 compared to 0.9236, and an accuracy of 0.8647 compared to 0.8602, respectively. However, this is a relatively small increase in performance, which may indicate that the features derived from the height and Rips filtrations are capturing redundant features to those derived from

Model	AUC	Youden	Accuracy	F1
H_0 Cubical	**0.9236**	**0.7200**	**0.8602**	**0.8567**
H_1 Cubical	0.8965	0.6197	0.8094	0.8115
$H_0 \cup H_1$ Cubical	0.9277	0.7086	0.8543	0.8522
H_0 Height	0.7392	0.3965	0.6983	0.6947
H_1 Height	0.7155	0.3039	0.6514	0.6590
$H0 \cup H_1$ Height	0.7430	0.3967	0.6983	0.6963
H_0 Rips	0.7205	0.3402	0.6703	0.6622
H_1 Rips	0.7222	0.3607	0.6807	0.6689
$H_0 \cup H_1$ Rips	0.7263	0.3578	0.6794	0.6648
Cubical/Height	0.9359	0.7186	0.8595	0.8552
Cubical/Rips	0.9167	0.6991	0.8498	0.8456
Rips/Height	0.7425	0.3915	0.6957	0.6933
Full	0.9315	0.7171	0.8589	0.8540
Full w/ Selection	**0.9364**	**0.7290**	**0.8647**	**0.8607**

Table 7.2: Model characteristics for random forest models on the validation set. The best performing model is the model with all features combined with feature selection, which achieves an accuracy of 0.8647. However, this is only marginally better than the model using only the H_0 cubical filtration PI which achieves an accuracy of 0.8602.

the cubical filtration. The random forest model using all features combined without feature selection performed worse than the model using all features combined with feature selection, with an AUC of 0.9277 compared to 0.9364, and an accuracy of 0.8543 compared to 0.8647, which may indicate that the feature selection algorithm is effective at removing noisy features that do not contribute to the performance of the classifier.

We use a t-SNE plot (*i.e.*, the first two components of a t-distributed stochastic neighbor embedding) to visualize the separation of the classes in the best performing model from the kNN approach, which used combined features from the H_0 and H_1 cubical filtrations. We carried this out using the scikit-learn Python library [228] taking default perplexity parameter of 30. This is provided in Figure 7.1. Panel (a) provides a t-SNE plot of the true distribution of classes for the training set, while panel (b) provides a t-SNE plot of the predicted distribution of classes for the validation set. In both panels, the points are colored according to grade, with blue points corresponding to grade 3 and orange points corresponding to grade 4. In panel (a), we see that there is generally good separation of the classes in the t-SNE plot, with some overlap between the classes. This suggests that the features derived from the combined H_0 and H_1 cubical filtration are able to capture important architectural features of the prostate cancer ROIs that are relevant for classifying cancer grade, but that there may be some noise in the features that is preventing perfect separation of the classes. In panel (b), we see that the distribution of points for the predicted distribution of classes in the t-SNE plot is generally the same as the true distribution of classes in the t-SNE plot. This suggests that the kNN model is able to learn a decision boundary that separates the classes in the t-SNE plot.

For the random forest approach, we use a variable importance plot to visualize the importance of the features in the best performing model, which was the model using all features combined with feature selection. This is provided in Figure 7.2, where we see that the most important features in the model are those derived from the cubical filtration and features derived from loop components (H_1). This may indicate that the features derived from the cubical filtration are capturing more important architectural or textural features of the prostate cancer ROIs than the features derived from the height and Rips filtrations and that the loop components are better capturing important architectural features of the prostate cancer ROIs, such as the size and shape of the glands, than the connected components.

(a) (b)

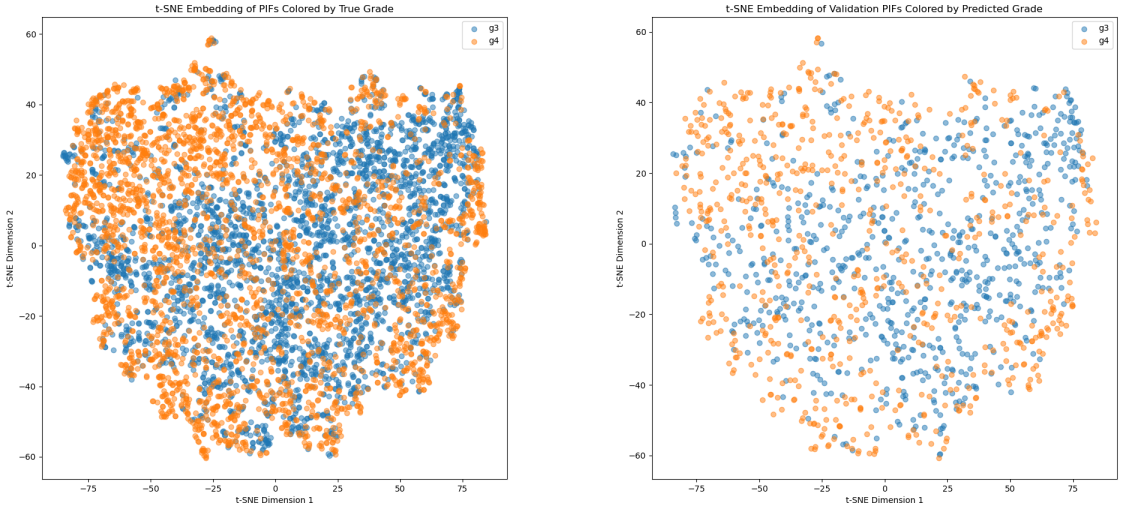


Figure 7.1: Stochastic neighbor embedding (t-SNE) over the full dataset with the true distribution of classes for the training set (a) and the predicted distribution of classes for the validation set (b) for the best performing kNN model, which was the combined H_0 and H_1 cubical model. In both panels, the points are colored according to grade, with blue points corresponding to grade 3 and orange points corresponding to grade 4.

Locations of the most important features in the birth-death plane for the cubical filtration are birth-death coordinates of about $(75, 75)$ in both the H_0 and H_1 diagrams and also birth-death coordinates of about $(150, 230)$ in the H_0 diagram. These coordinates correspond to intensity values in the grayscale image and thus can be used understand the types of topological features important for classifying prostate cancer grade. For instance, the important features in the H_0 diagram with birth-death coordinates of about $(150, 230)$ correspond to connected components that are born at an intensity value of about 150 and die at an intensity value of about 230, which likely corresponds to nuclei in the images, as those are the pixels with the highest intensity values in the grayscale images. Thus, this region of the birth-death plane is potentially approximating the number of nuclei in the image, which

is an important architectural feature of the prostate cancer ROIs and ought to be predictive when comparing Gleason grade 3 versus grade 4 as Gleason grade 4 is characterized by more crowded glands and thus more nuclei in the image.

Finally, we also include a binary mask of the selected features from the feature selection algorithm for the random forest model using all features combined with feature selection, which is provided in Figure 7.3. In these plots, we see that the feature selection algorithm selected features across all three filtrations, but that a large proportion of the selected features were those derived from the cubical filtration, which is consistent with the variable importance plot. Generally, the selected features correspond to features with higher variable importance. This plot provides a visual representation of the features in the filtrations that are important for classifying prostate cancer grade, giving a "topological signature", of sorts, of the topological features of the prostate cancer ROIs that are important for classifying cancer grade.

7.6 Discussion and Future Work

In this chapter, we have demonstrated the use of topological based features for predicting Gleason grade of ROIs in histopathology image data. Our ablation study revealed that features derived from the cubical filtration are most important for classifying cancer grade, compared to features derived from the height and rips filtrations.

Our study also revealed that the random forest approach using features derived from persistence images performs better than the kNN approach using L_1 distance between persistence images. It is possible that the random forest approach is retaining more information about the features in the persistence images (compared to computing a distance between persistence images) and is thus able to learn a more complex decision rule in the feature space. Finally, we provided visual representation of the important features in the birth-death plane, and feature-selection provides visualization of a topological signature for differentiating

Gleason grade 3 versus grade 4 prostate cancer ROIs.

Our top model (the random forest model using all features combined with feature selection) achieved an AUC of 0.9364 and an accuracy of 0.8647 for classifying Gleason grade 3 vs 4 in ROIs, which is promising for a model using only topological features, and suggests that topological features may be useful for classifying cancer grade in histopathology image data. The performance of the random forest models, indicate that topological features may offer an alternative or compatible set of features to the more commonly used CNN features for classifying prostate cancer. This points to the possibility of building multi-modal approaches that combine the advantages of CNNs and persistent homology derived features. A natural extension of this future work is the exploration of the extent to which CNNs are capturing similar features to persistent homology based features, or whether these features are unique.

This work focused on the binary classification of grade 3 vs grade 4 prostate cancer ROIs. However, the same approach could be extended to multi-class classification of Gleason grades 3, 4, 5, healthy tissue, stroma tissue, and other benign mimickers of prostate cancer, such as benign prostatic hyperplasia. This would enable a full image segmentation approach, where the goal is to classify each ROI in a WSI as belonging to one of the classes. While a more challenging task, it would be more clinically relevant, as it would allow for the identification of different grades of cancer within a single WSI, as well as the identification of benign tissue and other mimickers of cancer. Additionally, computation of Gleason grade (and ISUP grade) for the entire WSI, which is the current standard for grading prostate cancer in clinical practice, could be conducted.

This computer aided diagnostic approach could be used to assist pathologists in grading prostate cancer, and could potentially improve the accuracy and consistency of grading, as well as reduce the time and effort required for grading. In particular, when a patient has a biopsy with multiple cores, the model could segment each biopsy WSI and compute the proportion of tissue for each tissue type. This would allow for a more automated and

standardized approach to grading prostate cancer, which could be useful for improving the accuracy and consistency. It would also allow for the future possibility of more detailed and nuanced grading of prostate cancer as proportions of different tissue types (including subtypes of cancer) could be computed, which may be more informative for prognosis and treatment planning than the current approach of assigning a single grade to the entire WSI.

One important consideration for future work is the accurate estimation of uncertainty in the predictions. For instance, if the model is able to identify areas of the image that uncertain about Gleason grade, then this information could be useful for a pathologist to focus their attention on those areas of the image when grading the cancer. However, accurately estimating the uncertainty in the predictions is a challenging problem, especially when using complex models such as random forests or neural networks. Examples of approaches to estimating uncertainty in the predictions are to use bootstrapping or other resampling methods to estimate the variability in the predictions. For example, Mench and Hooker developed a general resampling approach based on U-statistics that allows for estimation of standard errors of predictions from machine learning models [195, 264] that could be used to estimate the uncertainty in the predictions from the random forest model.

Finally, we used a random forest classifier and a k-nearest neighbor classifier for our classification task, but there are many other types of classifiers that could be applied to this problem that better capture the structure of the functional data we are using as predictors. For example, there are variants of random forest classifiers that explicitly allow for functional covariates [197, 202, 237, 243]. Additionally, CNNs and other types of neural network architectures that are designed for functional data could also be applied to this problem and may better capture the structure of the functional data we are using as predictors. Using CNNs for this problem would also allow for the possibility of a multi-modal approach where features learned by the CNN are combined with features derived from persistence images to improve performance. Exploring the use of these other types of classifiers for this problem is

another potential area for future research.

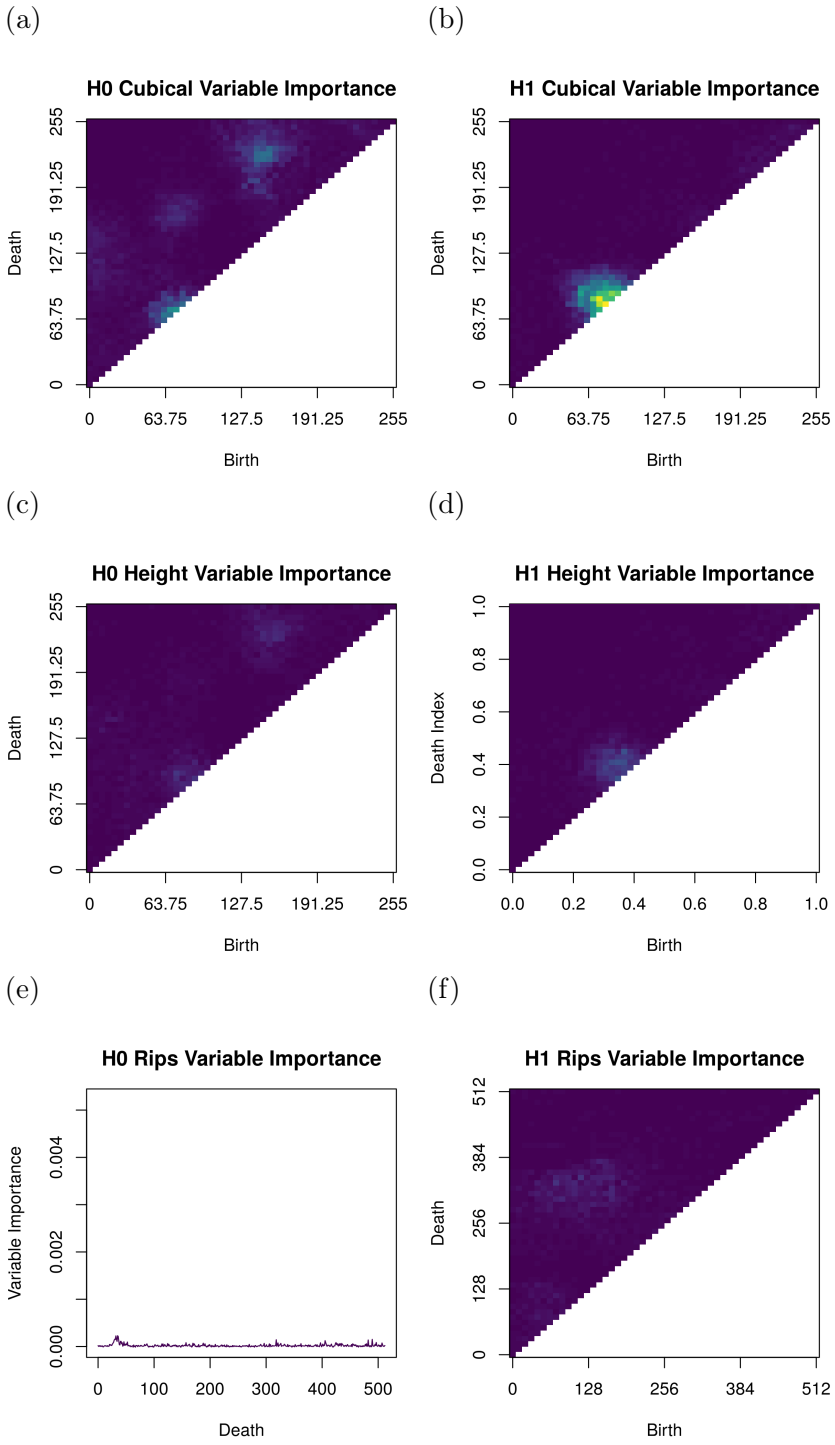


Figure 7.2: Variable importance for full random forest model with all features combined. Variable importance is on the same scale across all figures, with the most important features indicated by the high intensity regions in the variable importance maps.

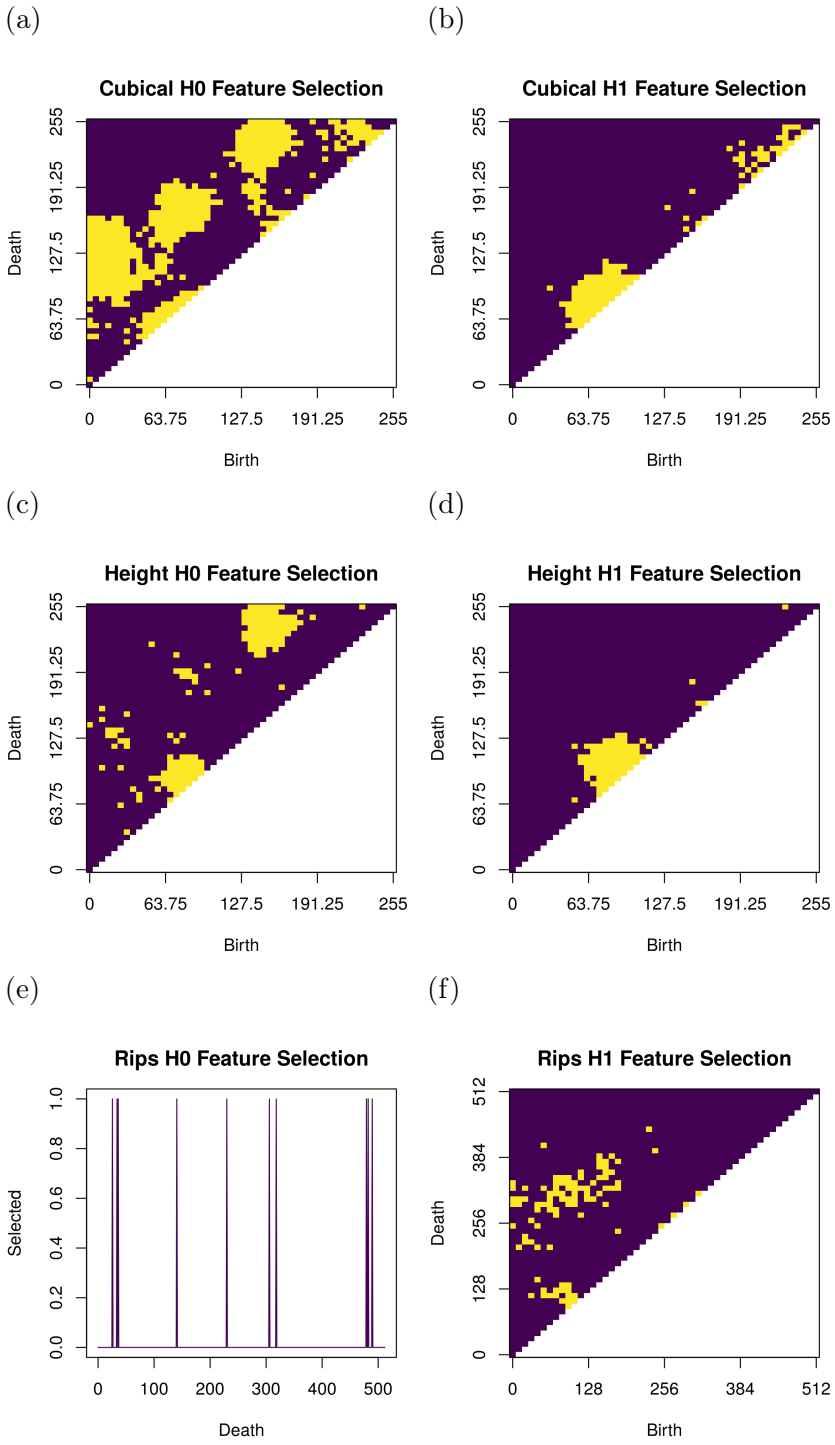


Figure 7.3: Boruta feature selection mask across all features for the combined model. The mask indicates which features were selected as important by the Boruta feature selection algorithm in yellow for the 2D diagrams and a value of 1 for the 1D diagram.

CHAPTER EIGHT

SUMMARY

In this chapter, we review the main contributions of this dissertation and discuss potential future research directions for the development of methodology for topological data analysis (TDA).

8.1 Summary of Contributions

In this dissertation, we have developed methods for TDA using point process and nonparametric regression techniques. The central theme of this dissertation is the development and improvement of methods for analyzing topological data using persistence intensity functions, an important summary measure of persistence diagrams. In particular, we have focused on the problems of estimating persistence intensity functions, conducting hypothesis testing with them, and using them as predictors to apply to classification of prostate cancer histopathology image data. In Chapter 2, we reviewed concepts from TDA, including simplicial complexes, homology, and persistent homology. We also defined persistence diagrams, persistence intensity functions and persistence images, which are important topological summaries in TDA. We introduced the problem of grading prostate cancer using histopathology image data, and developed a new generative point process model for simulating the location of nuclei in histopathology images.

Chapter 3 considers the use of a kernel-based mixed-curve model to estimate persistence intensity functions for hierarchically sampled persistence diagram data. Using this estimator, we developed a novel functional ANOVA hypothesis testing procedure for comparing persistence intensity functions across multiple groups, while accounting for the hierarchical structure of the data. We achieved this by expanding a permutation based local hypothesis

testing method, namely the Westfall-Young procedure, to construct the global test statistic as the minimum of the local adjusted test statistics, allowing for both local and global inferences across the domain of the persistence intensity function. We carried out simulation studies to evaluate the performance of this proposed method, and then applied it to prostate cancer histopathology data for identifying differences in the persistence intensity functions across Gleason grades. This is important work in the context of TDA, as it provides the first method for conducting hypothesis testing with persistence diagram data under a repeated measures design, which is common to encounter in practice.

Chapter 5 focuses on the problem of estimating a persistence intensity function using adaptive B-spline regression. Solving the knot-selection problem is a key step for adding flexibility to B-spline based methods, and is important for improving estimation. However, the optimization problem for knot selection is non-convex and is computationally expensive to solve. Thus, we adapted the factored evolutionary stochastic search algorithm for solving the knot-selection problem for B-spline regression, enabling more efficient estimation of persistence intensity functions using adaptive B-spline representations. We evaluated the performance of our proposed method through simulation studies, comparing it to traditional stochastic search methods for knot selection. Using a fixed number of fitness evaluations, our approach achieved the same or better estimation performance compared to traditional stochastic search methods and has the potential for further computational efficiency improvements.

In Chapter 6, we considered adaptive estimation of 2D surfaces using B-spline based methods, as persistence intensity functions are generally defined over a 2D domain. In the higher-dimensional case, B-spline based methods can be computationally expensive to use, and their representational efficiency can be poor. To address these issues, we investigated the use of hierarchical B-spline representations for estimating 2D surfaces. We developed a data-driven approach for selecting the hierarchical mesh for the B-spline representation, and tested the performance of this approach through simulation studies. We found that our

approach achieved better estimation performance than traditional tensor-product B-spline representation methods for estimating 2D surfaces, and that it has the potential for further representational and computational efficiency improvements. Chapter's 4 and 5 contribute to the development of methodology for TDA by providing new tools for estimating persistence intensity functions in an adaptive manner.

Finally, in Chapter 7, we applied persistence images to the problem of grading prostate cancer using histopathology data. We conducted an ablation study to compare the performance of various filtrations using k-nearest neighbor and random forest classifiers for predicting Gleason grade from image data. We found that persistence images can provide useful information for classification tasks in the context of prostate cancer grading and that cubical complex filtrations of histopathology data provided the most useful information for classification. We also found that random forest classifiers performed better than k-nearest neighbor classifiers for this task. This work contributes to TDA by demonstrating the potential of persistence images for analyzing complex data in practice and gives insights into the types of topological features that may be most useful for classification tasks in the context of prostate cancer grading. Overall, the work in this disseration has contributed to methodology for TDA using point process and nonparametric regression techniques and provides avenues for future research in this area.

8.2 Future Research Directions

There are several potential future research directions for the development of methodology for TDA using point process and nonparametric regression techniques. One main central theme of our work is in the reduction of assumptions when conducting analyses with persistence diagram data. For example, in Chapter 3, we developed a method for conducting hypothesis testing with persistence diagram data under a repeated measures design, which relaxes the assumption of independent and identically distributed data that is most often made in TDA.

This suggests that further relaxation of independence assumptions could be a fruitful area for future research. For example, one could investigate analysis methods under temporal or spatial dependence structures, which are also common in many applications of TDA.

While the methods we developed in Chapter 3 were for a repeated measures design, which is a specific type of dependence structure, there are many other types of dependence structures that could be considered. For example, one could consider spatial dependence structures for persistence diagram data arising from spatial data such as ROIs spatially indexed from WSIs, or temporal dependence structures for persistence diagram data arising from time series data. While the repeated measures design we considered can account for these types of dependence structures to some extent, they do not model the autocorrelation in a specific way compared those that are commonly used in spatial and temporal data analysis. We believe that the mixed-curve modeling approach paired with the local hypothesis testing method can be extended to these settings by estimating the appropriate local model that includes these spatial or temporal dependencies, and then applying similar Westfall-Young procedures for hypothesis testing.

Additionally, while our approach provided a method for conducting an ANOVA test for comparing persistence intensity functions across multiple groups while accounting for the hierarchical structure of the data, we did not develop methods for conducting *post hoc* pairwise comparisons among the groups. Thus, developing methods for conducting *post hoc* pairwise comparisons with persistence diagram data under a repeated measures design is another potential area for future research. We believe this can be achieved by applying the same local hypothesis testing method, but with the appropriate permutations for conducting pairwise comparisons, and then subsequently applying a Bonferroni correction to the minimum local adjusted test statistic to account for the multiple testing. In addition, hypothesis tests other than the ANOVA, such as a 1-sample test against a known baseline function, could be developed using the Romano-Wolf procedure, as its methods apply more generally to

bootstrap and resampling based methods. However, this would require further investigation of the properties of the Romano-Wolf procedure in the context of functional data, as analytical results for this procedure are currently unknown in this context.

In Chapters 5 and 6, we developed methods for estimating persistence intensity functions using adaptive B-spline regression. In general, these methods relax the assumption of spatial homogeneity of the persistence intensity function covariance structure, which is the standard assumption made when estimating persistence intensity functions. Additionally, in Chapter 6, we applied an edge-correction method when estimating persistence intensity functions to minimize bias near the boundaries of the domain. This is important because much of the mass of the persistence intensity function are typically found near the boundaries of the domain, and thus bias in these regions can have a large impact on estimation performance. Historically, this step has been avoided, by using weighting methods to down-weight the contribution of points near the boundaries. While it generally makes sense to estimate persistence intensity functions using well known methods to reduce estimation bias, it is unclear what effects these assumptions would have on downstream tasks such as classification. Thus, further investigation of the effects of these assumptions on downstream tasks is another potential area for future research.

Our focus in this dissertation has been on the persistence intensity function as a summary of persistence diagrams, but there are many other types of functional summaries that have been developed in TDA, such as persistence landscapes, persistence silhouettes, and smooth Euler characteristic curves. Generally, the hierarchical mixed-curve modelling approach and our local hypothesis testing procedure is agnostic to the type of functional summary being used, and thus could be applied to these other types of summaries. However, the properties of our methods in the context of these other types of summaries is not known, and thus further investigation of our estimation and hypothesis testing approach for repeated measures across these topological descriptors is another potential area for future research.

Another direction to explore is the use of these other types of topological summaries for conducting classification tasks. In Chapter 6, we compared different filtrations and different classifiers for classification of prostate cancer histopathology data. These other topological summaries may capture different aspects of the topological structure of the data and could potentially provide different information for classification tasks compared to the persistence images we considered in Chapter 6, and thus could be useful for improving classification performance. Comparisons of these different types of summaries over the same data such as the prostate cancer histopathology data we consider and other types of image data could provide insights into the types of topological features that are most useful for these kinds of classification tasks.

One potential direction is to investigate the use of other types of nonparametric regression methods for estimating persistence intensity functions, such as Gaussian process regression or neural network functional representations. These methods may provide different advantages and disadvantages compared to the kernel-based and B-spline based methods we consider. For example, one approach which may extend our kernel-based method is to use a Bayesian mixed-model locally instead of our current likelihood-based regression method. This may help deal with linear separation estimation issues that can arise when applying local linear regression to persistence intensity function estimation, as the Bayesian kernel-based methods allow for incorporation of prior information that can help regularize estimation in these cases. Additionally, taking a Bayesian approach such as this may allow for the propagation of uncertainty from the estimation step into downstream tasks such as classification.

Our analysis in Chapter 7 estimated probabilities of class membership for the different Gleason grades. We split our data into a training and test sets by splitting at the WSI level to avoid issues with data leakage arising from psuedoreplication of patches within the same WSI. However, the classifiers we used still made predictions at the patch level assuming independence of the patches, which is not the case for those data. Because of this, estimation

of probabilities of class membership for the different Gleason grades may be biased, and thus the performance of our classifiers may be suboptimal and standard error estimates of probabilities may be underestimated. Thus, further investigation of methods for classification that account for the hierarchical structure of the data is another potential area for future research. For example, one potential approach is to use mixed-effect models for classification, which can account for the hierarchical structure of the data. For example, extensions to random forest classifiers to the mixed-effect setting have been developed [126, 140, 229], and thus could be applied to our classification problem. Additionally, mixed-effect models using neural network functional representations have recently become an area of research in the machine learning community [188, 269, 284, 307]. These methods have the potential to provide flexible representations for classification tasks while also accounting for the hierarchical structure of the data. Some of these methods take a Bayesian approach, which may also allow for the propagation of uncertainty from the PIF estimation step into these down-stream tasks of classification.

We finish with a description of a more idealized estimation procedure for the classification task we considered in Chapter 7. Persistence intensity functions are intensity functions of the persistence diagram treating it as a point process. As a consequence, bias and estimation error in the estimation of persistence intensity functions themselves are important to consider. Thus, accounting for edge-effect bias, inhomogeneity in the covariance structure, and lack of independence in the data are important for improving estimation of persistence intensity functions. An ideal classification analysis would incorporate these considerations when estimating persistence intensity functions, and would consider the estimation of persistence intensity functions as an intermediate step in the classification pipeline, to then propagate this estimation error into the classification step. We believe a Bayesian approach could make this more feasible, as it would allow for hierarchical modelling of this process and thus the propagation of uncertainty throughout the process and into the classification step more

natural. Posterior predictive distributions for class membership could then be obtained that would account for uncertainty in the various steps of the process. Thus, further investigation of Bayesian approaches for classification using persistence intensity functions, we believe, is an important area to consider for future research.

Overall, there are many potential future research directions for the development of methodology for TDA using point process and nonparametric regression techniques. We have highlighted some potential directions, but there are many other avenues to explore in this area. We hope that our work in this dissertation will inspire further researchers to contribute to the development of new methods for analyzing topological data in practice.

Glossary

abelian group An **abelian group** is a group G in which the group operation is commutative. That is, for all $a, b \in G$, we have $a \cdot b = b \cdot a$. In an abelian group, the order of the elements does not affect the result of the operation.. 30, 212

Borel set A **Borel set** is any set in a topological space that can be formed from open sets (or, equivalently, from closed sets) through the operations of countable union, countable intersection, and relative complement.. 38

free abelian group A **free abelian group** is an abelian group that has a basis, which is a set of elements such that every element of the group can be uniquely expressed as a finite linear combination of these basis elements with integer coefficients. In other words, a free abelian group is isomorphic to a direct sum of copies of the integers $\mathbb{Z}$.. 30

group A **group** is a set G equipped with a binary operation $\cdot$ that satisfies the following axioms:

- **Closure:** For all $a, b \in G$, the result of the operation $a \cdot b$ is also in G.
- **Associativity:** For all $a, b, c \in G$, we have $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
- **Identity element:** There exists an element $e \in G$ such that for all $a \in G$, we have $e \cdot a = a \cdot e = a$.
- **Inverse element:** For each $a \in G$, there exists an element $a^{-1} \in G$ such that $a \cdot a^{-1} = a^{-1} \cdot a = e$.

. 30, 212

Hadamard product If $\mathbf{A}$ and $\mathbf{B}$ are two matrices of the same dimensions, then the **Hadamard product** of $\mathbf{A}$ and $\mathbf{B}$, denoted $\mathbf{A} \odot \mathbf{B}$ is defined by the element-wise

product

$$(\mathbf{A} \odot \mathbf{B})_{ij} = a_{ij} \cdot b_{ij},$$

resulting in a matrix $\mathbf{A} \odot \mathbf{B}$ of the same dimensions as $\mathbf{A}$ and $\mathbf{B}$. 18

homeomorphism A **homeomorphism** is a continuous function $f : X \to Y$ between two topological spaces X and Y that has a continuous inverse function $f^{-1} : Y \to X$. In other words, f is a bijection and both f and f^{-1} are continuous. If such a function exists, we say that X and Y are **homeomorphic**, meaning they are topologically equivalent.. 214

homomorphism A structure preserving map between two algebraic structures of the same type. That is, a map $f : A \to B$ between sets A and B such that $f(x \cdot y) = f(x) \cdot f(y)$ $\forall$ pairs $x, y \in A$. 30, 31

homotopic equivalence Two topological spaces X and Y are **homotopically equivalent** if there exist continuous maps $f : X \to Y$ and $g : Y \to X$ such that $g \circ f$ is homotopic to the identity map on X and $f \circ g$ is homotopic to the identity map on Y.. 29

Kronecker product If $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{B}$ is a $p \times q$ matrix, then the **Kronecker product** of $\mathbf{A}$ and $\mathbf{B}$, denoted $\mathbf{A} \otimes \mathbf{B}$ is defined by the block matrix

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \dots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{n1}\mathbf{B} & \dots & a_{nn}\mathbf{B} \end{bmatrix}$$

. 90

manifold A **manifold** is a topological space that locally resembles Euclidean space. More formally, an n-dimensional manifold is a topological space M such that for every point

$p \in M$, there exists an open neighborhood U of p and a homeomorphism $\phi : U \to V$, where V is an open subset of $\mathbb{R}^n$. This means that around every point, the manifold looks like $\mathbb{R}^n$, even though the global structure of the manifold may be more complex.. 29

subgroup A **subgroup** is a subset H of a group G that is itself a group under the operation of G. That is, H must satisfy the following conditions:

- **Closure:** For all $a, b \in H$, the result of the operation $a \cdot b$ is also in H.
- **Identity element:** The identity element e of G is also in H.
- **Inverse element:** For each $a \in H$, the inverse element a^{-1} is also in H.

. 31

submanifold A **submanifold** is a subset N of a manifold M that is itself a manifold, and the inclusion map $i : N \hookrightarrow M$ is an embedding. This means that N inherits the topological and differentiable structure from M, and locally around each point in N, it looks like a lower-dimensional Euclidean space.. 29

REFERENCES CITED

[1] K. Abramowicz, A. Pini, L. Schelin, S. Sjöstedt de Luna, A. Stamm, and S. Vantini. Domain Selection and Familywise Error Rate for Functional Data: A Unified Framework. *Biometrics*, 79(2):1119–1132, 2023.

[2] I. S. Abramson. On Bandwidth Variation in Kernel Estimates-A Square Root Law. *The Annals of Statistics*, 10(4):1217–1223, 1982.

[3] H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier. Persistence Images: A Stable Vector Representation of Persistent Homology. *Journal of Machine Learning Research*, 18(8):1–35, 2017.

[4] R. J. Adler and S. Agami. Modelling persistence diagrams with planar point processes, and revealing topology with bagplots. *Journal of Applied and Computational Topology*, 3(3):139–183, 2019.

[5] R. J. Adler, S. Agami, and P. Pranav. Modeling and Replicating Statistical Topology and Evidence for CMB Nonhomogeneity. *Proceedings of the National Academy of Sciences*, 114(45):11878–11883, 2017.

[6] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the Surprising Behavior of Distance Metrics in High Dimensional Space. In *International conference on database theory*, pages 420–434. Springer, 2001.

[7] E. Aguilar, H. Elizalde, D. Cárdenas, O. Probst, P. Marzocca, and R. A. Ramirez-Mendoza. An Adaptive Curvature-guided Approach for the Knot-placement Problem in Fitted Splines. *Journal of Computing and Information Science in Engineering*, 18(4):041013–1–8, 2018.

[8] M. J. Anderson. A New Method for Non-parametric Multivariate Analysis of Variance. *Austral Ecology*, 26(1):32–46, 2001.

[9] E. Arvaniti, K. S. Fricker, M. Moret, N. Rupp, T. Hermanns, C. Fankhauser, N. Wey, P. J. Wild, J. H. Rueschoff, and M. Claassen. Automated Gleason Grading of Prostate Cancer Tissue Microarrays via Deep Learning. *Scientific reports*, 8(1):12054, 2018.

[10] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An Optimal Algorithm for Approximate Nearest Neighbor Searching Fixed Dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998.

[11] A. Atak, K. Buchin, M. Hagedoorn, J. Heinrichs, K. Hogreve, G. Li, and P. Pawelczyk. Computing Maximum Polygonal Packings in Convex Polygons using Best-fit, Genetic Algorithms and ILPs (CG Challenge). In *40th International Symposium on Computational Geometry (SoCG 2024)*, pages 83–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024.

[12] E. Aufhauser and M. M. Fischer. Log-linear Modelling and Spatial Analysis. *Environment and Planning A*, 17(7):931–951, 1985.

[13] M. Azadkia. Optimal Choice of k for k-nearest Neighbor Regression. *arXiv preprint arXiv:1909.05495*, 2019.

[14] A. Baddeley, M. Berman, N. I. Fisher, A. Hardegen, R. K. Milne, D. Schuhmacher, R. Shah, and R. Turner. Spatial Logistic Regression and Change-of-Support in Poisson Point Processes. 2010.

[15] A. Baddeley, R. Moyeed, C. Howard, and A. Boyde. Analysis of a Three-dimensional Point Pattern with Replication. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 42(4):641–668, 1993.

[16] A. Baddeley, E. Rubak, and R. Turner. *Spatial Point Patterns: Methodology and Applications with R*, volume 1. CRC press Boca Raton, 2016.

[17] A. Baddeley and R. Turner. Practical Maximum Pseudolikelihood for Spatial Point Patterns: (with Discussion). *Australian & New Zealand Journal of Statistics*, 42(3):283–322, 2000.

[18] A. Baddeley and R. Turner. Spatstat: An R package for Analyzing Spatial Point Patterns. *Journal of Statistical Software*, 12:1–42, 2005.

[19] D. Bates, M. Mächler, B. Bolker, and S. Walker. Fitting Linear Mixed-Effects Models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015.

[20] D. Bates, M. Mächler, B. Bolker, and S. Walker. Fitting Linear Mixed-effects Models using lme4. *Journal of statistical software*, 67:1–48, 2015.

[21] M. Bedward, D. Eppstein, and P. Menzel. `packcircles`*: Circle Packing*, 2024. R package version 0.3.7.

[22] M. L. Bell. *The Use of Maximum Pseudolikelihood in Generalized Linear Mixed Models for the Analysis of Replicated Spatial Point Patterns*. University of Colorado Health Sciences Center, 2002.

[23] M. L. Bell and G. K. Grunwald. Mixed Models for the Analysis of Replicated Spatial Point Patterns. *Biostatistics*, 5(4):633–648, 2004.

[24] R. L. Belton, B. T. Fasy, R. Mertz, S. Micka, D. L. Millman, D. Salinas, A. Schenfisch, J. Schupbach, and L. Williams. Learning Simplicial Complexes from Persistence Diagrams. *arXiv preprint arXiv:1805.10716*, 2018.

[25] K. Benhenni, F. Ferraty, M. Rachdi, and P. Vieu. Local Smoothing Regression with Functional Data. *Computational Statistics*, 22(3):353–369, 2007.

[26] Y. Benjamini and Y. Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995.

[27] Y. Benjamini and D. Yekutieli. The Control of the False Discovery Rate in Multiple Testing Under Dependency. *Annals of Statistics*, pages 1165–1188, 2001.

[28] J. L. Bentley. Multidimensional Binary Search Trees used for Associative Searching. *Communications of the ACM*, 18(9):509–517, 1975.

[29] M. Berman and T. R. Turner. Approximating Point Process Likelihoods with GLIM. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 41(1):31–38, 1992.

[30] E. Berry, Y.-C. Chen, J. Cisewski-Kehe, and B. T. Fasy. Functional Summaries of Persistence Diagrams. *Journal of Applied and Computational Topology*, 4(2):211–262, 2020.

[31] J. Besag. Contribution to the Discussion on Dr. Ripley's Paper. *JR Stat Soc B*, 39:193–195, 1977.

[32] G. Biau, F. Bunea, and M. H. Wegkamp. Functional Classification in Hilbert Spaces. *IEEE Transactions on Information Theory*, 51(6):2163–2172, 2005.

[33] C. A. Biscio, N. Chenavier, C. Hirsch, and A. M. Svane. Testing goodness of fit for point processes via topological data analysis. *Electronic Journal of Statistics*, 14(1):1024–1074, 2020.

[34] A. J. Blumberg, I. Gal, M. A. Mandell, and M. Pancia. Robust Statistics, Hypothesis Testing, and Confidence Intervals for Persistent Homology on Metric Measure Spaces. *Foundations of Computational Mathematics*, 14(4):745–789, 2014.

[35] C. Bonferroni. Teoria Statistica Delle Classi e Calcolo Delle Probabilita. *Pubblicazioni del R istituto superiore di scienze economiche e commericiali di firenze*, 8:3–62, 1936.

[36] C. Bracco, C. Giannelli, and A. Sestini. Adaptive Scattered Data Fitting by Extension of Local Approximations to Hierarchical Splines. *Computer Aided Geometric Design*, 52:90–105, 2017.

[37] L. Breiman. Bagging Predictors. *Machine Learning*, 24(2):123–140, 1996.

[38] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[39] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and Regression Trees (The Wadsworth Statistics/Probability Series)*. Wadsworth Publishing, 1983.

[40] L. Breiman, W. Meisel, and E. Purcell. Variable Kernel Estimates of Multivariate Densities. *Technometrics*, 19(2):135–144, 1977.

[41] N. E. Breslow and D. G. Clayton. Approximate Inference in Generalized Linear Mixed Models. *Journal of the American Statistical Association*, 88(421):9–25, 1993.

[42] M. Brockmann, T. Gasser, and E. Herrmann. Locally Adaptive Bandwidth Choice for Kernel Regression Estimators. *Journal of the American Statistical Association*, 88(424):1302–1309, 1993.

[43] W. J. Browne. *Applying MCMC Methods to Multi-level Models*. PhD thesis, University of Bath Bath, 1998.

[44] P. Bubenik. Statistical Topological Data Analysis using Persistence Landscapes. *J. Mach. Learn. Res.*, 16(1):77–102, 2015.

[45] P. Bubenik and P. Dłotko. A Persistence Landscapes Toolbox for Topological Statistics. *Journal of Symbolic Computation*, 78:91–114, 2017.

[46] A. Buffa, G. Gantner, C. Giannelli, D. Praetorius, and R. Vázquez. Mathematical Foundations of Adaptive Isogeometric Analysis: A. Buffa et al. *Archives of Computational Methods in Engineering*, 29(7):4479–4555, 2022.

[47] W. Bulten, K. Kartasalo, P.-H. C. Chen, P. Ström, H. Pinckaers, K. Nagpal, Y. Cai, D. F. Steiner, H. Van Boven, R. Vink, et al. Artificial Intelligence for Diagnosis and Gleason Grading of Prostate Cancer: The PANDA Challenge. *Nature Medicine*, 28(1):154–163, 2022.

[48] W. Bulten, H. Pinckaers, H. Van Boven, R. Vink, T. De Bel, B. Van Ginneken, J. Van der Laak, C. Hulsbergen-Van de Kaa, and G. Litjens. Automated Deep-learning System for Gleason Grading of Prostate Cancer using Biopsies: A Diagnostic Study. *The Lancet Oncology*, 21(2):233–241, 2020.

[49] Z. Cai and H. Wu. Local Quasi-likelihood Method for Generalized Random Curve Models with Longitudinal Data. *Submitted for publication*, 2002.

[50] G. Carlsson and R. B. Gabrielsson. Topological Approaches to Deep Learning. In *Topological Data Analysis*, pages 119–146. Springer, 2020.

[51] G. Carlsson and M. Vejdemo-Johansson. *Topological Data Analysis with Applications*. Cambridge University Press, 2021.

[52] M. Carrière, F. Chazal, Y. Ike, T. Lacombe, M. Royer, and Y. Umeda. Perslay: A Neural Network Layer for Persistence Diagrams and New Graph Topological Signatures. In *International Conference on Artificial Intelligence and Statistics*, pages 2786–2796. PMLR, 2020.

[53] C. Cericola, I. J. Johnson, J. Kiers, M. Krock, J. Purdy, and J. Torrence. Extending Hypothesis Testing with Persistent Homology to Three or More Groups. *Involve, a Journal of Mathematics*, 11(1):27–51, 2017.

[54] F. Chazal, V. De Silva, M. Glisse, and S. Oudot. *The Structure and Stability of Persistence Modules*. Springer, 2016.

[55] F. Chazal, B. T. Fasy, F. Lecci, A. Rinaldo, and L. Wasserman. Stochastic Convergence of Persistence Landscapes and Silhouettes. In *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, pages 474–483, 2014.

[56] Y.-C. Chen, D. Wang, A. Rinaldo, and L. Wasserman. Statistical Analysis of Persistence Intensity Functions. *arXiv preprint arXiv:1510.02502*, 2015.

[57] S. N. Chiu, D. Stoyan, W. S. Kendall, and J. Mecke. *Stochastic Geometry and its Applications*. John Wiley & Sons, 2013.

[58] J. Cisewski-Kehe, B. T. Fasy, W. Hellwing, M. R. Lovell, P. Drozda, and M. Wu. Differentiating Small-scale Subhalo Distributions in CDM and WDM Models using Persistent Homology. *Physical Review D*, 106(2):023521, 2022.

[59] C. Conti, R. Morandi, C. Rabut, and A. Sestini. Cubic Spline Data Reduction Choosing the Knots from a Third Derivative Criterion. *Numerical Algorithms*, 28(1):45–61, 2001.

[60] L. Coradello, P. Antolin, R. Vázquez, and A. Buffa. Adaptive Isogeometric Analysis on Two-dimensional Trimmed Domains Based on a Hierarchical Approach. *Computer Methods in Applied Mechanics and Engineering*, 364:112925, 2020.

[61] T. Cover and P. Hart. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[62] D. D. Cox and J. S. Lee. Pointwise Testing with Functional Data using the Westfall-Young Randomization Method. *Biometrika*, 95(3):621–634, 2008.

[63] M. G. Cox. The Numerical Evaluation of B-Splines. *IMA Journal of Applied Mathematics*, 10(2):134–149, 1972.

[64] C. M. Crainiceanu, D. Ruppert, R. J. Carroll, A. Joshi, and B. Goodner. Spatially Adaptive Bayesian Penalized Splines with Heteroscedastic Errors. *Journal of Computational and Graphical Statistics*, 16(2):265–288, 2007.

[65] L. Crawford, A. Monod, A. X. Chen, S. Mukherjee, and R. Rabadán. Predicting Clinical Outcomes in Glioblastoma: An Application of Topological and Functional Data Analysis. *Journal of the American Statistical Association*, 115(531):1139–1150, 2020.

[66] A. Cuevas, M. Febrero, and R. Fraiman. An ANOVA Test for Functional Data. *Computational Statistics & Data Analysis*, 47(1):111–122, 2004.

[67] H. B. Curry and I. J. Schoenberg. On Pólya Frequency Functions IV: The Fundamental Spline Functions and their Limits. In *IJ schoenberg selected papers*, pages 347–383. Springer, 1988.

[68] H. Dai, J. S. Leeder, and Y. Cui. A Modified Generalized Fisher Method for Combining Probabilities from Dependent Tests. *Frontiers in Genetics*, 5:32, 2014.

[69] D. J. Daley, D. Vere-Jones, et al. *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*. Springer, 2003.

[70] J. Dauxois, A. Pousse, and Y. Romain. Asymptotic Theory for the Principal Component Analysis of a Vector Random Function: Some Applications to Statistical Inference. *Journal of multivariate analysis*, 12(1):136–154, 1982.

[71] C. De Boor. On Calculating with B-Splines. *Journal of Approximation Theory*, 6(1):50–62, 1972.

[72] C. De Boor. *A Practical Guide to Splines*. Springer-Verlag, 1978.

[73] C. De Boor and J. R. Rice. Least Squares Cubic Spline Approximation, II-Variable Knots. Technical Report CSD TR 21, Department of Computer Science, Purdue University, 1968.

[74] G. Dhom. Classification and Grading of Prostatic Carcinoma. In *Tumors of the Male Genital System*, pages 14–26. Springer, 1977.

[75] C. R. Dietrich and G. N. Newsam. Fast and Exact Simulation of Stationary Gaussian Processes through Circulant Embedding of the Covariance Matrix. *SIAM Journal on Scientific Computing*, 18(4):1088–1107, 1997.

[76] P. Diggle. A Kernel Method for Smoothing Point Process Data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 34(2):138–147, 1985.

[77] P. J. Diggle and R. J. Gratton. Monte Carlo Methods of Inference for Implicit Statistical Models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(2):193–212, 1984.

[78] P. J. Diggle, N. Lange, and F. M. Beneš. Analysis of Variance for Replicated Spatial Point Patterns in Clinical Neuroanatomy. *Journal of the American Statistical Association*, 86(415):618–625, 1991.

[79] P. J. Diggle, J. Mateu, and H. E. Clough. A Comparison Between Parametric and Non-parametric Approaches to the Analysis of Replicated Spatial Point Patterns. *Advances in Applied Probability*, 32(2):331–343, 2000.

[80] T. Dokken, T. Lyche, and K. F. Pettersen. Polynomial Splines Over Locally Refined Box-partitions. *Computer Aided Geometric Design*, 30(3):331–356, 2013.

[81] D. L. Donoho and I. M. Johnstone. Adapting to Unknown Smoothness via Wavelet Shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.

[82] S. Doyle, M. Hwang, K. Shah, A. Madabhushi, M. Feldman, and J. Tomaszeweski. Automated Grading of Prostate Cancer using Architectural and Textural Image Features. In *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1284–1287, Arlington, VA, USA, 2007. IEEE.

[83] M. Dwass. Modified Randomization Tests for Nonparametric Hypotheses. *The Annals of Mathematical Statistics*, pages 181–187, 1957.

[84] H. Edelsbrunner and J. Harer. *Computational Topology: an Introduction*. American Mathematical Society, 2010.

[85] H. Edelsbrunner, A. O. Ivanov, and R. N. Karasev. Current Open Problems in Discrete and Computational Geometry. *Modeling and Analysis of Information Systems*, 19(5):5–17, 2012.

[86] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological Persistence and Simplification. In *Proceedings 41st annual symposium on foundations of computer science*, pages 454–463. IEEE, 2000.

[87] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological Persistence and Simplification. *Discrete & Computational Geometry*, 28(4):511–533, Nov 2002.

[88] P. H. Eilers and B. D. Marx. Flexible Smoothing with B-splines and Penalties. *Statistical science*, 11(2):89–121, 1996.

[89] P. H. Eilers, B. D. Marx, and M. Durbán. Twenty Years of P-splines. *SORT: Statistics and Operations Research Transactions*, 39(2):0149–186, 2015.

[90] R. Engers. Reproducibility and Reliability of Tumor Grading in Urological Neoplasms. *World Journal of Urology*, 25(6):595–605, Dec. 2007.

[91] J. I. Epstein. *The Gleason Grading System, A Complete Guide for Pathologists and Clinicians*. Wolters Kluwer Health | Lippincott Williams and Wilkins, 2013.

[92] J. I. Epstein, W. C. Allsbrook Jr, M. B. Amin, L. L. Egevad, I. G. Committee, et al. The 2005 International Society of Urological Pathology (ISUP) Consensus Conference on Gleason Grading of Prostatic Carcinoma. *The American journal of surgical pathology*, 29(9):1228–1242, 2005.

[93] J. I. Epstein, L. Egevad, M. B. Amin, B. Delahunt, J. R. Srigley, P. A. Humphrey, G. Committee, and others. The 2014 International Society of Urological Pathology (ISUP) Consensus Conference on Gleason Grading of Prostatic Carcinoma: Definition of Grading Patterns and Proposal for a New Grading System. *The American Journal of Surgical Pathology*, 40(2):244–252, 2016.

[94] J. Fan and I. Gijbels. *Local Polynomial Modelling and its Applications: Monographs on Statistics and Applied Probability 66*. CRC Press, 1996.

[95] B. T. Fasy, J. Kim, F. Lecci, and C. Maria. Introduction to the R Package `TDA`. *arXiv preprint arXiv:1411.1830*, 2014.

[96] B. T. Fasy, F. Lecci, A. Rinaldo, L. Wasserman, S. Balakrishnan, and A. Singh. Confidence Sets for Persistence Diagrams. *The Annals of Statistics*, 42(6):2301–2339, 2014.

[97] L. Ferreira and D. B. Hitchcock. A Comparison of Hierarchical Methods for Clustering Functional Data. *Communications in statistics-simulation and computation*, 38(9):1925–1949, 2009.

[98] R. A. Finkel and J. L. Bentley. Quadtrees: A Data Structure for Retrieval on Composite Keys. *Acta informatica*, 4(1):1–9, 1974.

[99] E. Fix and J. L. Hodges. Discriminatory Analysis, Nonparametric Discrimination. 1951.

[100] D. R. Forsey and R. H. Bartels. Hierarchical B-Spline Refinement. In *Proceedings of the 15th annual conference on computer graphics and interactive techniques*, pages 205–212, 1988.

[101] J. H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.

[102] J. H. Friedman and B. W. Silverman. Flexible Parsimonious Smoothing and Additive Modeling. *Technometrics*, 31(1):3–21, 1989.

[103] P. Frosini. A Distance for Similarity Classes of Submanifolds of a Euclidean Space. *Bulletin of the Australian Mathematical Society*, 42(3):407–415, 1990.

[104] K. Fukunaga and L. Hostetler. Optimization of k nearest neighbor density estimates. *IEEE Transactions on Information Theory*, 19(3):320–326, 2003.

[105] A. Gálvez and A. Iglesias. Efficient Particle Swarm Optimization Approach for Data Fitting with Free Knot B-Splines. *Computer-Aided Design*, 43(12):1683–1692, 2011.

[106] A. Gálvez and A. Iglesias. Firefly Algorithm for Explicit B-Spline Curve Fitting to Data Points. *Mathematical Problems in Engineering*, 2013(1):528215, 2013.

[107] A. Gelman and J. Hill. *Data Analysis using Regression and Multilevel/Hierarchical Models*. Cambridge university press, 2006.

[108] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Courier Corporation, 2003.

[109] A. K. Ghosh. On Optimum Choice of k in Nearest Neighbor Classification. *Computational Statistics & Data Analysis*, 50(11):3113–3123, 2006.

[110] A. K. Ghosh. On Nearest Neighbor Classification using Adaptive Choice of k. *Journal of Computational and Graphical Statistics*, 16(2):482–502, 2007.

[111] C. Giannelli, B. Jüttler, and H. Speleers. THB-Splines: The Truncated Basis for Hierarchical Splines. *Computer Aided Geometric Design*, 29(7):485–498, 2012.

[112] E. Ginsberg, J. Schupbach, J. Sheppard, and N. Turk. Applying Factored Evolutionary Algorithms to the B-Spline Knot Selection Problem. In *2025 IEEE Symposium on Computational Intelligence in Artificial Life and Cooperative Intelligent Systems Companion (ALIFE-CIS Companion)*, pages 1–5. IEEE, 2025.

[113] V. Goepp, O. Bouaziz, and G. Nuel. Spline regression with automatic knot selection. *Computational Statistics & Data Analysis*, 202:108043, 2025.

[114] R. N. Goldman and T. Lyche. *Knot Insertion and Deletion Algorithms for B-Spline Curves and Surfaces*. SIAM, 1992.

[115] H. Goldstein. Multilevel Mixed Linear Model Analysis using Iterative Generalized Least Squares. *Biometrika*, 73(1):43–56, 1986.

[116] H. Goldstein. *Multilevel Statistical Models*. John Wiley & Sons, 2011.

[117] G. H. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10(2):413–432, 1973.

[118] T. Górecki and Ł. Smaga. fdANOVA: An R Software Package for Analysis of Variance for Univariate and Multivariate Functional Data. *Computational Statistics*, 34(2):571–597, 2019.

[119] U. Grenander. Stochastic Processes and Statistical Inference. *Arkiv för matematik*, 1(3):195–277, 1950.

[120] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A Kernel Two-sample Test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[121] A. Gretton, K. Fukumizu, Z. Harchaoui, and B. K. Sriperumbudur. A Fast, Consistent Kernel Two-sample Test. *Advances in Neural Information Processing Systems*, 22, 2009.

[122] A. Gummeson, I. Arvidsson, M. Ohlsson, N. C. Overgaard, A. Krzyzanowska, A. Heyden, A. Bjartell, and K. Aström. Automatic Gleason Grading of HE Stained Microscopic Prostate Images using Deep Convolutional Neural Networks. 2017.

[123] W. Guo. Functional Mixed Effects Models. *Biometrics*, 58(1):121–128, 2002.

[124] W. Guo. Functional Data Analysis in Longitudinal Settings using Smoothing Splines. *Statistical methods in medical research*, 13(1):49–62, 2004.

[125] A. Gálvez and A. Iglesias. Efficient Particle Swarm Optimization Approach for Data Fitting with Free Knot B-Splines. *Computer-Aided Design*, 43(12):1683–1692, Dec. 2011.

[126] A. Hajjem, F. Bellavance, and D. Larocque. Mixed-effects Random Forest for Clustered Data. *Journal of Statistical Computation and Simulation*, 84(6):1313–1328, 2014.

[127] P. Hall. On Global Properties of Variable Bandwidth Density Estimators. *The Annals of Statistics*, pages 762–778, 1992.

[128] P. Hall and N. Tajvidi. Permutation Tests for Equality of Distributions in High-dimensional Settings. *Biometrika*, 89(2):359–374, 2002.

[129] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, New York, NY, 2nd ed edition, 2009.

[130] A. Hatcher. *Algebraic Topology*. 2005.

[131] Y. Hiraoka, T. Nakamura, A. Hirata, E. G. Escolar, K. Matsue, and Y. Nishiura. Hierarchical Structures of Amorphous Solids Characterized by Persistent Homology. *Proceedings of the National Academy of Sciences*, 113(26):7035–7040, 2016.

[132] Y. Hiraoka, T. Shirai, and K. D. Trinh. Limit Theorems for Persistence Diagrams. *The Annals of Applied Probability*, 28(5):2740–2780, 2018.

[133] D. B. Hitchcock and M. C. Greenwood. Clustering Functional Data. *Handbook of cluster analysis*, pages 265–288, 2015.

[134] Y. Hochberg. A Sharper Bonferroni Procedure for Multiple Tests of Significance. *Biometrika*, 75(4):800–802, 1988.

[135] C. Hofer, R. Kwitt, M. Niethammer, and A. Uhl. Deep Learning with Topological Signatures. *Advances in neural information processing systems*, 30, 2017.

[136] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT press, 1992.

[137] S. Holm. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[138] G. Hommel. A stagewise Rejective Multiple Test Procedure Based on a Modified Bonferroni Test. *Biometrika*, 75(2):383–386, 1988.

[139] V. Howard, S. Reid, A. Baddeley, and A. Boyde. Unbiased Estimation of Particle Density in the Tandem Scanning Reflected Light Microscope. *Journal of Microscopy*, 138(2):203–212, 1985.

[140] J. Hu and S. Szymczak. A Review on Longitudinal Data Analysis with Random Forest. *Briefings in bioinformatics*, 24(2):bbad002, 2023.

[141] C.-H. Huang and D. Racoceanu. Automated High-grade Prostate Cancer Detection and Ranking on Whole Slide Images. 2017.

[142] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4135–4195, 2005.

[143] P. A. Humphrey. Gleason Grading and Prognostic Factors in Carcinoma of the Prostate. *Modern Pathology*, 17(3):292–306, Feb. 2004.

[144] J. B. Illian and D. K. Hendrichsen. Gibbs Point Process Models with Mixed Effects. *Environmetrics: The Official Journal of the International Environmetrics Society*, 21(3-4):341–353, 2010.

[145] J. B. Illian, S. H. Sørbye, and H. Rue. A Toolbox for Fitting Complex Spatial Point Process Models using Integrated Nested Laplace Approximation (INLA). *The Annals of Applied Statistics*, 6(4):1499–1530, 2012.

[146] J. B. Illian, S. H. Sørbye, H. Rue, and D. K. Hendrichsen. Using INLA to Fit a Complex Point Process Model with Temporally Varying Effects-A Case Study. *Journal of Environmental Statistics*, 3(7), 2012.

[147] G. M. James. Generalized Linear Models with Functional Predictors. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 64(3):411–432, 2002.

[148] L. Jánossy. On the Absorption of a Nucleon Cascade. In *Proceedings of the Royal Irish Academy. Section A: Mathematical and Physical Sciences*, volume 53, pages 181–188. JSTOR, 1950.

[149] O. Jiménez del Toro, M. Atzori, S. Otálora, M. Andersson, K. Eurén, M. Hedlund, P. Rönnquist, and H. Müller. Convolutional Neural Networks for an Automatic Classification of Prostate Tissue Slides with High-Grade Gleason Score. In M. N. Gurcan and J. E. Tomaszewski, editors, *Proc. SPIE Medical Imaging: 2017*, page 101400O, Mar. 2017.

[150] P. C. Johnson. Extension of Nakagawa & Schielzeth's R2GLMM to Random Slopes Models. *Methods in Ecology and Evolution*, 5(9):944–946, 2014.

[151] D. L. Jupp. The "lethargy" Theorem—A property of Approximation by γ-polynomials. *Journal of Approximation Theory*, 14(3):204–217, 1975.

[152] D. L. Jupp. Approximation to Data by Splines with Free Knots. *SIAM Journal on Numerical Analysis*, 15(2):328–343, 1978.

[153] B. Jüttler, U. Langer, A. Mantzaflaris, S. E. Moore, and W. Zulehner. Geometry+ Simulation Modules: Implementing Isogeometric Analysis. *PAMM*, 14(1):961–962, 2014.

[154] T. Kaczynski, K. Mischaikow, and M. Mrozek. *Computational Homology*, volume 157. Springer Science & Business Media, 2006.

[155] K. Karhunen. Zur Spektraltheorie Stochastischer Prozesse. *Ann. Acad. Sci. Fennicae, AI*, 34, 1946.

[156] A. Karr. *Point Processes and their Statistical Inference*. CRC Press, 1991.

[157] J. Kennedy and R. Eberhart. Particle Swarm Optimization. In *Proceedings of the International Conference on Neural Networks*, volume 4, pages 1942–1948. ieee, 1995.

[158] M. G. Kenward and J. H. Roger. Small Sample Inference for Fixed Effects from Restricted Maximum Likelihood. *Biometrics*, pages 983–997, 1997.

[159] R. King, J. B. Illian, S. E. King, G. F. Nightingale, and D. K. Hendrichsen. A Bayesian Approach to Fitting Gibbs Processes with Temporal Random Effects. *Journal of Agricultural, Biological, and Environmental Statistics*, 17(4):601–622, 2012.

[160] G. Kiss, C. Giannelli, and B. Jüttler. Algorithms and Data Structures for Truncated Hierarchical B–Splines. In *International Conference on Mathematical Methods for Curves and Surfaces*, pages 304–323. Springer, 2012.

[161] J. Kleffe. Principal Components of Random Variables with Values in a Seperable Hilbert Space. *Mathematische Operationsforschung und Statistik*, 4(5):391–406, 1973.

[162] M. Kohler. Universally Consistent Regression Function Estimation using Hierarchial B-Splines. *Journal of Multivariate Analysis*, 68(1):138–164, 1999.

[163] O. Kott, D. Linsley, A. Amin, A. Karagounis, C. Jeffers, D. Golijanin, T. Serre, and B. Gershman. Development of a Deep Learning Algorithm for the Histopathologic Diagnosis and Gleason Grading of Prostate Cancer Biopsies: A Pilot Study. *European Urology Focus*, 7(2):347–351, 2021.

[164] K. Kristensen, A. Nielsen, C. W. Berg, H. Skaug, and B. M. Bell. TMB: Automatic Differentiation and Laplace Approximation. *Journal of statistical software*, 70:1–21, 2016.

[165] M. B. Kursa and W. R. Rudnicki. Feature Selection with the Boruta Package. *Journal of statistical software*, 36:1–13, 2010.

[166] G. Kusano. On the Expectation of a Persistence Diagram by the Persistence Weighted Kernel. *Japan Journal of Industrial and Applied Mathematics*, 36(3):861–892, 2019.

[167] G. Kusano, Y. Hiraoka, and K. Fukumizu. Persistence Weighted Gaussian Kernel for Topological Data Analysis. In *International Conference on Machine Learning*, pages 2004–2013. PMLR, 2016.

[168] R. Kwitt, S. Huber, M. Niethammer, W. Lin, and U. Bauer. Statistical Topological Data Analysis-A Kernel Perspective. In *Advances in Neural Information Processing Systems*, pages 3070–3078, 2015.

[169] S. Landau and I. P. Everall. Nonparametric Bootstrap for k-functions Arising from Mixed-effects Models with Applications in Neuropathology. *Statistica Sinica*, pages 1375–1393, 2008.

[170] S. Landau, S. Rabe-Hesketh, and I. P. Everall. Nonparametric One-way Analysis of Variance of Replicated Bivariate Spatial Point Patterns. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, 46(1):19–34, 2004.

[171] P. Lawson, E. Berry, J. Q. Brown, B. T. Fasy, and C. Wenk. Topological Descriptors for Quantitative Prostate Cancer Morphology Analysis. In *Conf. Digital Pathology, SPIE Medical Imaging*, 2017.

[172] P. Lawson, J. Schupbach, B. T. Fasy, and J. W. Sheppard. Persistent Homology for the Automatic Classification of Prostate Cancer Aggressiveness in Histopathology Images. In *Medical Imaging 2019: Digital Pathology*, volume 10956, pages 72–85. International Society for Optics and Photonics, 2019.

[173] P. Lawson, A. B. Sholl, J. Q. Brown, B. T. Fasy, and C. Wenk. Persistent Homology for the Quantitative Evaluation of Architectural Features in Prostate Cancer Histology. *Scientific Reports*, 9(1):1139, Feb. 2019.

[174] S. Lee, G. Wolberg, and S. Y. Shin. Scattered Data Interpolation with Multilevel B-Splines. *IEEE Transactions on Visualization and Computer Graphics*, 3(3):228–244, 2002.

[175] P. A. Lewis and G. S. Shedler. Simulation of Nonhomogeneous Poisson Processes with Degree-two Exponential Polynomial Rate Function. *Operations Research*, 27(5):1026–1040, 1979.

[176] M. Li, H. An, R. Angelovici, C. Bagaza, A. Batushansky, L. Clark, V. Coneva, M. J. Donoghue, E. Edwards, D. Fajardo, et al. Topological Data Analysis as a Morphometric Method: using Persistent Homology to Demarcate a Leaf Morphospace. *Frontiers in Plant Science*, 9:553, 2018.

[177] W. Li, S. Xu, G. Zhao, and L. P. Goh. Adaptive Knot Placement in B-Spline Curve Approximation. *Computer-Aided Design*, 37(8):791–797, 2005.

[178] X. Li, K. Tang, M. N. Omidvar, Z. Yang, K. Qin, and H. China. Benchmark Functions for the CEC 2013 Special Session and Competition on Large-scale Global Optimization. *gene*, 7(33):8, 2013.

[179] Y. Li, M. Huang, Y. Zhang, J. Chen, H. Xu, G. Wang, and W. Feng. Automated Gleason Grading and Gleason Pattern Region Segmentation based on Deep Learning for Pathological Images of Prostate Cancer. *IEEE Access*, 8:117714–117725, 2020.

[180] M. J. Lindstrom and D. M. Bates. Newton—Raphson and EM Algorithms for Linear Mixed-effects Models for Repeated-measures Data. *Journal of the American Statistical Association*, 83(404):1014–1022, 1988.

[181] A. H. M. Linkon, M. M. Labib, T. Hasan, M. Hossain, et al. Deep Learning in Prostate Cancer Diagnosis and Gleason Grading in Histopathology Images: An Extensive Study. *Informatics in Medicine Unlocked*, 24:100582, 2021.

[182] G. Litjens, C. I. Sánchez, N. Timofeeva, M. Hermsen, I. Nagtegaal, I. Kovacs, C. Hulsbergen-van de Kaa, P. Bult, B. van Ginneken, and J. van der Laak. Deep Learning as a Tool for Increased Accuracy and Efficiency of Histopathological Diagnosis. *Scientific Reports*, 6:26286, 2016.

[183] D. O. Loftsgaarden and C. P. Quesenberry. A Nonparametric Estimate of a Multivariate Density Function. *The Annals of Mathematical Statistics*, 36(3):1049–1051, 1965.

[184] N. T. Longford. A Fast Scoring Algorithm for Maximum Likelihood Estimation in Unbalanced Mixed Models with Nested Random Effects. *Biometrika*, 74(4):817–827, 1987.

[185] J. Luo, H. Kang, and Z. Yang. Knot Placement for B-Spline Curve Approximation via L-infinity, 1-norm and Differential Evolution Algorithm. *Journal of Computational Mathematics*, 40(4):592–609, 2021.

[186] Z. Luo and G. Wahba. Hybrid Adaptive Splines. *Journal of the American Statistical Association*, 92(437):107–116, 1997.

[187] A. Madabhushi and G. Lee. Image Analysis and Machine Learning in Digital Pathology: Challenges and Opportunities. *Medical Image Analysis*, 33:170–175, 2016.

[188] F. Mandel, R. P. Ghosh, and I. Barnett. Neural networks for clustered and longitudinal data using mixed effects models. *Biometrics*, 79(2):711–721, 2023.

[189] R. Marcus, P. Eric, and K. R. Gabriel. On Closed Testing Procedures with Special Reference to Ordered Analysis of Variance. *Biometrika*, 63(3):655–660, 1976.

[190] C. Maria, J.-D. Boissonnat, M. Glisse, and M. Yvinec. The GUDHI Library: Simplicial Complexes and Persistent Homology. In *International congress on mathematical software*, pages 167–174. Springer, 2014.

[191] J. Mateu. Parametric Procedures in the Analysis of Replicated Pairwise Interaction Point Patterns. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, 43(3):375–394, 2001.

[192] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Routledge, New York, NY, 2019.

[193] M. R. McGuirl, A. Volkening, and B. Sandstede. Topological Data Analysis of Zebrafish Patterns. *Proceedings of the National Academy of Sciences*, 117(10):5113–5124, 2020.

[194] K. Meng, J. Wang, L. Crawford, and A. Eloyan. Randomness and Statistical Inference of Shapes via the Smooth Euler Characteristic Transform. *arXiv preprint arXiv:2204.12699*, 2022.

[195] L. Mentch and G. Hooker. Quantifying Uncertainty in Random Forests via Confidence Intervals and Hypothesis Tests. *Journal of Machine Learning Research*, 17(26):1–41, 2016.

[196] D. Michel and A. Zidna. A New Deterministic Heuristic Knots Placement for B-Spline Approximation. *Mathematics and Computers in Simulation*, 186:91–102, 2021.

[197] B. Mitchell and J. Sheppard. Spatially Biased Random Forests. In *FLAIRS*, pages 20–25, 2019.

[198] A. Mitra and Ž. Virk. The Space of Persistence Diagrams on n Points Coarsely Embeds into Hilbert Space. *Proceedings of the American Mathematical Society*, 149(6):2693–2703, 2021.

[199] S. Miyata and X. Shen. Adaptive Free-knot Splines. *Journal of Computational and Graphical Statistics*, 12(1):197–213, 2003.

[200] S. Miyata and X. Shen. Free-Knot Splines and Adaptive Knot Selection. *Journal of the Japan Statistical Society*, 35(2):303–324, 2005.

[201] S. D. Mohanty and E. Fahnestock. Adaptive Spline Fitting with Particle Swarm Optimization. *Computational Statistics*, 36(1):155–191, 2021.

[202] A. Möller, G. Tutz, and J. Gertheiss. Random Forests for Functional Covariates. *Journal of Chemometrics*, 30(12):715–725, 2016.

[203] J. Moller and R. P. Waagepetersen. *Statistical Inference and Simulation for Spatial Point Processes*. CRC Press, 2003.

[204] D. Morozov. Dionysus: A C++ Library for Computing Persistent Homology, 2007.

[205] J. S. Morris. Functional Regression. *Annual Review of Statistics and Its Application*, 2(1):321–359, 2015.

[206] F. Mosteller, J. W. Tukey, et al. Data Analysis, Including Statistics. *Handbook of Social Psychology*, 2:80–203, 1968.

[207] T. Mrkvička and M. Myllymäki. False Discovery Rate Envelopes. *Statistics and Computing*, 33(5):109, 2023.

[208] T. Mrkvička and M. Myllymäki. Comparison of Approaches for Local Testing with Functional Test Statistics. *Journal of Statistical Computation and Simulation*, 94(16):3555–3572, 2024.

[209] T. Mrkvicka, M. Myllymaki, M. Jilek, and U. Hahn. A One-way ANOVA Test for Functional Data with Graphical Interpretation. *arXiv preprint arXiv:1612.03608v4*, 2020.

[210] Y. Mun, I. Paik, S.-J. Shin, T.-Y. Kwak, and H. Chang. Yet Another Automated Gleason Grading System (YAAGGS) by Weakly Supervised Deep Learning. *npj Digital Medicine*, 4(1):99, 2021.

[211] J. R. Munkres. *Elements of Algebraic Topology*. CRC Press, 2018.

[212] M. Myllymäki, T. Mrkvička, P. Grabarnik, H. Seijo, and U. Hahn. Global Envelope Tests for Spatial Processes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(2):381–404, 2017.

[213] K. Nagpal, D. Foote, Y. Liu, Po-Hsuan, Chen, E. Wulczyn, F. Tan, N. Olson, J. L. Smith, A. Mohtashamian, J. H. Wren, G. S. Corrado, R. MacDonald, L. H. Peng, M. B. Amin, A. J. Evans, A. R. Sangoi, C. H. Mermel, J. D. Hipp, and M. C. Stumpe. Development and Validation of a Deep Learning Algorithm for Improving Gleason Scoring of Prostate Cancer. *arXiv:1811.06497 [cs]*, Nov. 2018. arXiv: 1811.06497.

[214] K. Nagpal, D. Foote, F. Tan, Y. Liu, P.-H. C. Chen, D. F. Steiner, N. Manoj, N. Olson, J. L. Smith, A. Mohtashamian, et al. Development and Balidation of a Deep Learning Algorithm for Gleason Grading of Prostate Cancer from Biopsy Specimens. *JAMA Oncology*, 6(9):1372–1380, 2020.

[215] M. Naser, M. K. Al-Bashiti, A. T. G. Tapeh, A. Naser, V. Kodur, R. Hawileh, J. Abdalla, N. Khodadadi, A. H. Gandomi, and A. D. Eslamlou. A Review of Benchmark and Test Functions for Global Optimization Algorithms and Metaheuristics. *Wiley Interdisciplinary Reviews: Computational Statistics*, 17(2):e70028, 2025.

[216] M. K. K. Niazi, K. Yao, D. Zynger, S. Clinton, J. Chen, M. Koyuturk, T. LaFramboise, and M. Gurcan. Visually Meaningful Histopathological Features for Automatic Grading of Prostate Cancer. *IEEE Journal of Biomedical and Health Informatics*, PP(99):1–1, 2016.

[217] T. E. Nichols and A. P. Holmes. Nonparametric Permutation Tests for Functional Neuroimaging: A Primer with Examples. *Human Brain Mapping*, 15(1):1–25, 2002.

[218] M. Nicolau, A. J. Levine, and G. Carlsson. Topology Based Data Analysis Identifies a Subgroup of Breast Cancers with a Unique Mutational Profile and Excellent Survival. *Proceedings of the National Academy of Sciences*, page 201102826, 2011.

[219] Y. Ogata and K. Katsura. Likelihood Analysis of Spatial Inhomogeneity for Marked Point Patterns. *Annals of the Institute of Statistical Mathematics*, 40(1):29–39, 1988.

[220] N. L. Olsen, A. Pini, and S. Vantini. False Discovery Rate for Functional Data. *Test*, 30(3):784–809, 2021.

[221] S. M. Omohundro. Five balltree construction algorithms. 1989.

[222] T. A. Ozkan, A. T. Eruyar, O. O. Cebeci, O. Memik, L. Ozcan, and I. Kuskonmaz. Interobserver Variability in Gleason Histological Grading of Prostate Cancer. *Scandinavian Journal of Urology*, 50(6):420–424, 2016.

[223] D. Pantazis, T. E. Nichols, S. Baillet, and R. M. Leahy. A Comparison of Random Field Theory and Permutation Methods for the Statistical Analysis of MEG Data. *Neuroimage*, 25(2):383–394, 2005.

[224] F. Papangelou. The Conditional Intensity of General Point Processes and an Application to Line Processes. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 28(3):207–226, 1974.

[225] H. Park and J.-H. Lee. B-Spline Curve Fitting Based on Adaptive Curve Refinement using Dominant Points. *Computer-Aided Design*, 39(6):439–451, 2007.

[226] V. Patrangenaru, P. Bubenik, R. L. Paige, and D. Osborne. Topological Data Analysis for Object Data. *arXiv preprint arXiv:1804.10255*, 2018.

[227] V. Patrangenaru and L. Ellingson. *Nonparametric Statistics on Manifolds and their Applications to Object Data Analysis*. CRC Press, Taylor & Francis Group Boca Raton, 2016.

[228] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[229] M. Pellagatti, C. Masci, F. Ieva, and A. M. Paganoni. Generalized Mixed-effects Random Forest: A Flexible Approach to Predict University Student Dropout. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 14(3):241–257, 2021.

[230] L. Piegl and W. Tiller. *The NURBS Book*. Springer Science & Business Media, 2012.

[231] P. M. Pierorazio, P. C. Walsh, A. W. Partin, and J. I. Epstein. Prognostic gleason grade grouping: Data based on the modified gleason scoring system. *BJU international*, 111(5):753, 2013.

[232] J. C. Pinheiro and D. M. Bates. *Mixed-effects Models in S and S-PLUS*. Springer, 2000.

[233] A. Pini and S. Vantini. Interval-wise Testing for Functional Data. *Journal of Nonparametric Statistics*, 29(2):407–424, 2017.

[234] J. Pittman. Adaptive Splines and Genetic Algorithms. *Journal of Computational and Graphical Statistics*, 11(3):615–638, 2002.

[235] D. N. Politis. Adaptive Bandwidth Choice. *Journal of Nonparametric Statistics*, 15(4-5):517–533, 2003.

[236] W. Poole, D. L. Gibbs, I. Shmulevich, B. Bernard, and T. A. Knijnenburg. Combining Dependent P-values with an Empirical Adaptation of Brown's Method. *Bioinformatics*, 32(17):i430–i436, 2016.

[237] T. Pospisil and A. B. Lee. (f) RFCDE: Random Forests for Conditional Density Estimation and Functional Data. *arXiv preprint arXiv:1906.07177*, 2019.

[238] F. Pourakpour, Á. Szölgyén, R. Nateghi, D. A. Gutman, D. Manthey, and L. A. Cooper. HistomicsTK: A Python Toolkit for Pathology Image Analysis Algorithms. *SoftwareX*, 31:102318, 2025.

[239] P. Pranav, H. Edelsbrunner, R. Van de Weygaert, G. Vegter, M. Kerber, B. J. Jones, and M. Wintraecken. The Topology of the Cosmic Web in Terms of Persistent Betti Numbers. *Monthly Notices of the Royal Astronomical Society*, 465(4):4281–4310, 2017.

[240] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2025.

[241] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2025.

[242] R. Rabadan and A. J. Blumberg. *Topological Data Analysis for Genomics and Evolution: Topology in Biology*. Cambridge University Press, 2019.

[243] R. Rahman, S. R. Dhruba, S. Ghosh, and R. Pal. Functional Random Forest with Applications in Dose-response Predictions. *Scientific Reports*, 9(1):1628, 2019.

[244] P. Ramón, M. de la Cruz, J. Chacón-Labella, and A. Escudero. A New Non-parametric Method for Analyzing Replicated Point Patterns in Ecology. *Ecography*, 39(11):1109–1117, 2016.

[245] J. O. Ramsay. When the Data are Functions. *Psychometrika*, 47(4):379–396, 1982.

[246] J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer, 1997.

[247] C. E. Rasmussen. Gaussian Processes in Machine Learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

[248] S. L. Rathbun and N. Cressie. Asymptotic Properties of Estimators for the Parameters of Spatial Inhomogeneous Poisson Point Processes. *Advances in Applied Probability*, 26(1):122–154, 1994.

[249] A. Razdan. Knot Placement for B-Spline Curve Approximation. Technical Report No Number, Tempe, AZ: Arizona State University, 1999.

[250] J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt. A Stable Multi-scale Kernel for Topological Machine Learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4741–4748, 2015.

[251] H. Rezaeilouyeh, A. Mollahosseini, and M. H. Mahoor. Microscopic Medical Image Classification Framework via Deep Learning and Shearlet Transform. *Journal of Medical Imaging*, 3(4):044501, Nov. 2016.

[252] J. R. Rice. On the Degree of Convergence of Nonlinear Spline Approximation. In *Approximations with Special Emphasis on Spline Functions*, pages 349–365. New York: Academic Press, 1969.

[253] B. D. Ripley. The Second-order Analysis of Stationary Point Processes. *Journal of applied probability*, 13(2):255–266, 1976.

[254] V. Robins. Towards Computing Homology from Finite Approximations. In *Topology Proceedings*, volume 24, pages 503–532, 1999.

[255] V. Robins and K. Turner. Principal Component Analysis of Persistent Homology Rank Functions with Case Studies of Spatial Point Patterns, Sphere Packing and Colloids. *Physica D: Nonlinear Phenomena*, 334:99–117, 2016.

[256] A. Robinson and K. Turner. Hypothesis Testing for Topological Data Analysis. *Journal of Applied and Computational Topology*, 1(2):241–261, 2017.

[257] D. Rogers. G/SPLINES: A Hybrid of Friedman's Multivariate Adaptive Regression Splines (MARS) Algorithm with Holland's Genetic Algorithm. Technical report, 1991.

[258] J. P. Romano and M. Wolf. Exact and Approximate Stepdown Methods for Multiple Hypothesis Testing. *Journal of the American Statistical Association*, 100(469):94–108, 2005.

[259] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[260] H. Rue, S. Martino, and N. Chopin. Approximate Bayesian Inference for Latent Gaussian Models by using Integrated Nested Laplace Approximations. *Journal of the Royal Statistical Society: Series b (Statistical Methodology)*, 71(2):319–392, 2009.

[261] O. Schabenberger and C. A. Gotway. *Statistical Methods for Spatial Data Analysis*. Chapman and Hall/CRC, 2017.

[262] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers. Cell Detection with Star-convex Polygons. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 265–273. Springer, 2018.

[263] I. J. Schoenberg. Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions: Part A. 1946.

[264] J. Schupbach, J. W. Sheppard, and T. Forrester. Quantifying Uncertainty in Neural Network Ensembles using U-statistics. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[265] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche. T-spline Simplification and Local Refinement. *ACM transactions on graphics (TOG)*, 23(3):276–283, 2004.

[266] S. G. Self and K.-Y. Liang. Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. *Journal of the American Statistical Association*, 82(398):605–610, 1987.

[267] M. Shinozuka and G. Deodatis. Simulation of Stochastic Processes by Spectral Representation. 1991.

[268] M. Shinozuka and G. Deodatis. Simulation of Multi-dimensional Gaussian Stochastic Fields by Spectral Representation. 1996.

[269] G. Simchoni and S. Rosset. Integrating Random Effects in Deep Neural Networks. *Journal of Machine Learning Research*, 24(156):1–57, 2023.

[270] V. Skytt and T. Dokken. Scattered data approximation by lr b-spline surfaces: A study on refinement strategies for efficient approximation. In *Geometric Challenges in Isogeometric Analysis*, pages 217–258. Springer, 2022.

[271] T. A. Snijders and R. Bosker. Multilevel Analysis: An Introduction to Basic and Advanced Multilevel Modeling. 2011.

[272] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian Optimization of Machine Learning Algorithms. *Advances in Neural Information Processing Systems*, 25, 2012.

[273] S. S. Sørensen, C. A. Biscio, M. Bauchy, L. Fajstrup, and M. M. Smedskjaer. Revealing Hidden Medium-range Order in Amorphous Materials Using Topological Data Analysis. *Science Advances*, 6(37):eabc2320, 2020.

[274] T. Sousbie. The Persistent Cosmic Web and its Filamentary Structure–I. Theory and Implementation. *Monthly Notices of the Royal Astronomical Society*, 414(1):350–383, 2011.

[275] T. Sousbie, C. Pichon, and H. Kawahara. The Persistent Cosmic Web and its Filamentary Structure–II. Illustrations. *Monthly Notices of the Royal Astronomical Society*, 414(1):384–403, 2011.

[276] J. R. Stark, S. Perner, M. J. Stampfer, J. A. Sinnott, S. Finn, A. S. Eisenstein, J. Ma, M. Fiorentino, T. Kurth, M. Loda, et al. Gleason score and lethal prostate cancer: does 3+4=4+3? *Journal of Clinical Oncology*, 27(21):3459, 2009.

[277] B. Stolz. Computational Topology in Neuroscience. *Unpublished doctoral dissertation. University of Oxford*, 2014.

[278] C. J. Stone, M. H. Hansen, C. Kooperberg, and Y. K. Truong. Polynomial Splines and their Tensor Products in Extended Linear Modeling: 1994 Wald Memorial Lecture. *The Annals of Statistics*, 25(4):1371–1470, 1997.

[279] M. Stone. Cross-validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, 1974.

[280] R. Storn and K. Price. Differential Evolution–A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces. *Journal of Global Optimization*, 11:341–359, 1997.

[281] D. Stoyan and H. Stoyan. *Fractals, Random Shapes and Point Fields: Methods of Geometrical Statistics*, volume 302. Wiley-Blackwell, 1994.

[282] S. Strasser, J. Sheppard, N. Fortier, and R. Goodman. Factored Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 21(2):281–293, 2016.

[283] W. W. Stroup. Rethinking the analysis of non-normal data in plant and soil science. *Agronomy Journal*, 107(2):811–827, 2015.

[284] M.-N. Tran, N. Nguyen, D. Nott, and R. Kohn. Bayesian Deep Net GLM and GLMM. *Journal of Computational and Graphical Statistics*, 29(1):97–113, 2020.

[285] K. Turner, Y. Mileyko, S. Mukherjee, and J. Harer. Fréchet Means for Distributions of Persistence Diagrams. *Discrete & Computational Geometry*, 52(1):44–70, 2014.

[286] K. Turner, S. Mukherjee, and D. M. Boyer. Persistent Homology Transform for Modeling Shapes and Surfaces. *Information and Inference: A Journal of the IMA*, 3(4):310–344, 2014.

[287] K. Turner, S. Mukherjee, and D. M. Boyer. Persistent Homology Transform for Modeling Shapes and Surfaces. *Information and Inference: A Journal of the IMA*, 3(4):310–344, 2014.

[288] S. Ullah and C. F. Finch. Applications of Functional Data Analysis: A Systematic Review. *BMC medical research methodology*, 13(1):43, 2013.

[289] G. Unther Greiner and K. Hormann. Interpolating and Approximating Scattered 3D-data with Hierarchical Tensor Product B-Splines. In *Proceedings of Chamonix*, volume 1, 1996.

[290] D. Ushizima, D. Morozov, G. H. Weber, A. G. Bianchi, J. A. Sethian, and E. W. Bethel. Augmented Topological Descriptors of Pore Networks for Material Science. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2041–2050, 2012.

[291] R. Van De Weygaert, G. Vegter, H. Edelsbrunner, B. J. Jones, P. Pranav, C. Park, W. A. Hellwing, B. Eldering, N. Kruithof, E. P. Bos, et al. Alpha, Betti and the Megaparsec Universe: on the Topology of the Cosmic Web. In *Transactions on Computational Science XIV*, pages 60–101. Springer, 2011.

[292] F. Van den Bergh and A. P. Engelbrecht. A Cooperative Approach to Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):225–239, 2004.

[293] M. Van Lieshout. Non-parametric Adaptive Bandwidth Selection for Kernel Estimators of Spatial Intensity Functions. *Annals of the Institute of Statistical Mathematics*, 76(2):313–331, 2024.

[294] O. Vsevolozhskaya, M. Greenwood, and D. Holodov. Pairwise Comparison of Treatment Levels in Functional Analysis of Variance with Application to Erythrocyte Hemolysis. *The Annals of Applied Statistics*, 8(2):905–925, 2014.

[295] O. A. Vsevolozhskaya, M. C. Greenwood, G. Bellante, S. L. Powell, R. L. Lawrence, and K. S. Repasky. Combining Functions and the Closure Principle for Performing Follow-up Tests in Functional Analysis of Variance. *Computational Statistics & Data Analysis*, 67:175–184, 2013.

[296] O. A. Vsevolozhskaya, M. C. Greenwood, S. L. Powell, and D. V. Zaykin. Resampling-based Multiple Comparison Procedure with Application to Point-wise Testing with Functional Data. *Environmental and Ecological Statistics*, 22(1):45–59, 2015.

[297] J.-L. Wang, J.-M. Chiou, and H.-G. Müller. Functional Data Analysis. *Annual Review of Statistics and its application*, 3:257–295, 2016.

[298] W. Wang, H. Wang, G. Dai, and H. Wang. Visualization of Large Hierarchical Data by Circle Packing. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 517–520, 2006.

[299] M. Weigert and U. Schmidt. Nuclei Instance Segmentation and Classification in Histopathology Images with StarDist. In *2022 IEEE International Symposium on Biomedical Imaging Challenges (ISBIC)*, pages 1–4. IEEE, 2022.

[300] M. Weigert, U. Schmidt, R. Haase, K. Sugawara, and G. Myers. Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy. In *Proceedings of the*

IEEE/CVF Winter Conference on Applications of Computer Vision, pages 3666–3673, 2020.

[301] P. H. Westfall and S. S. Young. *Resampling-based Multiple Testing: Examples and Methods for p-value Adjustment*. John Wiley and Sons, 1993.

[302] H. Wilson, P. Diggle, and C. V. Howard. Methods for the Analysis of Replicated Spatial Point Patterns in Clinical Neuro-anatomy. *Advances in Applied Probability*, 30(2):293–294, 1998.

[303] A. T. Wood and G. Chan. Simulation of Stationary Gaussian Processes in $[0, 1]^d$. *Journal of computational and graphical statistics*, 3(4):409–432, 1994.

[304] H. Wu and J.-T. Zhang. Local Polynomial Mixed-effects Models for Longitudinal Data. *Journal of the American Statistical Association*, 97(459):883–897, 2002.

[305] H. Wu and J.-T. Zhang. *Nonparametric Regression Methods for Longitudinal Data Analysis: Mixed-effects Modeling Approaches*. John Wiley & Sons, 2006.

[306] K. Xia and G.-W. Wei. Persistent Homology Analysis of Protein Structure, Flexibility, and Folding. *International Journal for Numerical Methods in Biomedical Engineering*, 30(8):814–844, 2014.

[307] Y. Xiong, H. J. Kim, and V. Singh. Mixed Effects Neural Networks (menets) with Applications to Gaze Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7743–7752, 2019.

[308] G. Xu, C. Zhao, A. Jalilian, R. Waagepetersen, J. Zhang, and Y. Guan. Nonparametric Estimation of the Pair Correlation Function of Replicated Inhomogeneous Point Processes. *Electronic Journal of Statistics*, 14(2):3730–3765, 2020.

[309] M. Xu and P. T. Reiss. Distribution-free Pointwise Adjusted P-values for Functional Hypotheses. In *International Workshop on Functional and Operatorial Statistics*, pages 245–252. Springer, 2020.

[310] X. Xu, J. Cisewski-Kehe, S. B. Green, and D. Nagai. Finding Cosmic Voids and Filament Loops using Topological Data Analysis. *Astronomy and Computing*, 27:34–52, 2019.

[311] M.-M. Yau and S. N. Srihari. A Hierarchical Data Structure for Multidimensional Digital Images. *Communications of the ACM*, 26(7):504–515, 1983.

[312] R. Yeh, Y. S. Nashed, T. Peterka, and X. Tricoche. Fast Automatic Knot Placement Method for Accurate B-Spline Curve Fitting. *Computer-Aided Design*, 128:102905, 2020.

[313] F. Yoshimoto, T. Harada, and Y. Yoshimoto. Data Fitting with a Spline using a Real-coded Genetic Algorithm. *Computer-Aided Design*, 35(8):751–760, 2003.

[314] H. Zhang and Z. Wu. The Generalized Fisher's Combination and Accurate p-value Calculation under Dependence. *Biometrics*, 79(2):1159–1172, 2023.

[315] J. Zhang. Analysis of Variance for Functional Data. *Monographs on Statistics and Applied Probability*, 127:127, 2014.

[316] S. Zhou and X. Shen. Spatially Adaptive Regression Splines and Accurate Knot Selection Schemes. *Journal of the American Statistical Association*, 96(453):247–259, 2001.

[317] A. Zomorodian and G. Carlsson. Computing Persistent Homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.