Orthogonality in Machine Learning

William Greenall

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Statistical Science

University College London

June 21, 2024

I, William Greenall, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

In this thesis I focus on the applications and relevance of orthogonality in various topics in machine learning. The theme of the thesis is that different viewpoints of the concept of orthogonality and Hilbert spaces in general can be utilised to improve the performance of machine learning algorithms, as well as inform development of new ones. The approach taken focuses in part on the rich and interesting theory of orthogonal polynomials, which are heretofore underutilised in machine learning methods as a tool for feature construction

First, I look at a sparse Gaussian process schema relying on appropriate construction of orthogonal basis functions, as well as relevant theory that shows that orthonormality is an important feature of the chosen sparse method. This yields a novel approach to feature construction and sparse Gaussian process regression.

Next, I utilise orthogonality and an appropriately defined inner product as a tool for a new form of interpretable feature construction in problems with dynamic graphs. The approach centres on comparison between graphs via an implicit measure of orthogonality of their matching polynomials. This is applied to anomaly detection as a guiding example, using a ”landmarks” strategy.

Finally, I propose a new type of Gaussian Cox process, which yields application of orthogonal series estimate models in order to

construct a rapid Bayesian inference scheme, bypassing the usual difficulties of the highly non-Gaussian likelihood. This is then extended, through appropriate approximation schemata for higher-order Gaussian moments, to stochastic classification models, yielding a rapid and flexible stochastic classifier, whose predictions can be interpreted as exact probabilities and yield direct uncertainty quantification. This stands in contrast to standard models that train on degenerate distributions to yield probabilistic predictions in an ad-hoc fashion.

Impact Statement

The work in this thesis may have impact in both academic and industrial settings. The approach developed in the first chapter should improve predictive capability in any situation where Gaussian process models are used. This could be widespread, given that Gaussian process models are widely-used paradigm in general machine learning problems. The computational cost of methods translates directly to computing time, which has a cost both in financial and energy terms. As a result it is not easy to quantify *ex ante* the potential impact of the work in the first chapter. I expect to publish a paper based on the material in this chapter over the course of the next year at a top machine learning conference.

Graph-based methods have proliferated, and interpretability is a key concern in many of these models. The work in the second chapter should improve the interpretability of graph-based models, and so could have impact in any situation where such models are used. This could be widespread, given the increasing use of graph-based models in many areas of machine learning. Again, it is not easy to quantify *ex ante* the potential impact of the work in the second chapter. I also expect to find an appropriate venue for publication of a paper based on the material in this chapter over the course of the next year.

The work in the third chapter is more directly applicable to a

specific industrial setting. Point process data is widespread in many areas, and the computational efficiency exhibited by the method may have many applications in industry. As noted in the thesis, the work in this chapter comprises the basis of a paper that has been submitted to a top machine learning conference, and I expect to receive feedback in the next couple of months.

Because the thesis focuses on methodological development, the impact of the work is likely to be felt in the medium to long term. The work is likely to be of interest to researchers in machine learning and statistics, and so the impact is likely to be felt in the academic community more than in industry. However, I expect that review feedback will provide examples of appropriate, unforeseen applications of the methods described in the thesis.

Acknowledgements

I would like to thank my supervisor, Petros, who was the best supervisor I could have asked for. He was there whenever I needed him, and perhaps just as importantly, he was not there when I didn't.

I am grateful to Eleanna, for supporting me throughout.

I am thankful for my parents, for their support and encouragement throughout my studies.

Finally, i would like to dedicate this thesis to the memory of Mike Tsionas, teacher and friend, whom I never got to thank for the effort he made to make sure I understood his lessons.

Notation

$(\mathcal{X}, \mathcal{F}, \nu)$: A measure space on a set $\mathcal{X}$, with measure ν and an appropriately-defined σ-algebra $\mathcal{F}$.

$\mathbb{E}_f[x]$: The expection of a random variable x w.r.t the density f.

$\mathbb{V}_f[x]$: The variance of a random variable x w.r.t the density f.

$\mathrm{Cov}_f[x, y]$: The covariance between random variables x, y w.r.t the density f.

$k(x, x')$: A kernel function describing the covariance between x and x', in the Gaussian process formulation.

$D_{KL}(f \,||\, g)$: The Kullback-Leibler divergence from g to f.

$\mathcal{GP}$: A Gaussian process.

θ_i: The random coefficient for the i-th basis function in the Gaussian process.

ϕ_i: an element of an orthonormal basis.

λ: An eigenvalue of the Mercer kernel representation.

m: The number of basis functions in the truncation; i.e. the bandwidth of a truncated kernel.

$\mathcal{H}$: A Hilbert space of functions, endowed with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$.

$T_k[f]$: Operator applying kernel k to the function f.

$\mathbf{x}$: A vector of inputs.

$\mathcal{X}$: The space of inputs to a Gaussian process.

$\mathbf{y}$: A vector of outputs/labels.

$\mathcal{Y}$: The space of outputs to a Gaussian process.

$\mathcal{L}^2(\nu)$: The space of square-integrable function, with inner product constructed with measure ν.

l^2: The space of square-integrable sequences.

$A \odot B$: The Hadamard product of two matrices A and B.

$\mathcal{L}$: A linear moment functional.

$\left\langle \mathcal{L}, x^k \right\rangle$: A moment linear functional applied to the monomial x^k.

$\mathcal{G}$: a graph, constructed of nodes and edges.

σ: a complete node sequence.

$k_m^{\nu}(\cdot,\cdot)$: A Christoffel-Darboux kernel of order m with measure ν.

haf(A): the hafnian of a matrix A.

per(A): the permanent of a matrix A.

det(A): the determinant of a matrix A.

S_n: the set of all permutations of n elements.

μ_k: The k-th moment of a Linear moment functional.

w: A weight function for construction of an orthogonal basis.

$\ell(x)$: A positive-valued link function.

Ψ: An intensity measure.

ψ: An intensity (density) function.

$\mathcal{N}-\Gamma^{-1}$: A normal-inverse-gamma distribution.

η: A weighting parameter for the conjugate Bayesian model.

lhaf(A): the loop hafnian of a matrix A.

$\mathcal{I}_n$: the set of all involutions of n elements.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Orthogonality is a fundamental concept in many theoretical and practical applications of statistical science and machine learning. The concept of orthogonality is intertwined with the concept of the inner product, which is usually described as a measure of the similarity between two objects. The inner product is a fundamental concept in linear algebra and its applications extend to more exotic mathematical structures.

Spaces of objects endowed with an inner product are not limited to classical vector spaces. Function spaces can be endowed with an inner product, and so the concept of orthogonality can be extended to functions. Constructing models using orthogonal functions is a common practice in many fields, as it yields various advantages. To state just two examples, in signal processing, the Fourier transform decomposes a signal into a sum of orthogonal functions, which can be used to analyze and process the signal (Rudin, 1976); in quantum mechanics, the eigenfunctions of Hermitian operators form an orthogonal basis, which is used to represent quantum states (B. C. Hall, 2013).

In this thesis I present three applications of orthogonality to

machine learning problems. First, I highlight the importance of orthonormality in the choice of basis for constructing sparse Gaussian process models. Gaussian process models use the properties of the Gaussian distribution to represent information about infinite dimensional operators using finite dimensional matrices. Many applications of Gaussian processes rely on approximations to the operator, but ignore the extent to which this operator remains well-approximated by a finite-dimensional counterpart. I show that, in one case of sparse approximation in Gaussian process models, if the basis functions used to represent the behaviour of the operator are not orthonormal, the finite-dimensional approximation will be poor. I then present a way to construct asymptotically orthonormal basis functions for the Gaussian process, and show that this yields a sparse model that is asymptotically exact. This yields a novel approach to feature construction and sparse Gaussian process regression.

Next, I propose a method for embedding and comparing graphs by calculating an orthogonal polynomial sequence for each graph. Each graph yields a corresponding inner product, and graphs can be compared based on their inner product embeddings in an interpretable fashion. To do this I utilise the matching polynomial of a graph. Certain graphs have matching polynomials that are also orthogonal polynomials. I show that by appropriate application of the spectral theorem, we can construct an estimator for the measure of orthogonality of a given graph's matching polynomial. I exhibit through examples how graphs that have similar measures of orthogonality can then be considered to be similar. This is applied to anomaly detection as a guiding example, using a "landmarks" strategy that captures anomalous graphs as differing excessively from a given baseline.

Finally, I look at point process modelling. The standard approaches to Gaussian Cox point process modelling yield complex likelihood functions, and much of the literature has considered ways to render the problem tractable. I propose a new type of Gaussian Cox process which yields a representation of the process as a sum of orthogonal functions, which in turn provides a rapid Bayesian inference scheme, bypassing the usual difficulties of the highly non-Gaussian likelihood. This is then extended, through appropriate approximation schemata for higher-order Gaussian moments, to stochastic classification models, yielding a flexible stochastic classifier whose predictions can be interpreted as exact probabilities and yield direct uncertainty quantification. This places the approach in contrast to standard models that train on degenerate distributions to yield probabilistic predictions in an ad-hoc fashion.

Chapter 2

Preliminaries

In this chapter, we present some of the preliminary concepts and definitions that will be found throughout the thesis. The aim is to provide sound theoretical background for the methods that will be developed through the main chapters. Concepts specific to a given chapter will be presented there; more general background is found in the present chapter.

2.1 Stochastic processes

Stochastic processes are a fundamental concept in probability theory. They link the analysis of given random variables to generalised collections of random variables, providing the ability to model complex random phenomena.

Definition 1 (Stochastic Process)**.** *A stochastic process f on $\mathcal{X}$ is a collection of random variables $\{f(x)\}$, indexed by x in index set $\mathcal{X}$.*

In any given application, the interpretation of the index variable will be key to the applicability of the stochastic process model. A natural one-dimensional stochastic process model might have time as the index variable, and a given realisation of the stochastic process

is a function describing the development of some variable of interest over time.

Naturally, stochastic processes are trivial unless some model of dependence is constructed that describes the relationship between the random variables that constitute the realisation of the process. So-called *second-order* stochastic processes yield a form whose dependence model is described by a covariance function

Definition 2. *A second-order stochastic process is a stochastic process* $f(x)$ *on* $\mathcal{X}$ *such that*

$$\mathbb{E}\left[f(x)^2\right] \leq +\infty \; \forall \, x \in \mathcal{X}$$

We also define the covariance function, which is will be all that is necessary to describe the dependence model of the type of second-order stochastic processes that we will work with in this thesis.

Definition 3. *A covariance function* $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *is a function of two arguments that is symmetric* $(c(x,x') = c(x',x))$*, and positive definite, meaning that*

$$\sum_{i=1}^{N} \sum_{j=1}^{N} c(x_i,x_j)\alpha_i\alpha_j \geq 0$$

for any N*, any* $x_1, x_2, ..., x_N \in \mathcal{X}$ *and any* $\alpha_i \in \mathbb{R}$*.*

The two concepts are tied together by Loève's theorem, which constructs a bijection between second-order stochastic processes and covariance functions.

Theorem 1 (Loève theorem (Loève, 1977))**.** *A function* $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *is the covariance function of a second-order stochastic process if and only*

if it is symmetric and positive definite.

2.2 Functional analysis

Various basic concepts from functional analysis will be used throughout the thesis, and we present the preliminary ideas here. First, we define a Hilbert space:

Definition 4 (Hilbert Space)**.** *A Hilbert space is an inner product space that is complete with respect to the distance function implied by the inner product.*

That is, a vector space whose elements can be compared using the inner product. Often covariance functions are described as *kernels*. A kernel provides a generalisation of a positive definite matrix to an infinite dimensional Hilbert space. Specifically, just as matrices are used to represent operators on vector spaces, kernels are used to represent operators on function spaces. The connection between kernels and operators on such function spaces is to be found in the concept of the Hilbert-Schmidt integral operator.

Definition 5 (Hilbert-Schmidt Operator)**.** *A Hilbert-Schmidt operator is a bounded operator $T : \mathcal{H} \to \mathcal{H}$ acting on a Hilbert space $\mathcal{H}$. On measure space $(\mathcal{X}, \mathcal{F}, \nu)$ with measure ν, the Hilbert-Schmidt operator associated with a kernel k is:*

$$T_k[f](x) = \int_{\mathcal{X}} k(x,y) f(y) d\nu(y) \tag{2.1}$$

Further to the discussion regarding the view of the Hilbert-Schmidt as operator on function spaces, such operators can be viewed to have eigenfunctions, which we define as follows:

Definition 6. *An eigenfunction of a Hilbert-Schmidt operator connected to a kernel k and measure space $(\mathcal{X}, \mathcal{F}, \nu)$ is a function ϕ such that:*

$$\int_{\mathcal{X}} k(x,x')\phi(x')\,d\nu(x') = \lambda\phi(x)$$

where λ is a scalar referred to as an eigenvalue of the operator.

Just as positive definite matrices can be diagonalised; that is, written as the composition of a diagonal matrix of eigenvalues, and a matrix formed of the eigenvectors, kernels can be written as the composition of a set of eigenvalues and these eigenfunctions, a result known as Mercer's theorem. This will be central in much of the work presented in this thesis.

Theorem 2 (Mercer Theorem (Mercer, 1909))**.** *Given a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, define the kernel operator*

$$T_k[f](x') = \int_{\mathcal{X}} f(x)k(x,x')d\nu(x) \tag{2.2}$$

for the measure ν. Consider a sequence $\{\phi_i\}$ of normalised eigenfunctions ϕ_i and eigenvalues $\lambda_i > 0$ of this operator:

$$\int \phi_i(x)k(x,x')d\nu(x) = \lambda_i\phi_i(x')$$

where the sequence of eigenvalues $\{\lambda_i\}_{i=1}^{\infty}$ is positive, non-increasing and absolutely summable. Then one can write the kernel function $k(x,x')$ in terms of these eigenfunctions and eigenvalues:

$$k(x,x') = \sum_{i=0}^{\infty} \lambda_i\phi_i(x)\phi_i(x') \tag{2.3}$$

The relation between Hilbert function spaces and kernels is to

be found in the concept of the reproducing kernel Hilbert space. To understand the relevance of the reproducing kernel Hilbert space, we first present the Riesz representation theorem.

Theorem 3 (Riesz Representation Theorem)**.** *Let $\mathcal{H}$ be a Hilbert space imbued with inner product $\langle x, y \rangle$. Denote its dual space (the space of linear functionals operating on $\mathcal{H}$) as $\mathcal{H}^*$. Then, for every continuous bounded linear functional $z \in \mathcal{H}^*$ there exists a unique element f_z of $\mathcal{H}$ such that:*

$$z(c) = \langle c, f_z \rangle \text{ , for } c \in \mathcal{H}.$$

That is, for any continuous linear functional defined on elements of the RKHS $\mathcal{H}$, there is an element f_z of $\mathcal{H}$ such that application of the functional z to an element c is equivalent to taking the inner product between the element f_z and c.

This motivates the reproducing kernel Hilbert space.

Definition 7 (Reproducing Kernel Hilbert Space)**.** *For some input space $\mathcal{X}$, a reproducing kernel Hilbert space $\mathcal{H}_k$ is a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$ such that the evaluation functional on $\mathcal{H}_k$ is a continuous linear functional.*

By the Riesz representation theorem (Theorem 3), for each $x \in \mathcal{X}$ there exists an element $k(\cdot, x) \in \mathcal{H}_k$ such that evaluation of a function f at x is

$$\langle f, k(.,x) \rangle = f(x).$$

Given the eigenrepresentation of the kernel (2.3) via eigenfunctions $\{\phi_i\}_{i=0}^{\infty}$ and eigenvalues $\{\lambda_i\}_{i=0}^{\infty}$, the reproducing kernel Hilbert space

is then the space of functions:

$$\mathcal{H}_k = \left\{ f : f(x) = \sum_{i=0}^{\infty} \alpha_i \phi_i(x); \sum_{i=0}^{\infty} \frac{\alpha_i^2}{\lambda_i} < \infty \right\}$$

with coefficients $\alpha_i \in \mathbb{R}$. If functions $f, g \in \mathcal{H}_k$ have respective sequences of coefficients $\{f_i\}_{i=0}^{\infty}, \{g_i\}_{i=0}^{\infty}$, then their reproducing kernel Hilbert space inner product is defined:

$$\langle f, g \rangle_{\mathcal{H}_k} = \sum_{i=0}^{+\infty} \frac{f_i g_i}{\lambda_i}.$$

The corresponding norm for the RKHS $\mathcal{H}_k$ is:

$$||f||_{\mathcal{H}_k} = \sum_{i=0}^{+\infty} \frac{f_i^2}{\lambda_i}.$$

2.3 Gaussian processes

Gaussian processes offer a flexible approach to modelling prior belief over functions. Given the above, we can now introduce the key use of the above concepts that will appear in the work in this thesis.

The Gaussian process is a stochastic process with particularly desirable properties.

Definition 8 (Gaussian Process)**.** *A Gaussian process is a stochastic process f, on an index space $\mathcal{X}$, such that any finite subset of values $\{f(x_1), f(x_2), ..., f(x_n)\}$ of f evaluated at a vector $\{x_1, x_2, ..., x_n\} \in \mathcal{X}^d$, has a joint Gaussian distribution. Denoting the index set $\mathcal{X} \subseteq \mathbb{R}^n$, and the*

output dimension o, define the functions

$$s : \mathcal{X} \to \mathbb{R}^o,$$
$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+,$$

where k is a positive definite kernel function. A Gaussian process is completely defined via its mean function s and covariance function k, by:

$$\mathbb{E}\left[f(x)\right] = s(x) \forall x \in \mathcal{X}$$
$$Cov\left[f(x), f(x')\right] = k\left(x, x'\right).$$

If f is a sample from a Gaussian process with mean function s and covariance function k, then we can write

$$f \sim \mathcal{GP}(s(\cdot), k(\cdot,\cdot)).$$

It is well known that the multivariate Gaussian distribution is fully defined by its mean vector and covariance matrix. The Gaussian process inherits this property in that it is fully defined by its mean function $s(\cdot)$ and the covariance function $k(\cdot,\cdot)$.

2.3.1 Gaussian process as prior belief

The Gaussian process is useful as a mechanism for expressing prior belief over function spaces (Rasmussen and C. K. I. Williams, 2018). Much of the preliminary information in this section comes from that book and the citations found therein. The choice of the mean function and the kernel in the Gaussian process allows a practitioner to express

specific prior belief about how the modelled function behaves.

The kernel regulates the differentiability class of the functions that the Gaussian process can represent (Reade, 1992). For example, the squared exponential kernel implies that a function drawn from the Gaussian process will be infinitely differentiable. The Matérn class, on the other hand, (Matern, 1960) allows one to control the differentiability class of the functions drawn from the Gaussian process by choice of its key parameter.

Examples of prior Gaussian process samples are presented in Figure 2.1.

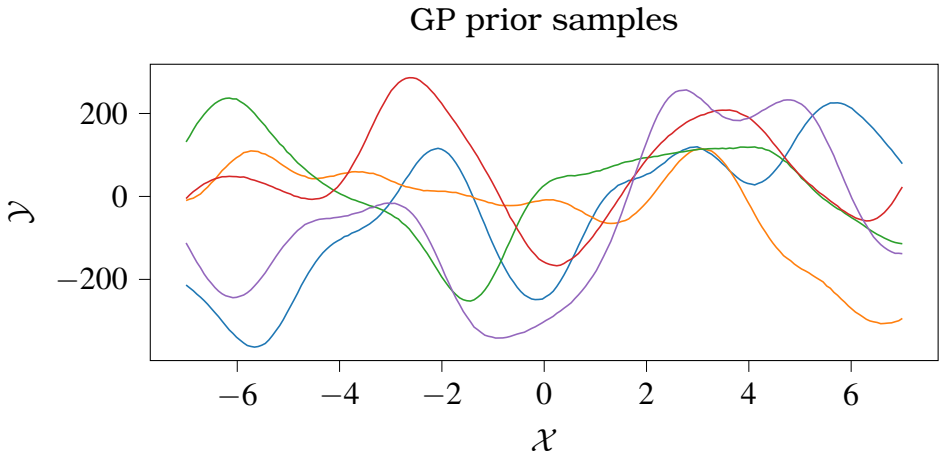

Figure 2.1: Samples from a prior Gaussian process under a squared exponential kernel with length-scale parameter 1.

Kernel functions can be described as either stationary or non-stationary. A stationary kernel $k(x,x')$ is one that is a function only of the $x-x'$, and not the value locations. A further refinement of stationary kernels are *isotropic* kernels, which are a function only of a distance between their inputs. The squared exponential kernel is an example of an isotropic stationary kernel.

A non-stationary kernel, however, cannot be expressed as a function of the distance between its inputs.

Another subclass of kernels is the set of degenerate kernels. A degenerate kernel is one whose Mercer representation is a finite sum of functions. Machine learning methods that attempt to construct approximate representations of kernels by a finite sum of basis functions yield a degenerate kernel. We highlight this distinction, because the role of degenerate kernels is pervasive in the work presented herein. A Gaussian process that does not use a degenerate kernel can be described as *infinite-dimensional*; this describes the size of the space of functions that the Gaussian process can represent, and is separate from the dimensionality of the input vector to the sample functions.

In general, it is not possible to truly generate samples from an infinite Gaussian process, and details on the standard process for sampling from Gaussian processes are outlined in Chapter 3. However, for finite-dimensional Gaussian processes, i.e. those connected to degenerate kernels, sampling can be achieved in a simple fashion if the Mercer representation of the kernel is available. This is a result of the Karhunen-Loève theorem.

Theorem 4 (Karhunen-Loève Theorem, (Karhunen, 1947))**.** *Suppose that $f(x)$ is a zero-mean square integrable stochastic process, with continuous covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, over a measure space $(\mathcal{X}, \mathcal{F}, \nu)$ for some measure ν. Then, $k(\cdot,\cdot)$ is a Mercer kernel. Denote by $\{\phi_i\}_{i=0}^{\infty}$ the eigenfunctions, and by $\{\lambda_i\}_{i=0}^{\infty}$ the eigenvalues of the corresponding Hilbert-Schmidt operator $T_k[\cdot]$ (see* (2.1)*). Then there exist random variables $\{\theta_i\}_{i=0}^{\infty}$ such that:*

$$f(x) = \sum_{i=0}^{\infty} \theta_i \phi_i(x) \tag{2.4}$$

where the random variables $\{\theta_i\}_{i=0}^{\infty}$ are uncorrelated, and θ_i has vari-

ance λ_i.

In Theorem 4 no reference is made to the degeneracy of the kernel. However, if the kernel is degenerate, it is simple to generate a Gaussian process sample as $f(x) = \sum_{i=0}^{m} \theta_i \phi_i(x)$. where $\theta_i \sim \mathcal{N}(0, \lambda_i)$. It is easy to see that this yields the correct covariance:

$$\begin{aligned}
\mathbb{E}\left[f(x)f(x')\right] &= \mathbb{E}\left[\sum_{i=0}^{m}\sum_{j=0}^{m}\theta_i\theta_j\phi_i(x)\phi_j(x')\right] \\
&= \sum_{i=0}^{m}\sum_{j=0}^{m}\mathbb{E}\left[\theta_i\theta_j\right]\phi_i(x)\phi_j(x') \\
&= \sum_{i=0}^{m}\sum_{j=0}^{m}\mathbb{E}\left[\theta_i^2\right]\delta_{ij}\phi_i(x)\phi_j(x') \qquad (2.5) \\
&= \sum_{i=0}^{m}\lambda_i\phi_i(x)\phi_i(x') \\
&= k(x,x')
\end{aligned}$$

where (2.5) follows from the independence of θ_i, θ_j.

2.3.2 Gaussian process as posterior belief

Updating from the Gaussian process prior to the posterior based on observations yields what is referred to as the posterior Gaussian process. Intuitively, the update formulae have the effect of restricting the samples generated from the Gaussian process to be consistent with the observations made. To appreciate this we first describe a standard sampling regime. We assume a measure space $(\mathcal{X}, \mathcal{F}, \nu)$, and that the practitioner has access to a set of observations $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$, where the input values $\{x_i\}_{i=1}^{N}$ are sampled from $\mathcal{X}$ according to ν. The output values y_i are assumed to be generated as the output of some function f at the input points x_i corrupted by Gaussian noise

$\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$:

$$\mathbf{y} = f(\mathbf{x}) + \varepsilon$$

where ε is a vector of Normally distributed random variables.

Given a kernel k and assuming a mean function constant at zero, the posterior Gaussian process sample has mean function, evaluated at test points x^*:

$$s(x^*)|\mathcal{D} = k(x^*, X)\left(K(X,X) + \sigma^2 I\right)^{-1}\mathbf{y} \tag{2.6}$$

and covariance function:

$$K(x^*)|\mathcal{D} = k(x^*, x^*) - k(x^*, X)\left(K(X,X) + \sigma^2 I\right)^{-1}k(X, x^*) \tag{2.7}$$

The above formulae, when used to generate posterior Gaussian process sample, lead to sample functions as in Figure 2.2.

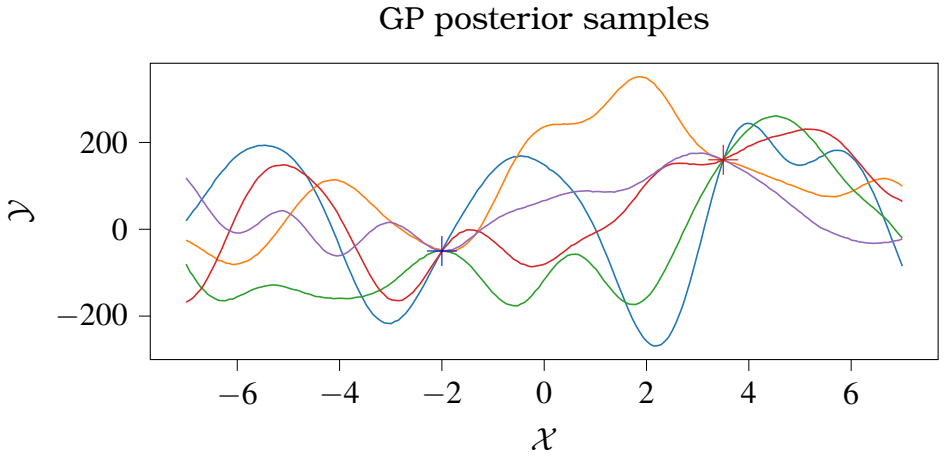

Figure 2.2: Samples from a posterior Gaussian process with squared exponential kernel with length-scale parameter 1, conditioned on observations at $x = -2$ and $x = 3.5$.

2.3.3 Inference in Gaussian process models

Inference in Gaussian process models essentially refers to the method by which one chooses the hyperparameters of the kernel. Since the choice of the kernel affects the behaviour of the functions represented by the Gaussian process, the choice of hyperparameters is crucial to the performance of the Gaussian process model.

Such inference is typically performed using the marginal likelihood function. In order to use the full likelihood function, one must take into account the inputs, the observations, and the latent values of the Gaussian process function f. These values are unknown, and so one must integrate out the latent values f (Rasmussen and C. K. I. Williams, 2018). This yields the marginal likelihood function:

$$-\frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln \mid K + \sigma^2 I_N \mid -\frac{1}{2}\mathbf{y}'\left(K + \sigma^2\right)^{-1}\mathbf{y} \tag{2.8}$$

which depends only on the observed values.

The standard approach to selection of kernel hyperparameters is to maximise the marginal likelihood function with respect to the kernel hyperparameters. This is typically done using gradient-based optimisation methods. Details of construction of the gradient of this function can be found in Rasmussen and C. K. I. Williams (2018).

2.4 Orthogonal Polynomials

Another topic recurrent in the work in this thesis is that of orthogonal polynomials. The standard introduction to orthogonal polynomials is given in Chihara (2011) and sources therein. In many applications it is necessary or desirable to maintain an orthonormal basis of functions. This is particularly true in the context of Gaussian processes, where

the Mercer representation of the kernel requires an orthonormal sequence of eigenfunctions.

Chapter 3 will illustrate a connection between orthogonal polynomials and Gaussian processes, wherein we develop a method for generalising the construction of Mercer kernels by constructing basis functions from orthogonal polynomials.

In Chapter 4, we present a use of orthogonal polynomials to construct embeddings of graph structure, and an application of this approach to embedding dynamic sequences of graphs.

Orthogonal polynomials also see much use in random matrix theory. Distributions over random matrices are usually confined to spaces of unitarily equivalent matrices; that is, matrices that differ only by an orthogonal transformation. Distributions over such matrices are said to be invariant under unitary transformations, and a specific distribution over eigenvaluess described as a *unitary ensemble* (Deift, 2000). In that setting, the k-point correlation function of the point process describing the eigenvalues of the matrix is given by the determinant of a Gram matrix of a kernel formed using the orthogonal polynomials associated with the given distribution. This is known as the *Christoffel-Darboux kernel* (Tao, 2012).

The Christoffel-Darboux kernel also plays a role in approximation theory (Lasserre, Pauwels, and Putinar, 2022). Data from a given distribution can be used to calculate sequences of orthogonal polynomials. Conversely, the orthogonal polynomials can be used to construct the Christoffel-Darboux kernel, which provides information about the measure of orthogonality. This connection will be used extensively in Chapter 4.

We begin with the concept of the linear moment functional, the

standard point of departure for an analysis of orthogonal polynomials. Whilst in most cases, such polynomials will be defined with respect to a distribution (or at least, a general measure on a measure space), they technically only require the definition of a functional mapping polynomials to real numbers. We mostly follow the notation used by Chihara (2011).

Definition 9 (Linear Moment Functional)**.** *Define $\mathcal{L}$ to be a function on the vector space of polynomials $\mathcal{P}$. We write application of $\mathcal{L}$ to a polynomial $P(x) \in \mathcal{P}$ as:*

$$\langle \mathcal{L}, P(x) \rangle$$

For a sequence $\mu = \{\mu_n\}_{n=0}^{\infty}$, $\mathcal{L}$ is called a linear moment functional *if it has the following two properties:*

- *$\mathcal{L}$ maps the monomials to the sequence μ; i.e. $\langle \mathcal{L}, x^j \rangle = \mu_j$.*
- *$\mathcal{L}$ is linear; i.e. $\langle \mathcal{L}, aP_1(x) + bP_2(x) \rangle = a \langle \mathcal{L}, P_1(x) \rangle + b \langle \mathcal{L}, P_2(x) \rangle$.*

A linear moment functional can be characterised as *positive definite*, depending on its behaviour when applied to everywhere-positive polynomials (Chihara, 2011). However, to avoid unnecessary intermediate definitions we define positive definiteness of linear moment functions directly in terms of the Hankel determinants of appropriately defined moment matrices.

Definition 10 (Positive Definite Linear Moment Functional)**.** *Define a linear moment functional L to be a mapping from the space of polynomials $\mathcal{P}$ to complex numbers: $\mathcal{L} : \mathcal{P} \to \mathbb{C}$. Such a linear moment functional is uniquely defined by the system $\mathcal{L}[x^j] = \mu_j$, for some sequence $\{\mu_j\}_{i=0}^{\infty}$. A*

linear moment functional $\mathcal{L}$ with moment sequence $\mu = \{\mu_i\}_{i=0}^{\infty}$ is called positive definite if the following inequality holds for all i:

$$H_n \equiv \begin{vmatrix} \mu_0 & \mu_1 & \dots & \mu_i \\ \mu_1 & \mu_2 & \dots & \mu_{i+1} \\ \mu_2 & \mu_3 & \dots & \mu_{i+2} \\ \dots & \dots & \dots & \dots \\ \mu_i & \mu_{i+1} & \dots & \mu_{2i} \end{vmatrix} > 0 \; \forall \; i \tag{2.9}$$

The linear moment functional construction however can be developed by representing the functional as a Stieltjes integral. That is, we can equate linear moment functionals with integrals like:

$$\left\langle \mathcal{L}, x^k \right\rangle = \int_{\mathbb{R}} x^k d\nu(x) \tag{2.10}$$

for an appropriate measure ν. This is a result of the representation theorem for moment functionals (Chihara, 2011).

Given the concept of the linear moment functional, the notion of an inner product on the space of polynomials can be defined. This is how we can consider the concept of the orthogonal polynomial sequence; without a concept of inner product on polynomials, such a concept would not be feasible. The linear moment functional concept does not require in and of itself an integral representation. As shown by Chihara (2011), this integral representation is a result of the representation theorem for moment functionals, which shows that for a given linear moment functional as defined above, there exists an appropriate distribution function ν that yields the integral representation as in 2.10. However, construction of an inner product on polynomials is valid without this integral representation via a

purely algebraic consideration of the ring of polynomials (Lang, 2002). Specifically, it is valid to write the inner product of two polynomials as: $\langle \mathcal{L}, P_i P_j \rangle$ since the product of two polynomials is again a polynomial. Thus, it is possible to construct an inner product on spaces of polynomials using a linear moment functional as defined above without yet needing to consider integrals or measures.

We can thus define the orthogonal polynomial sequence:

Definition 11 (Orthogonal Polynomial Sequence)**.** *Suppose a moment linear functional $\mathcal{L}$. An Orthogonal Polynomial Sequence (OPS) is a sequence of polynomials $\{P_n\}$ s.t.: for all integers n, m:*

- $P_n(x)$ *is a polynomial of degree n,*
- $\langle \mathcal{L}, P_n(x) P_m(x) \rangle = 0, n \neq m$
- $\langle \mathcal{L}, P_n^2(x) \rangle \neq 0.$

and its normalised counterpart:

Definition 12 (Orthonormal Polynomial Sequence)**.** *Given a linear moment functional $\mathcal{L}$, an orthonormal polynomial sequence $\{P_n\}$ is a polynomial sequence such that*

- $\{P_n\}$ *is an OPS;*
- $\langle \mathcal{L}, P_n^2(x) \rangle = 1.$

A key property of orthogonal polynomials is the three-term recurrence (Chihara, 2011; Ding and Trogdon, 2021).

Theorem 5 (Orthogonal Polynomial Recurrence (Chihara, 2011))**.** *Let $\mathcal{L}$ be a linear moment functional with corresponding moment sequence $\{\mu_i\}_{i=0}$. Let $\{P_n(x)\}_{n=0}^{\infty}$ be an OPS with respect to $\mathcal{L}$. Then there exist*

sequences of coefficients, $\{\beta_n\}_{n=0}$, $\beta_n \in \mathbb{R}$, *and* $\{\gamma_n\}_{n=0}$, $\gamma_n \in \mathbb{R}^+$ *such that the following three-term recurrence holds for* $\{P_n(x)\}_{n=0}^{\infty}$,

$$\begin{aligned} P_{-1}(x) &= 0, \\ P_0(x) &= 1, \\ P_n(x) &= (x-\beta_n)P_{n-1}(x) - (\gamma_n)P_{n-2}(x). \end{aligned} \tag{2.11}$$

The inverse of this property is also true, and is known as *Favard's Theorem* (Chihara, 2011).

Theorem 6 (Favard's Theorem (Favard, 1935))**.** *Let* $\{\beta_n\}_{n=0}$ *be an arbitrary real sequence, and* $\{\gamma_n\}_{n=0}$ *be a sequence of positive real numbers. Let* $\{P_n(x)\}_{n=0}^{\infty}$ *be a polynomial sequence such that* $P_0(x) = 1$, *and following the recurrence:*

$$P_n(x) = (x-\beta_n)P_{n-1}(x) - \gamma_n P_{n-2}(x).$$

where we write $P_{-1}(x) = 0$ *so the recurrence holds for all* n. *Then* $\{P_n(x)\}_{n=0}^{\infty}$ *is an orthogonal polynomial sequence (OPS); and there is a unique moment functional* $\mathcal{L}$ *s.t.* $\langle \mathcal{L}, 1\rangle = \gamma_1$ *and* $\langle \mathcal{L}, P_j(x)P_k(x)\rangle = B_j\delta_{jk}$ *for some constant* B_j *depending on the order of the polynomial.*

This implies that the space of orthogonal polynomials is dense, in the sense that perturbations to the recurrence coefficients lead to a sequence of polynomials that is still orthogonal with respect to some measure. Understanding the effect of perturbation of the recurrence coefficients on the measure of orthogonality is a topic of active research (Ding and Trogdon, 2021).

This result will be used thoroughly in the present thesis. In Chapter 3 it will allow us to construct, given information on a distribution

(or measure) ν, a sequence of orthonormal basis functions. In Chapter 4, it will allow us to consider specific sequences of polynomials defined to represent properties of graphs. This will allow us to extract structural information about sequences of graphs for a general graph feature construction method, and detect anomalous graphs as a result.

Chapter 3

Favard Kernels for Sparse Gaussian Process Models

3.1 Introduction

The Gaussian process models presented in Chapter 2 are a powerful tool for conducting both classification and regression problems (Rasmussen and C. K. I. Williams, 2018), as well as more exotic applications such as in modelling robotic dynamics (Deisenroth and Rasmussen, 2011), normalising flows for generative modelling (Maroñas et al., 2021), reinforcement learning applications (Strens, 2000; Fan, Chen, and Wang, 2018), and Bayesian optimisation (Brochu, Cora, and Freitas, 2010).

As noted in the preliminary section, the standard approach to inference over the hyperparameters of the kernel function is achieved by optimising the marginal likelihood of the Gaussian process model. This relies on the conditioning property of Gaussian processes. This is the fact that conditioning only on the observations, i.e. a finite subset of the possible inputs to the unknown function, inference is the same as if one had taken the rest of the unobserved function into

account.

The optimisation of the marginal likelihood (2.8) is however a non-trivial task in big data settings. It requires the evaluation, at each optimisation step, of both the inverse and the determinant of the covariance matrix of the Gaussian process, which is of dimension $N \times N$, where N is the number of data points. Both these operations are of complexity $\mathcal{O}\left(N^3\right)$, which is prohibitive in settings where large sample sizes are available.

Furthermore, the generation of posterior samples carries a similar complexity, given that the standard approach requires generation of the Cholesky decomposition of the covariance Gram matrix, evaluated at the datapoints. The result is that sparsification techniques for Gaussian process models are highly desirable.

Classical approaches to the problem of speeding up Gaussian process inference include variational methods, such as (Titsias, 2009), which constructs a variational approximation of the posterior, and selects in practice an appropriate subset of the observed data to be used as inducing points. The resulting model is then optimised using the variational lower bound on the marginal likelihood.

The concept of selection of inducing points in the domain of the Gaussian process is extended to the spectral domain by the Variational Fourier Features method (Hensman, Durrande, and Solin, 2018). The approach here projects the Gaussian process onto a windowed Fourier basis.

This manages to speed up inference by lowering the rank of the corresponding covariance matrix; if the number of inducing points $m << N$, then the high computational complexity of the optimisation of the Gaussian process marginal likelihood can be avoided. The

complexity of this method is on the order of $\mathcal{O}\left(m^2N\right)$ for initial computation, and then $\mathcal{O}\left(m^3\right)$. This is equivalent to the method outlined in this chapter.

Another approach (Rahimi and Recht, 2007) follows the observation noted above that the kernel represents an inner product on an appropriately defined space. The Random Fourier Feature technique uses randomised sampling and Bochner's theorem (Puckette and Rudin, 1965) to generate "Fourier" features, whose inner product is *in expectation* equal to the kernel. The method achieves good approximation performance (Hoang et al., 2020), but appears to require large feature counts e.g. $m \approx 5000$ to achieve good performance (Rahimi and Recht, 2007).

Methods that utilise the spectral properties of the kernel, both as a matrix and an operator, have also been considered. The idea is to use projections of the kernel onto a finite dimensional basis (Trecate, C. K. Williams, and Opper, 1999). Such methods can often achieve complexity linear in the sample size (Solin and Särkkä, 2020; Daskalakis, Dellaportas, and Panos, 2022). Another approach is to use the Nyström method (Girolami, 2002; Rasmussen and C. K. I. Williams, 2018), which attempts to construct approximately orthogonal features by noting that the Hilbert-Schmidt operator formulation (see Section 2.1) can be thought of as an appropriately defined expectation. The method generates approximate features by sampling from the appropriate input density ν, and then generating vector evaluations from the orthogonal basis that would form the Mercer eigenfunction expansion of the kernel under ν. However, as noted by Flaxman, Teh, and Sejdinovic (2017), the result differs whether one includes the observed data points in the constructed basis or not,

and this has to be taken into account when utilising that method.

The quintessential spectral approach is, where available, to utilise the Mercer eigenfunction expansion of the kernel (Zhu et al., 1998; Fasshauer, 2012a). If the eigenfunction expansion is available, then the properties of the Gaussian process modelled using such a kernel approximation are state-of-the-art (Braun, 2006; Daskalakis, Dellaportas, and Panos, 2022). In general this decomposition is not available for arbitrary kernels and measures, as they rely on the solutions to difficult integral equations.

One approach to solve this problem (Daskalakis, Dellaportas, and Panos, 2022) is to utilise a neural network training phase that learns the mapping from the input space to a space on which the inputs are Gaussian-distributed; however, this training phase likely induces complexity great enough that the sparsity gains are lost. More recent approaches (Cunningham et al., 2023) essentially ignore the problem of orthonormality of the features, and instead use B-splines to construct features that can be used to construct the Gaussian process samples, to achieve sparse kernel matrices.

In this chapter we present a method that utilises a Mercer kernel construction with focus on the necessity that the basis functions be orthonormal with respect to the input distribution. The motivation for this is to be found in Theorem 7. This theorem shows that a necessary and sufficient condition for the eigenvalues of the kernel matrix to be consistent with the eigenvalues of the covariance operator of the Gaussian process is that the basis functions be orthonormal with respect to the input distribution. Previous work has only yielded the sufficiency of this condition (see Braun, 2006).

As we note in Section 3.3, the importance of the orthonormality

of the projected basis has been largely ignored in the literature. The only work that appears to have attempted to look at this in depth is by C. K. I. Williams and Seeger (2000), in the context of SVM classification models. In the literature on Gaussian process sparsification, there is much attention paid to the convergence of the operators that are being approximated, but little attention paid to the convergence of the corresponding finite-dimensional representations of those operators. The work in this chapter aims to address this gap in the literature.

3.2 Related Work

The approach to GP sparsification presented by Solin and Särkkä (2020) views the kernel as a pseudo-differential operator. This operator can be written as a formal series of Laplace operators, and the kernel can be approximated using the eigenfunctions of these Laplace operators on some given space. These eigenfunctions are in essence orthogonal with respect to a uniform input measure. The authors also note that it is possible to consider the inner products in terms of an input density, stating that the approximation error in the technique they propose will be small if the input measure is close to constant in the region of the data. Difficulties in constructing the eigenfunctions of the weighted linear operator implied by an input density in that manner mean that we expect our method to prove useful in such situations. Our method does not require explicit statement or consideration of boundary conditions, nor requires that the input density be close to uniform around the data. Furthermore, the scalar version of their approach is connected to the eigenfunctions of a Sturm-Liouville operator. However, as has been known for almost a century (Bochner, 1929), the only polynomial solutions to this differential equation are

the classical orthogonal polynomials. Our approach essentially generalises that technique by allowing for (quasi-)polynomial functions that are orthogonal with respect to more exotic measures than those corresponding to the classical orthogonal polynomials, and therefore we consider our method to sit complementary to the Hilbert-space methods described by Solin and Särkkä (2020).

Recent work has also achieved $\mathcal{O}\left(m^3\right)$ (see Cunningham et al., 2023) which construct the basis functions as a set of B-splines. Their approach however suffers from problems in high-dimensions, and currently only approximates Matérn kernels. Since our approach can include representations of Matérn kernels, the squared exponential kernel, and non-stationary kernels, we believe that our approach can be considered competitive to theirs. We also explicitly take into account the orthogonality of the basis functions with respect to the input measure, which avoids divergence of the matrix eigenvalues from the operator eigenvalues (see Theorem 7).

3.3 Motivation

We assume a measure space $(\mathcal{X}, \mathcal{F}, \nu)$; and an output space $\mathcal{Y}$. The measure ν describes the distribution from which inputs $\mathbf{x} \in \mathcal{X}$ are drawn, and the practitioner is provided with a joint sample $(\mathbf{x}, \mathbf{y}) \in \mathcal{X}^N \times \mathcal{Y}^N$. where N is the sample size. The output vector is assumed to have been generated as $\mathbf{y} = \hat{f}(\mathbf{x}) + \varepsilon$, where $\hat{f} : \mathcal{X} \to \mathcal{Y}$ is an unknown function, and $\varepsilon \in \mathcal{Y}^N$ is a zero-mean Gaussian distributed noise vector. It is assumed that the practitioner aims to construct a Bayesian model over the unknown function, in the form of a Gaussian process.

3.3.1 Gaussian processes

Assume a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, and a mean function $s : \mathcal{X} \to \mathbb{R}$. We can denote that a random function is generated as a sample from a Gaussian process by writing: $f \sim \mathcal{GP}(s(\cdot), k(\cdot,\cdot))$.

3.3.1.1 Applying Mercer's theorem

For a given kernel k and input measure ν, by Mercer's theorem (Theorem 2), (Mercer, 1909), the kernel can be written

$$k(x,x') = \sum_{i=0}^{\infty} \lambda_i \phi_i(x) \phi_i(x') \tag{3.1}$$

where the functions $\phi_i : \mathcal{X} \to \mathbb{R}$ are an orthonormal sequence with respect to ν. Depending on the measure, and specifically depending on the properties of the decay of its moments, this sequence of functions may constitute a basis in $\mathcal{L}^2(\nu)$ (Deift, 2000; Chihara, 2011). The eigenvalues λ_i are non-negative and decreasing in i. If the representation in terms of orthonormal basis functions is available, then an approximate Gaussian process likelihood can be written as:

$$\begin{aligned} \mathcal{L}(\mathbf{x},\mathbf{y}) = &-\frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln \mid \Phi\Lambda\Phi' + \sigma^2 I_N \mid \\ &-\frac{1}{2}\mathbf{y}'\left(\Phi\Lambda\Phi' + \sigma^2\right)^{-1}\mathbf{y} \end{aligned} \tag{3.2}$$

where Φ is an $N \times m$ matrix of basis functions evaluated at the inputs; Λ is a diagonal matrix comprising the perator eigenvalues $\{\lambda_i\}_{i=0}^{m}$; σ^2 is the variance parameter of the noise variable ε; I_N is an $N \times N$ identity matrix. The idea is that we can approximate the Gaussian process kernel using a finite sum of m basis functions, as the decay of the eigenvalues λ_i will allow us to truncate the sum in Equation 3.1 to a finite sum. When the marginal likelihood written in this form, it is then

possible to speed up the evaluation of the Gaussian process likelihood via the Woodbury-Sherman-Morrison (WSM) formulae (Rasmussen and C. K. I. Williams, 2018):

$$|\Phi\Lambda\Phi' + \sigma^2 I_N| = \sigma^{2N}|\Lambda||\Lambda^{-1} + \sigma^{-2}\Phi'\Phi| \tag{3.3}$$

$$(\Phi\Lambda\Phi' + \sigma^2 I_N)^{-1} = \sigma^{-2} I_N - \sigma^{-2}\Phi\left(\sigma^2\Lambda^{-1} + \Phi'\Phi\right)^{-1}\Phi'. \tag{3.4}$$

Note that the inverse and determinant terms in the above formulae are of complexity $\mathcal{O}\left(m^3\right)$, where m is the number of basis functions used in the degenerate Mercer decomposition approximation.

This is the key to the approach taken to several sparse Gaussian process models (Wilson et al., 2020; Cunningham et al., 2023; Daskalakis, Dellaportas, and Panos, 2022). Usage of the WSM formulae allows the likelihood to be evaluated in $\mathcal{O}\left(mN^2\right)$ time. However, in the method presented in the current chapter, and those by Hensman, Durrande, and Solin (2018) and Cunningham et al. (2023), the basis functions and therefore the matrix $\Phi'\Phi$ are calculated beforehand which allows us to achieve $\mathcal{O}\left(m^3\right)$ time complexity in the repeated likelihood evaluation step. This yields large gains when $m << N$.

However, in many cases these eigenfunctions are not available. The reason for this is that the integral equation that defines the eigenfunctions ϕ_i is often not solvable analytically. Classic examples are the case of the squared exponential kernel under Gaussian-distributed inputs (Zhu et al., 1998; Fasshauer, 2012b), and the Matérn kernel under uniform inputs (Daskalakis, Dellaportas, and Panos, 2022).

3.3.1.2 Classic example

To aid the exposition, we present the classic example of the squared exponential kernel under Gaussian-distributed inputs. The standard smooth exponential kernel with $l > 0$ and $\sigma^2 > 0$ is defined as

$$k\left(x,x'\right) = \sigma^2 \exp\left(-\frac{||x-x'||^2}{2l^2}\right)$$

where $||\cdot||$ is the Euclidean norm. Under the assumption of Gaussian input measure with mean 0, with precision $\alpha > 0$, the corresponding decomposition of the kernel is given by (Zhu et al., 1998)

$$\phi_i(x) = d_i H_i(\sqrt{2c}x)\exp\left\{-(c-\alpha)^2\right\}$$

where H_i represents the i-th Hermite polynomial, d_i is a normalising coefficient, $c = \sqrt{\alpha^2 + 2\alpha l}$ and these basis functions are orthonormal with respect to the zero-mean Gaussian distribution with precision α.

As noted above, in order to utilise the advantages of the Mercer decomposition for a given kernel it is necessary to calculate the eigenfunctions and eigenvalues $\{\phi_i, \lambda_i\}_{i=0}$ that fulfill the following equation:

$$T_k\left[\phi_i\right](x') := \int_{\mathcal{X}} k\left(x,x'\right)\phi_i(x)d\nu(x) = \lambda_i\phi_i(x').$$

Because of the difficulty of this integral for arbitrary measures, it might seem natural to ignore the dependence on the input measure and simply use the standard Gaussian input Mercer decomposition, without concern for the input distribution. We will show that this method is not valid and will lead to non-representative learnt distributions.

We also consider a more conceptual aspect of Gaussian process models. The Gaussian process is a prior over functions, and the associated covariance function describes essentially an operator on functions (via (2.1)). The usefulness of the Gaussian process is based on the conditioning property of the Gaussian distribution (Rasmussen and C. K. I. Williams, 2018). Specfically, the Gaussian process represents an infinite-dimensional distribution on function evaluations, but conditioning on a finite number of points yields a finite-dimensional Gaussian distribution. This is the key to the computational tractability of Gaussian process models, as we can essentially ignore the infinite-dimensional nature of the process and reach the same conclusions by conditioning on the observed values.

In this way, learning the parameters of the Gaussian process is equivalent to learning the operator on function spaces represented by the kernel. This operator is represented in finite dimensions by using a matrix. Naturally, one would expect that the covariance *matrix* should converge in some sense to the same operator as the covariance *function*. It has been shown elsewhere (Braun, 2006) that a sufficient condition for such a convergence is that the eigenfunctions of the covariance operator are orthonormal with respect to the input distribution. However, we now present an extension to this theorem showing that orthonormality is a necessary and sufficient condition for such convergence.

Theorem 7 (Eigenvalue Consistency)**.** *Suppose* $(\mathcal{X}, \mathcal{F}, \nu)$ *a measure space with* ν *absolutely continuous with respect to Lebesgue measure. Assume we can sample from* $\mathcal{X}$*, to generate* $\mathbf{x} = \{x_i\}_{i=0}^{N}$*. Let* $\{\phi_i\}_{i=0}^{m}$ *be a sequence of functions* $\phi_i : \mathcal{X} \to \mathbb{R}$ *Construct the matrix* $K = \frac{1}{N}\Phi\Lambda\Phi'$

where Λ is a diagonal matrix with $\Lambda_{ii} = \lambda_i$, and $\Phi_{ij} = \phi_j(x_i)$. The i-th decreasingly ordered eigenvalue of a matrix A is denoted $\lambda_i(A)$. Then, as $N \to \infty$, and for $i \in \{1,2,...,m\}$,

$$\frac{|\lambda_i - \lambda_i(K)|}{\lambda_i} \to_{a.s.} 0$$

if and only if $\{\phi_i\}_{i=0}^m$ are orthonormal w.r.t ν, and where by $\to_{a.s.}$ we denote almost sure convergence.

Proof. Proof in Appendix B. □

This theorem essentially states that the use of incorrect orthonormal basis functions decouples the Gaussian process model from the underlying operator on function spaces. In Figure 3.1, we present empirically the effect of an incorrect input distribution on estimators of the kernel eigenvalues. Estimators for the kernel eigenvalues can be constructed by scaling the first m eigenvalues of the kernel matrix by $\frac{1}{N}$.

The results in Figure 3.1 show that the (scaled) finite dimensional representation of the kernel as an operator will not converge to the true operator if the basis functions are not orthonormal. This will be especially egregious when it comes to evaluate models by comparison of the covariance matrix; distance measures between Normal distributions such as the Kullback-Leibler divergence will produce incorrect or misleading values when applied to comparisons between Gaussian processes.

To clarify this point, note that convergence in probability of the covariance matrix eigenvalues implies convergence of both its trace and its determinant. The Kullback-Leibler divergence between two

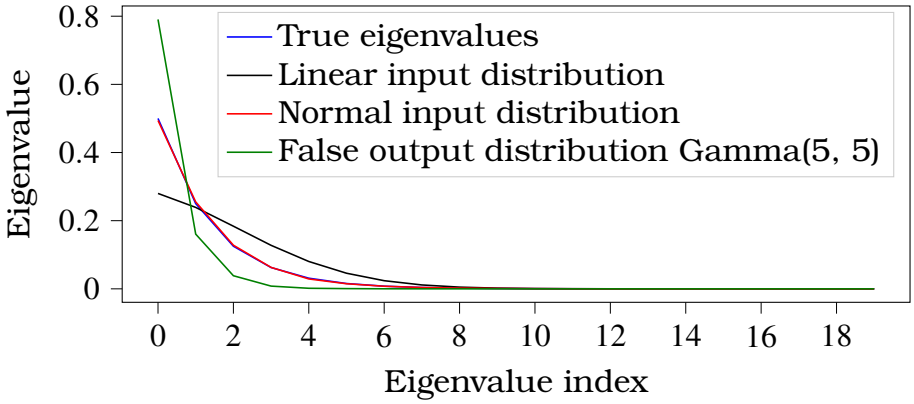

Figure 3.1: Eigenvalue consistency failure in non-orthonormal basis functions. The first 20 eigenvalues of $\frac{1}{N}\Phi'\Lambda\Phi$ evaluated at a sample of size 1000 under different input distributions, where Φ is the "Fasshauer" basis, orthogonal with respect to a Normal distribution. The red line represents the eigenvalue estimates under the correct input distribution, which is $\mathcal{N}(0,0.25)$; the blue line is the correct set of eigenvalues. The black line ("Linear input distribution") shows the eigenvalues of the kernel matrix evaluated at evenly spaced samples, and the green line is calculated under a sample distributed according to a Gamma distribution.

$d-$dimensional multivariate Normal distributions with the same mean is given by

$$D_{KL}(\mathcal{N}_1 \,||\, \mathcal{N}_2) = \frac{1}{2}\left(\mathrm{Tr}K_{\mathcal{N}_2}^{-1}K_{\mathcal{N}_1} - d + \log\frac{\det K_{\mathcal{N}_2}}{\det K_{\mathcal{N}_1}}\right).$$

where TrA denotes the trace of a matrix A; det$A$ denotes the determinant of a matrix A; and $K_{\mathcal{N}_i}$ denotes the covariance matrix of a multivariate Normal distribution. The terms in this expression depend directly on the eigenvalues of the matrices $K_{\mathcal{N}_1}$ and $K_{\mathcal{N}_2}$. Comparison between Gaussian processes, by comparing the finite-dimensional covariance matrices using the Kullback-Leibler divergence will be misleading if the eigenvalues of the covariance matrix do not converge to the eigenvalues of the covariance operator.

In Figure 3.2, we present the effect on the KL-divergence between

two d-dimensional Gaussian random variables; one with covariance matrix approximated by the Mercer approximation, and one with the full kernel. In this case the Mercer approximation is constructed using the correct eigenfunctions. In Figure 3.3, we present the effect on the KL-divergence between two Gaussian random variables; one with covariance matrix approximated by the Mercer approximation, and one with the full kernel, where now the Mercer approximation matrix uses inputs from a $Gamma(5,5)$ distribution. This clarifies that, despite attempting to approximate the same operator, the Mercer approximation using incorrect eigenfunctions can lead to finite-dimensional representations that are very different. This will need to be taken account in comparison of any sparse Gaussian process models.

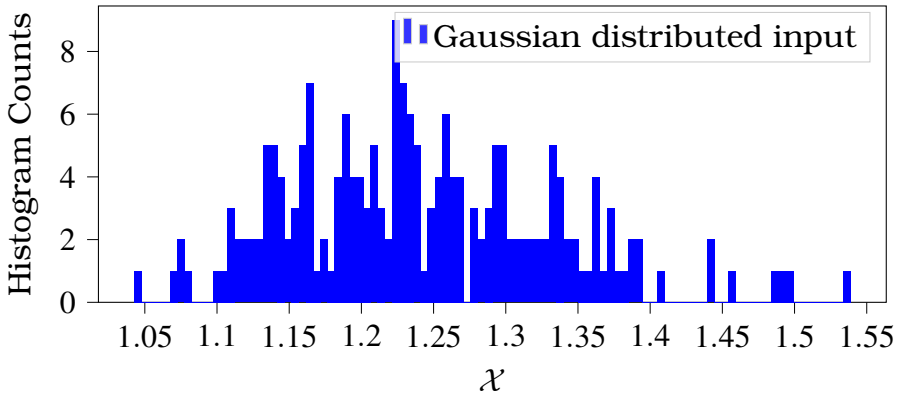

Figure 3.2: KL divergences of the approximate kernel matrix from the true kernel matrix under a Gaussian input distribution. The approximate kernel matrix is calculated using the "Fasshauer" basis with 20 basis functions. The true kernel matrix is calculated using the Gaussian kernel with length-scale 1.0. We generate 1000 Gaussian-distributed input samples, and calculate the KL divergence from the approximate to the true kernel matrices for each of these samples. The resulting KL divergences are plotted as a histogram. The input distribution is the correct distribution ($\mathcal{N}(0,0.25)$).

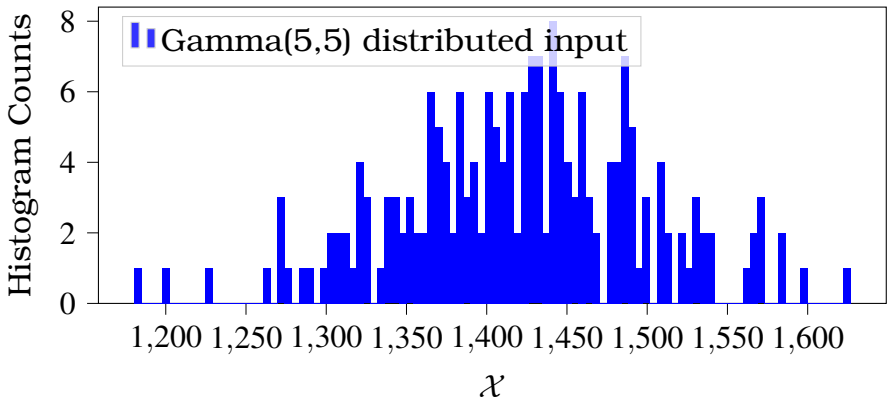

Figure 3.3: KL divergence of the approximate kernel matrix from the true kernel matrix under a Gamma$(5,5)$ input distribution. The approximate kernel matrix is calculated using the "Fasshauer" basis with 20 basis functions. The true kernel matrix is calculated using the Gaussian kernel with length-scale 1.0. We generate 1000 Gamma-distributed input samples, and calculate the KL divergence from the approximate to the true kernel matrices for each of these samples. The resulting KL divergences are plotted as a histogram. The input distribution is Gamma$(5,5)$.

3.4 Method

We now present a method for achieving fast Gaussian process training whilst capturing orthonormality of the basis functions. We begin with the definition of a *truncated* reproducing kernel Hilbert space (RKHS).

Definition 13 (Truncated Reproducing Kernel Hilbert Space)**.** *For some input space $\mathcal{X} \subset \mathbb{R}$, and a measure ν on $\mathcal{X}$, define a sequence of functions $\phi_i : \mathcal{X} \to \mathbb{R}$, such that $\{\phi_i\}_{i=0}^m$ are orthonormal with respect to ν. Define $\lambda = \{\lambda_i\}_{i=0}^m$ to be a strictly positive, decreasing sequence of real numbers. A truncated reproducing kernel Hilbert space $\mathcal{H}_k^m$ is a reproducing kernel Hilbert space (RKHS) comprising the space of functions $f : \mathcal{X} \to \mathbb{R}$:*

$$\mathcal{H}_k^m = \left\{ f : f(x) = \sum_{i=0}^{m} \alpha_i \phi_i(x) \right\}$$

with coefficients $\alpha_i \in \mathbb{R}$. We call m the order of $\mathcal{H}_k^m$. The truncated reproducing kernel Hilbert space inner product between functions $f, g \in \mathcal{H}_k^m$, with respective sequences of coefficients $\{f_i\}_{i=0}^m, \{g_i\}_{i=0}^m$, is defined as:

$$\langle f, g \rangle_{\mathcal{H}_k^m} = \sum_{i=0}^{m} \frac{f_i g_i}{\lambda_i}.$$

Given the measure space $(\mathcal{X}, \mathcal{F}, \nu)$, and the kernel k, denote the corresponding RKHS as $\mathcal{H}_k^m$ where m is the dimension of the space. Define a mean function $s \in \mathcal{H}_k^m$, so that $s(x) = \sum_{i=0}^m s_i \phi_i(x)$ for $\{s_i\}_{i=0}^m$ real. By definition, the elements of $\mathcal{H}_k^m$ can be written as linear combinations of the functions ϕ_i, so we will often refer to ϕ_i as *basis* functions, without special regard for the specific function space.

A Gaussian process $f \sim \mathcal{GP}(s(\cdot), k(\cdot,\cdot))$ can be written:

$$f(x) = \sum_{i=0}^{m} \theta_i \phi_i(x) \tag{3.5}$$

where $\theta_i \sim \mathcal{N}(s_i, \lambda_i)$. This is a result of Theorem 4. Given the basis functions, generation of Gaussian process samples can be achieved by generating the coefficients $\{\theta_i\}_{i=0}^m$.

The method we present allows for application of this orthonormal basis decomposition for arbitrary input distribution settings. We will need to acquire sequences of orthonormal functions with respect to essentially arbitrary input measures with finite moments. This is made possible via the application of Favard's theorem (Theorem 6, see Section 2.4).

The theorem provides us with a simple method for construction orthonormal basis functions. Provided we can calculate the recur-

rence coefficients for a given distribution, it is possible to generate the basis functions. Construction of recursive formulae for these coefficients, and analysis of the conditioning of the mapping from moments of the measure ν to coefficients has been carried out by Gautschi (1982, and 2004). Asymptotics of orthogonal polynomials are considered by Deift et al. (1999) and more recently by Ding and Trogdon (2021) in connection to random matrix theory.

The following theorem clarifies the usefulness of the orthogonal polynomial approach to Gaussian process modelling.

Theorem 8. *Assume $(\mathcal{X}, \mathcal{F}, \nu)$ a measure space with a measure ν, and that the moments $\{\mu_i\}_{i=0}$ of ν fulfil Carleman's condition; $\sum_{n=0}^{+\infty} \mu_{2n}^{-\frac{1}{2n}} = +\infty$. Then, there exists an orthonormal sequence of functions $\phi_i : \mathcal{X} \to \mathbb{R}$ of the form $\phi_i(x) = d_i P_i(x) w^{\frac{1}{2}}(x)$ for $i \in \mathbf{N}$, where:*

- *$d_i \in \mathbb{R}^+$ is a normalising constant;*
- *P_i a polynomial $P_i : \mathcal{X} \to \mathbb{R}$ belonging to an OPS $\{P_i(x)\}_{i=0}^{\infty}$ which is orthogonal w.r.t to the measure with density $w(x)d\nu(x)$;*
- *w is any square-integrable weight function $w : \mathcal{X} \to \mathbb{R}$ such that $w(x) \leq 1$,*

and, if the support of ν is compact, then the functions $\{\phi_i\}$ form a basis in $\mathcal{L}_2[wd\nu]$.

Proof. Proof in Appendix B. □

The remarks regarding the weight function are important, because they allow us to control the extreme behaviour of polynomials far from the region containing their roots. The form described in Theorem 8 is found in the construction of Tronarp and Karvonen (2022), who

construct basis functions for specific kernels and measures. In this way, the method we describe generalises their as our functional forms encompass the ones found in their constructions.

Under the sampling paradigm described at the beginning of the previous section, for large N we can construct basis functions that are approximately orthonormal with respect to the input distribution ν as follows.

For general measures ν, define the sequence of power moments $\{\mu_i\}_{i=0}^{\infty} = \int_{\mathcal{X}} x^i d\nu$, assuming that ν is such that all these moments exist.

Define the Hankel determinants:

$$H_n = \begin{vmatrix} \mu_0 & \mu_1 & \dots & \mu_n \\ \mu_1 & \mu_2 & \dots & \mu_{n+1} \\ & & \dots & \\ \mu_n & \mu_{n+1} & \dots & \mu_{2n} \end{vmatrix} \tag{3.6}$$

$$H'_n = \begin{vmatrix} \mu_0 & \mu_1 & \dots & \mu_{n-2} & \mu_n \\ \mu_1 & \mu_2 & \dots & \mu_{n-1} & \mu_{n+1} \\ & & \dots & & \\ \mu_{n-1} & \mu_n & \dots & \mu_{2n-3} & \mu_{2n-1} \end{vmatrix}. \tag{3.7}$$

Using these, write the coefficient recurrence formulae (Gautschi, 1982):

$$\beta_n = \frac{H'_n}{H_n} - \frac{H'_{n-1}}{H_{n-1}} \tag{3.8}$$

$$\gamma_n = \frac{H_n H_{n-2}}{H_{n-1}^2}. \tag{3.9}$$

We will construct an approximately orthonormal sequence using

the empirical counterparts of these Hankel determinants,

$$\hat{H}_n = \begin{vmatrix} \hat{\mu}_0 & \hat{\mu}_1, & ..., & \hat{\mu}_n \\ \hat{\mu}_1 & \hat{\mu}_2, & ..., & \hat{\mu}_{n+1} \\ & & ..., & \\ \hat{\mu}_n & \hat{\mu}_{n+1} & ..., & \hat{\mu}_{2n} \end{vmatrix} \tag{3.10}$$

$$\hat{H}'_n = \begin{vmatrix} \hat{\mu}_0 & \hat{\mu}_1, & ..., & \hat{\mu}_{n-2} & \hat{\mu}_n \\ \hat{\mu}_1 & \hat{\mu}_2, & ..., & \hat{\mu}_{n-1} & \hat{\mu}_{n+1} \\ & & ..., & & \\ \hat{\mu}_{n-1} & \hat{\mu}_n & ..., & \hat{\mu}_{2n-3} & \hat{\mu}_{2n-1} \end{vmatrix}. \tag{3.11}$$

We then use these to construct empirical versions of the recurrence coefficients, written

$$\hat{\beta}_n^N = \frac{\hat{H}'_n}{\hat{H}_n} - \frac{\hat{H}'_{n-1}}{\hat{H}_{n-1}} \tag{3.12}$$

$$\hat{\gamma}_n^N = \frac{\hat{H}_n \hat{H}_{n-2}}{\hat{H}_{n-1}^2}. \tag{3.13}$$

We present a simple convergence theorem that shows that, using the empirical coefficient definitions above, we can construct an asymptotically orthonormal sequence that, given the result of Theorem 7 will lead to convergent eigenvalue estimators.

Theorem 9. *Given an absolutely continuous measure ν on a space $\mathcal{X} \subset \mathbb{R}$, denote its monomial moments by $\{\mu_i\}_{i=0}^{\infty}$. Define the Hankel determinants as in* (3.6), (3.7). *From these define the recurrence coefficients* (3.8), (3.9). *By Theorem (6), construct the OPS $\{P_i\}_{i=0}^{\infty}$ using $\{\beta_i\}_{i=1}^{\infty}, \{\gamma_i\}_{i=1}^{\infty}$.*

Denoting the sample moments $\hat{\mu}_j$, construct the Hankel matrices constructed from these as $\hat{H}_n, \hat{H'}_n$ and the resulting empirical recurrence

coefficient sequences $\hat{\beta}, \hat{\gamma}$. Via Favard's theorem, use these to construct $\left\{\hat{P}_i(x)\right\}_{i=0}$, the corresponding empirical orthogonal polynomial sequence. Then, as $N \to \infty$, $\hat{P}_i(x) \to_p P_i(x)$, for all i.

Proof. Proof in Appendix B. □

An initial approach to construction of a Gaussian process model consists first of calculating the empirical moments $\mu_j^w = \frac{1}{N}\sum_{i=0}^{N} x_i^j w(x_i)$ and apply Theorem 9. However, in practice we use the *modified* moment Chebyshev method, described by Gautschi (2004). The standard approach to construction orthogonal sequences on vector spaces, given an appropriate inner product, is the Gram-Schmidt process. However, this process is numerically unstable. As explained by Gautschi (2004), the modified moment Chebyshev method achieves the same result whilst using a more numerically stable process, which avoids the calculation of the especially ill-conditioned determinant formulae (3.6), (3.7).

3.4.1 Parameter Learning and Order Selection

3.4.1.1 Parameter learning

Construction of a kernel, given an orthonormal basis $\{\phi_i\}_{i=0}^{m}$, requires selection of a sequence of eigenvalues. As explained elsewhere (Reade, 1984; Reade, 1992; Kanagawa et al., 2018), the behaviour of the eigenvalues of the covariance operator has direct effects on the behaviour of samples. Specifically, the rate of decay of the eigenvalues specifies the smoothness of the Gaussian process sample functions, in terms of their differentiability class.

Kernel choice therefore directly affects the space of functions that are representable by a given Gaussian process, and this effect

depends on the behaviour of the operator eigenvalues. It is in a sense equivalent to choose either a given kernel or a specific eigenvalue sequence, as both represent the (strict) prior belief over the class of functions that the Gaussian process is capable of representing. For example, the eigenvalues of the squared exponential kernel decay exponentially, and the eigenvalues of the Matérn kernel decay at a polynomial rate (Kanagawa et al., 2018).

We parameterise Favard kernels by choosing a decreasing function $\zeta_\beta : \mathbb{N} \to \mathbb{R}^+$, indexed by a vector of hyperparameters $\beta \in \mathcal{B} \subset \mathbb{R}^d$, where d is some number of parameters. The eigenvalues $\{\lambda_i\}$ are constructed by evaluating the function $\lambda_i = \zeta_\beta(i)$ which are then used as the eigenvalues for the Favard kernel. The parameters β are chosen by maximisation of the marginal likelihood (3.2) via gradient-based optimisation.

Following similar notation to that from (Rasmussen and C. K. I. Williams, 2018), the derivative of the kernel matrix with respect to a given parameter is then given by

$$\frac{\partial K}{\partial \beta} = \sum_{i=0}^{m} \frac{\partial \lambda_i}{\partial \beta} \phi_i(x)\phi_i(x').$$

and we use the term $\frac{\partial K}{\partial \beta}$ to calculate the gradient of the likelihood with respect to β in the standard likelihood optimisation method.

3.4.1.2 Order Selection

The order, or dimension m, of the reproducing kernel Hilbert space is a key parameter in the construction of the kernel. One approach to selection of m is that of Trecate, C. K. Williams, and Opper (1999). That method selects the order of the kernel by choosing m such that

"signal" for the $(m+1)$-th basis function is outweighed by observation noise. In the notation above, this means $m^* = \min_m m$ s.t. $\lambda_{m+1} < \sigma$. In our setting, however, the noise parameter σ must be estimated, and its estimate depends on m. We attempted to use the approach by (Trecate, C. K. Williams, and Opper, 1999), but it did not converge to a unique choice of m^*. Specifically, selecting the m^* according to the prescribed rule, and then re-estimating the noise variance parameter, yields a noise variance parameter that is smaller than the "next" eigenvalue term. Instead, we present an iterative approach to order selection, based on the following theorem.

Theorem 10. *Suppose a measure space $(\mathcal{X}, \mathcal{F}, \nu)$, random variable $X \sim \nu$ and a sample from a Gaussian process $f \sim \mathcal{GP}(0, k(\cdot,\cdot))$. Define $\mathcal{H}$ as the reproducing kernel Hilbert space for $k(\cdot,\cdot)$. Assume that f belongs to $\mathcal{H}_k^{m^*}$, a projection of $\mathcal{H}$ onto a finite set of basis functions $\{\phi_i\}_{i=0}^{m^*}$ orthonormal w.r.t. ν. Assume there are data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=0}^{N}$ s.t. that $\mathbf{y}_i = f(\mathbf{x}_i) + \varepsilon_i$ for some noise $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. Denote by $\hat{m}$ the chosen order of a Gaussian process model trained on the data; by $\mathbb{E}_\varepsilon[\cdot]$ the expectation taken w.r.t. the distribution of ε; and by $\hat{\sigma}^2$ a consistent estimator of the noise parameter σ^2. Then, for any $\delta > 0$, there exists $N^* > 0$ s.t. $\mathbb{E}_\varepsilon\left[\hat{\sigma}^2\right] \leq \sigma^2 + \delta + \eta_m$ where $\eta_m \geq 0$ and $\eta_m = 0$ when $\hat{m} = m^*$, for all $N > N^*$.*

Proof. Proof in Appendix B. □

This theorem indicates that an iterative process to selecting m can be carried out as follows. First, estimate $\hat{\sigma}^2$ conditional on the order $\hat{m}$ beginning at e.g. $\hat{m} = 2$. Denoting this conditional estimator as $\hat{\sigma}^2(\hat{m})$, when $\hat{m} \leq m*$, we can expect that $\hat{\sigma}^2(\hat{m})$ approaches σ from above, and then begins increasing again when $\hat{m} > m*$. Thus one can

choose $\hat{m} = \min_m m$ s.t. $\hat{\sigma}^2(m+1) > \hat{\sigma}^2(m)$.

3.4.2 Order selection examples

In this section we present the application of this order selection method to two synthetic datasets. We generate true functions $f = \sum_{i=0}^{m^*} \theta_i \phi_i$, written as the sum of $m^* \in \{8, 12\}$ basis functions. We generate inputs $x \sim \mathcal{N}(0,5)$ and noise as $\varepsilon \sim \mathcal{N}(0, 0.5)$. Then, outputs are constructed as

$$\mathbf{y} = f(\mathbf{x}) + \varepsilon$$

for a sample size of $N = 1000$. Using the same basis functions,we construct a Favard kernel with a sequence of eigenvalues $\{\lambda_i\}_{i=0}^{\hat{m}}$ for different orders $\hat{m}$. In case 1, the eigenvalues are the same as those as derived by Fasshauer (2012b) for the smooth exponential case, and in case 2, we construct eigenvalues that are written: $\lambda_i = \left(\frac{\alpha}{i+\xi}\right)^{\kappa}$ where α, ξ are hyperparameters, and κ is a decay parameter that regulates the rate of decay of the eigenvalues. We refer to these as "polynomial" eigenvalues because they exhibit polynomial decay of degree κ.

The likelihood is then optimised to choose parameters of the eigenvalues in each case. Starting with $\hat{m} = 3$, we optimise the likelihood until convergence of σ. Then, we increase $\hat{m}$ by 1 and repeat the process. For the purposes of exposition, we continue this until $\hat{m} = 25$.

Figures 3.6, 3.7 show the results of the application of this approach to the "polynomial" eigenvalues, and Figures 3.4 and 3.5 show the results of this process in the "exponential" case. Whilst the change in the noise parameter estimate is small at the minimum (due to scaling, it is not obvious on the graph), it is positive after the

minimum. Therefore, the diagrams below illustrate the validity of method outlined in section 3.4.1. Furthermore, as one would expect from Theorem 10, this growth is larger for polynomial eigenvalues, as the bias term η_m is the sum of larger "extra" eigenvalues, because the polynomial eigenvalues decay more slowly than those in the exponential ("Fasshauer") case. In all cases presented below, the true noise parameter is $\sigma^2 = 0.5$, and the dashed vertical line corresponds to the true value, m^*.

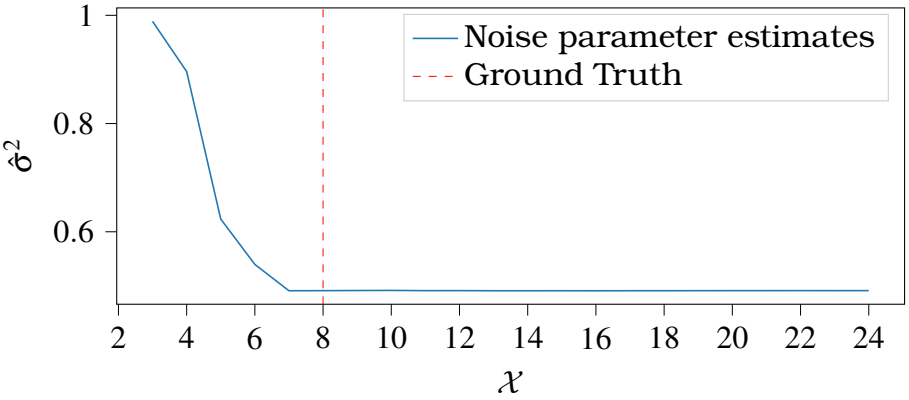

Figure 3.4: Noise parameter estimates for exponential eigenvalues. True order: $m^* = 8$.

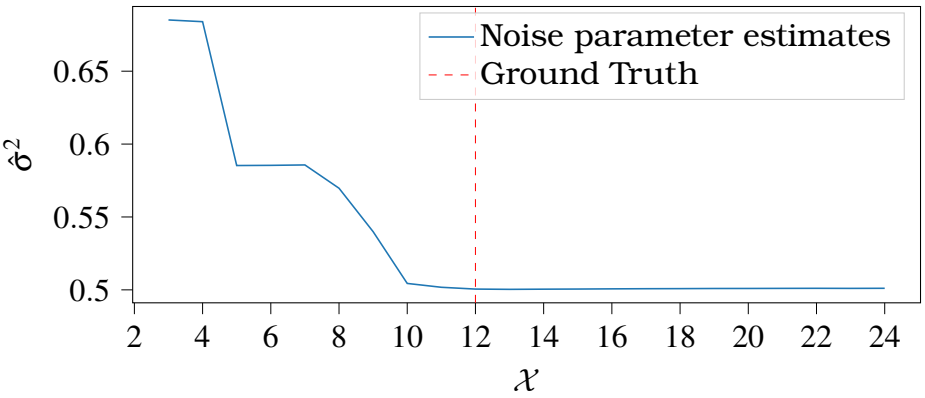

Figure 3.5: Noise parameter estimates for exponential eigenvalues. True order: $m^* = 12$.

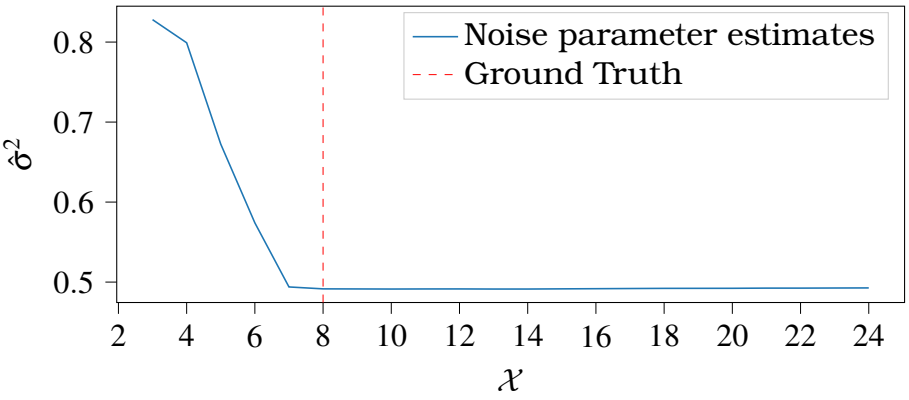

Figure 3.6: Noise parameter estimates for polynomial eigenvalues. True order: $m^* = 8$.

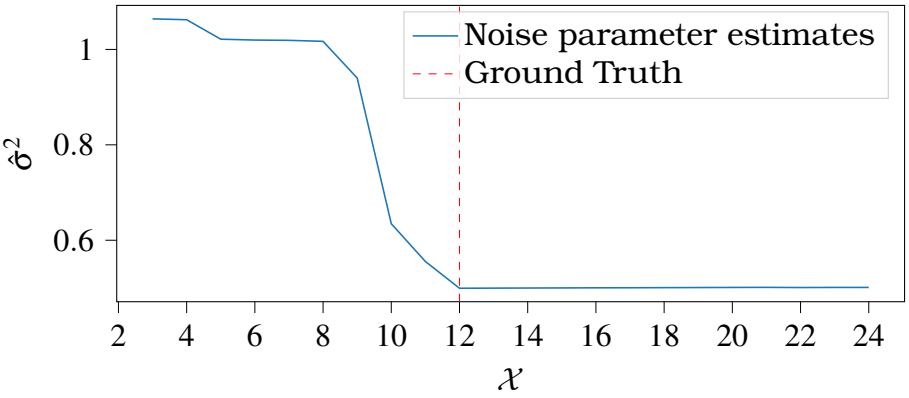

Figure 3.7: Noise parameter estimates for polynomial eigenvalues. True order: $m^* = 12$.

3.5 Posterior Sampling

The standard method (Rasmussen and C. K. I. Williams, 2018) for generating Gaussian process samples involves the construction of the Cholesky decomposition of the covariance matrix. One chooses an ordered sequence of test points $\mathbf{x} = \{x_i\}_{i=0}^{T}$ and generates the Gram matrix $K = k(\mathbf{x}, \mathbf{x}')$ by evaluating the kernel at the test points, and calculate its Cholesky decomposition $K^{\frac{1}{2}}$. Next, one generates a random standard Normal vector $\mathbf{z}$, and construct $\mathbf{f} = K^{\frac{1}{2}}\mathbf{z}$. This yields in $\mathbf{f}$ an approximate Gaussian process sample, and this is the method by which the examples in Figure 2.1 were generated.

Sampling from the posterior is achieved in a similar fashion. There the covariance matrix used is the posterior covariance matrix (2.7), and the same Cholesky decomposition approach is used. However, calculation of the Cholesky decomposition is computationally expensive, with $\mathcal{O}\left(T^3\right)$ time-complexity. This can be prohibitive when generating large numbers of "fine" samples (i.e. high T).

The Karhunen-Loève expansion (Karhunen, 1947) of a Gaussian process sample in the truncated reproducing kernel Hilbert space provides a simple method to sample from a Gaussian process. There are several useful aspects of this viewpoint of stochastic processes is that sample generation is simple and rapid; as well as providing direct access to function evaluations at any potential input point, as well as differentiability of samples. This is useful in hierarchical models where one may want to differentiate with respect to a parameter through a generated Gaussian process sample. Furthermore, the Favard kernel basis functions follow by construction a three-term recurrence; this allows for efficient evaluation of Gaussian process samples via the Clenshaw algorithm (Clenshaw, 1955).

However, there are serious downsides to this approach. The full basis function representation of a stochastic process requires a countably infinite number of basis functions to completely represent the sample. Our limitation to a truncated RKHS means that the method proposed in this chapter is technically *exact*, but the behaviour of the Gaussian process model is affected as a result. As we can see in Figure 3.8 the samples generated exhibit a decay far from the data. This is the result of using degenerate kernels (Rasmussen and Candela, 2005). Technically the model represents both a prior belief of a constant, zero-valued mean function and a prior belief of

a constant, zero-valued variance. This is interpreted as the model exhibiting very high confidence about its zero-mean prior. In practice, it is often desirable to avoid this collapse when this does not represent the true belief of the practitioner. In the literature, this phenomenon has been referred to as *variance starvation* (Mutny and Krause, 2018; Calandriello et al., 2019; Wilson et al., 2020).

In thie section we propose an approach to avoiding this variance starvation phenomenon. The technique we describe is an extension of a technique presented by Wilson et al. (2020). For clarity, note that a posterior Gaussian process sample, evaluated at a vector of test points $\mathbf{x}^*$ can be written:

$$f_{\mathbf{y}} = f^* + k(\mathbf{x}^*, \mathbf{x}) \left(K + \sigma^2 I_N\right)^{-1} (\mathbf{y} - f) \tag{3.14}$$

where here, $f_{\mathbf{y}}$ is the sample from the Gaussian process posterior, evaluated at a set of test points, $\mathbf{y}$ is the vector of observed function values, K the kernel Gram matrix evaluated at the sample $\mathbf{x}$, and f, f^* represent the value of of the prior sample evaluated at the data points and the test points respectively.

What (3.14) makes clear is that the posterior sample can be decomposed into a prior component and a posterior, data-informed component. This is essentially Matheron's rule applied to Gaussian process samples (Wilson et al., 2020).

The approach taken by Wilson et al. (2020) is to write these prior and posterior components in different bases, based on a random Fourier feature approach (Rahimi and Recht, 2007). This relies on Bochner's theorem to construct an approximation to the inner product represented by the kernel as long as the appropriate spectral density

is used to generate the features. However, Bochner's theorem is not useful here as it applies to *stationary* kernels. Our Mercer-type Favard kernels are non-stationary, since they cannot be written as a function only of the distance between inputs.

As a result, in this section we present a method based on Yaglom's theorem (Yaglom, 1987), (Genton, 2001). This extends Bochner's theorem to non-stationary kernels, by extending the spectral distribution to 2-dimensions.

Theorem 11 (Yaglom's Theorem, (Yaglom, 1987), (Genton, 2001))**.** *A non-stationary kernel $k(x,y)$ is positive definite if and only if it has the form:*

$$k(x,y) = \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} cos\left(2\pi(\omega_1 x - \omega_2 y)\right) F(d\omega_1, d\omega_2)$$

where F represents a positive bounded symmetric measure.

We now show how we can construct appropriate features for a complex Gaussian process prior to acquire a stochastic process that covaries appropriately. First, we define non-stationary random Fourier features:

Definition 14 (Non-stationary random Fourier features)**.** *Assume a method to sample from $F_{\Omega_1,\Omega_2}(\omega_1,\omega_2)$ is available. Generate a sample $\{\omega_{i1},\omega_{i2}\}_{i=0}^{R} \sim F_{\Omega_1,\Omega_2}(\omega_1,\omega_2)$, and a sample $\{b_i\}_{i=0}^{R} \sim$ Unif $[0,2\pi]$. We define, for $i \in \{1,2,...,R\}$, two sets of non-stationary random Fourier*

features:

$$\phi_i^{NS} = \cos(\omega_{i1}x + b_i) + \cos(\omega_{i2}x + b_i),$$
$$\phi_i^{'NS} = \begin{cases} \cos(\omega_{i1}x + b_i), & i \text{ odd}, \\ \cos(\omega_{i2}x + b_i), & i \text{ even}, \end{cases}$$

and present the following theorem.

Theorem 12. *Assume a symmetric, non-stationary kernel* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. *By Theorem 11, it has a 2-d spectral density* $F_{\Omega_1,\Omega_2}(\omega_1, \omega_2)$. *Define features* $\left\{\phi_i^{NS}\right\}_{i=0}^{R}, \left\{\phi_i^{'NS}\right\}_{i=0}^{R}$ *as in Definition 14. Define a* complex *Gaussian process* $\mathbf{h}(x)$*:*

$$\mathbf{h}(x) = \mathbf{f}(x) + j\mathbf{g}(x)$$

where $\mathbf{f}(x) = \sum_i \theta_i \phi_i^{NS}(x)$; $\mathbf{g}(x) = \sum_i \theta_i' \phi_i^{'NS}(x)$; $\theta_i, \theta_i' \sim \mathcal{N}(0,1)$, *and* $j = \sqrt{-1}$ *is the imaginary unit. Then,*

$$\mathbb{E}\left[\mathbf{h}(x)\mathbf{h}(x')\right] = k\left(x, x'\right).$$

Proof. Proof in Appendix B. □

Given observations, Algorithm 1 allows us to construct *complex* Gaussian process

$$\mathbf{h}(x^*) + \phi(x^*)\Lambda\Phi'\left(\Phi\Lambda\Phi' + \sigma^2 I_N\right)^{-1}(\mathbf{y} - \mathbf{h}(\mathbf{x}))$$

where $\phi(x^*)$ is a $1 \times m$ vector of the m basis functions evaluated at the test point x^*. Conditioning on the observations, the posterior

Gaussian distribution for function weights is:

$$\mathcal{N}\left(\mu_{\theta|\mathbf{y}}, K_{\theta|\mathbf{y}}\right)$$

where:

$$\mu_{\theta|\mathbf{y}} = (\Phi'\Phi + \sigma I)^{-1}\Phi'\mathbf{y}$$
$$K_{\theta|\mathbf{y}} = (\Phi'\Phi + \sigma I)^{-1}$$

where as above Φ is the $N \times m$ matrix of basis functions evaluated at the data points. We can use these weights to generate the posterior component of the sample, and sample the weights for the prior component as described in Theorem 12 and Algorithm 1.

Application of this method yields results as in Figure 3.8. The blue line is the ground truth; the red line represents the real component of the now complex Gaussian process model. As a result, the Gaussian process is able to maintain the variance far from the data that one might desire from a general functional prior.

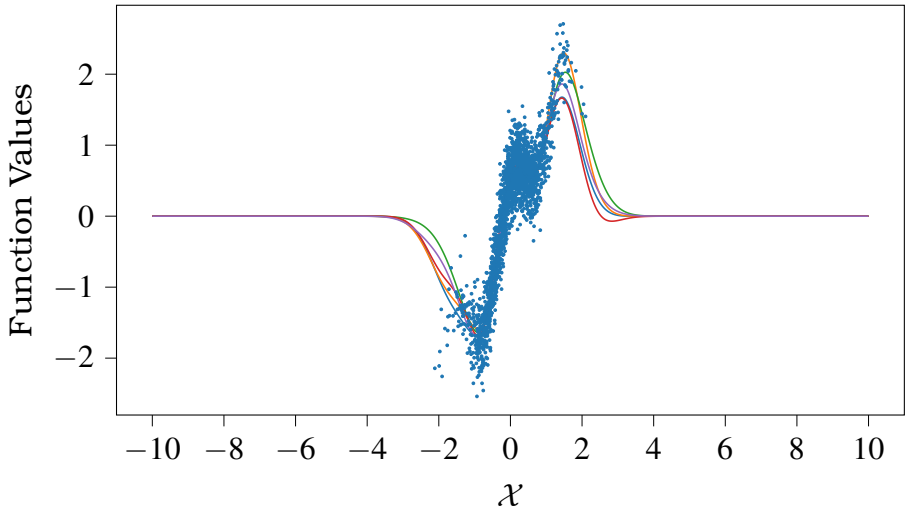

Figure 3.8: Standard posterior samples from a Favard kernel GP.

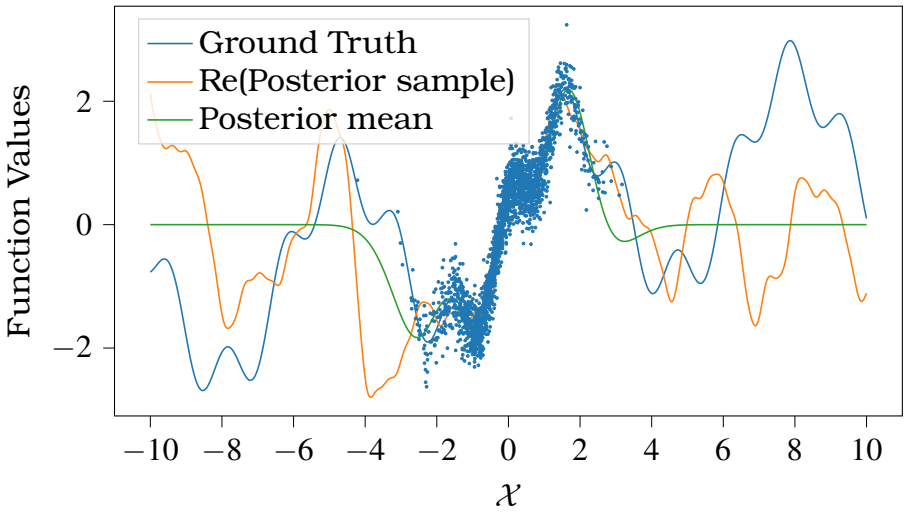

Figure 3.9: Example of posterior sample augmentation with the complex GP setup outlined in Section 3.5.

Using the features defined in Definition 14 it is possible to generate Gaussian process priors with covariance consistent with the (non-stationary) Favard kernel. Naturally, in order to use the method described above it is necessary to estimate the spectral density of the kernel; this process is described in Algorithm 1.

To summarise, we conduct a Fast Fourier transform (FFT) of the Favard kernel, which yields the spectral density evaluations at a grid of points which we refer to as *nodes*. To get the weights on the different sine basis functions, we could sample from the spectral density at these nodes. However, doing so will yield a set of weights that will induce periodicity in any generated sample functions, because the frequencies will be rational multiples of each other. In order to avoid the resulting periodicity that will occur, we use these weights to construct a Gaussian mixture model:

$$\sum_{i=1}^{m} w_i \mathcal{N}(\mu_i, v_i).$$

where the mean of each component normal distribution comprising this mixture is one of the nodes from the FFT, and the standard deviation is the same as half the distance between nodes, and the weights are the values of each node in the spectrum given by the FFT. Samples from this Gaussian mixture then approximate the spectral density of the kernel. Another approach deals with this periodicity by sampling from a mixture of uniform distributions centred at these nodes and with width equal to half distance between nodes, covering the frequency domain in a set of squares. This approach yields an approximating kernel similar to the non-stationary spectral kernel method (Remes, Heinonen, and Kaski, 2017).

3.6 Simulation studies

3.6.1 Simulated Data

In this section we present some simple examples of the Favard kernel method in comparison to the Mercer approach. In Figure 3.10, we present an example of the Favard vs Mercer basis. The eigenvalues are the same, with the same parameters applied to each. The inputs are sampled from a Gamma$(3,3)$ distribution and the outputs are generated by a test function with added Gaussian noise. The lines above and below the posterior mean represent the 5% and 95% quantiles over 5000 random samples from the posterior, for normalised input data.

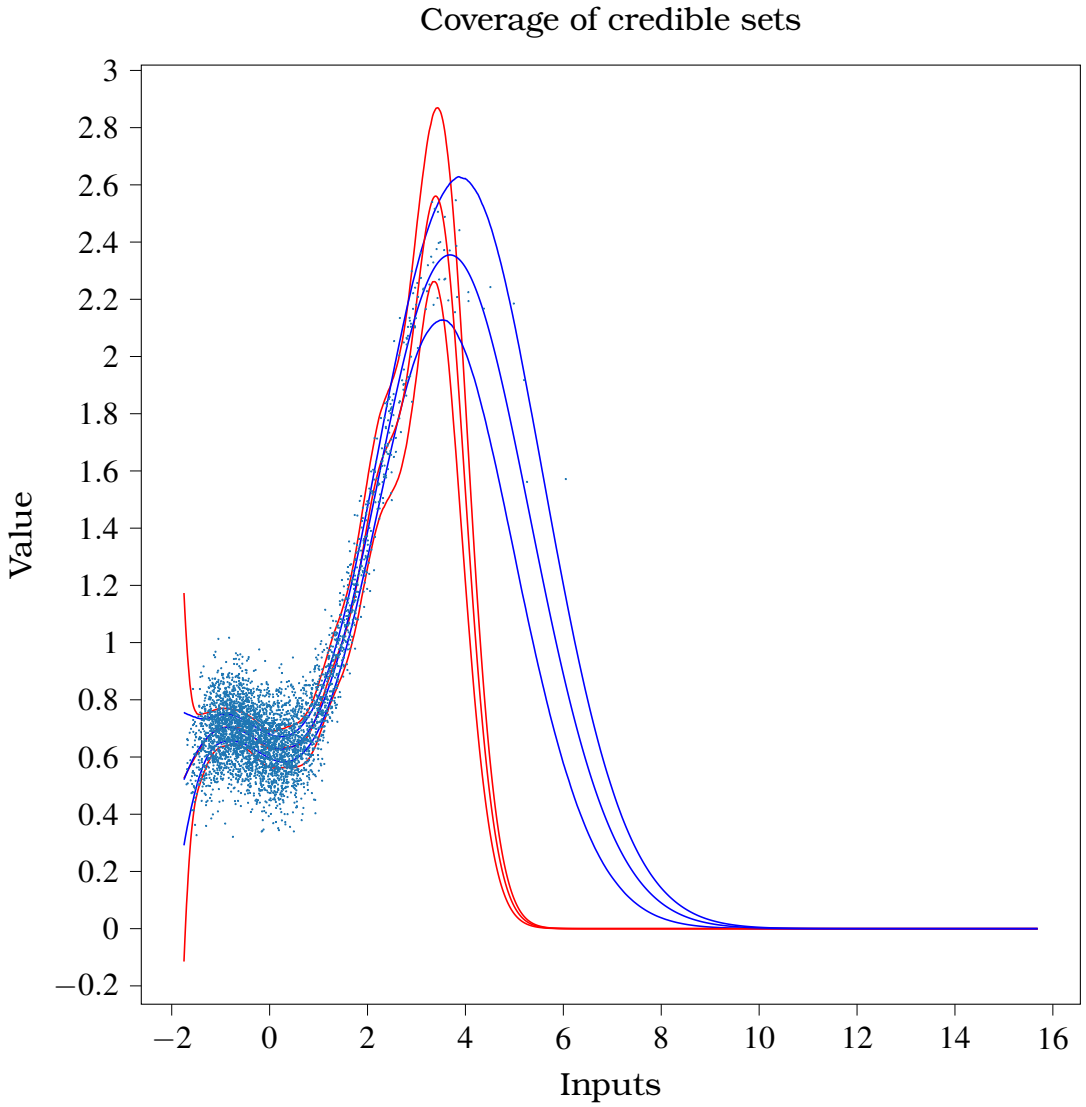

Figure 3.10: Comparison of the uncertainty in the Mercer case and the Favard case. The inputs are sampled from a Gamma(3,3) distribution; the outputs are generated by a test function with added Gaussian noise, where the function is $1.5\sin(x/2)+0.5\cos(2x)+x/8$ and the noise is distributed according to $N(0,0.1)$. The weight function on the Favard generated basis is $\exp(-x^2/4)$; the eigenvalues are the same between the two cases.Presented are 5% and 95% quantiles of random GP samples for the Mercer(in red) and Favard cases (in blue) as well as posterior means (dashed lines, Mercer in red, Favard in blue).

In Figure 3.11, we present the same example, but with the Fourier sampling for the Favard case, which clarifies its role in representing

uncertainty far from the data.

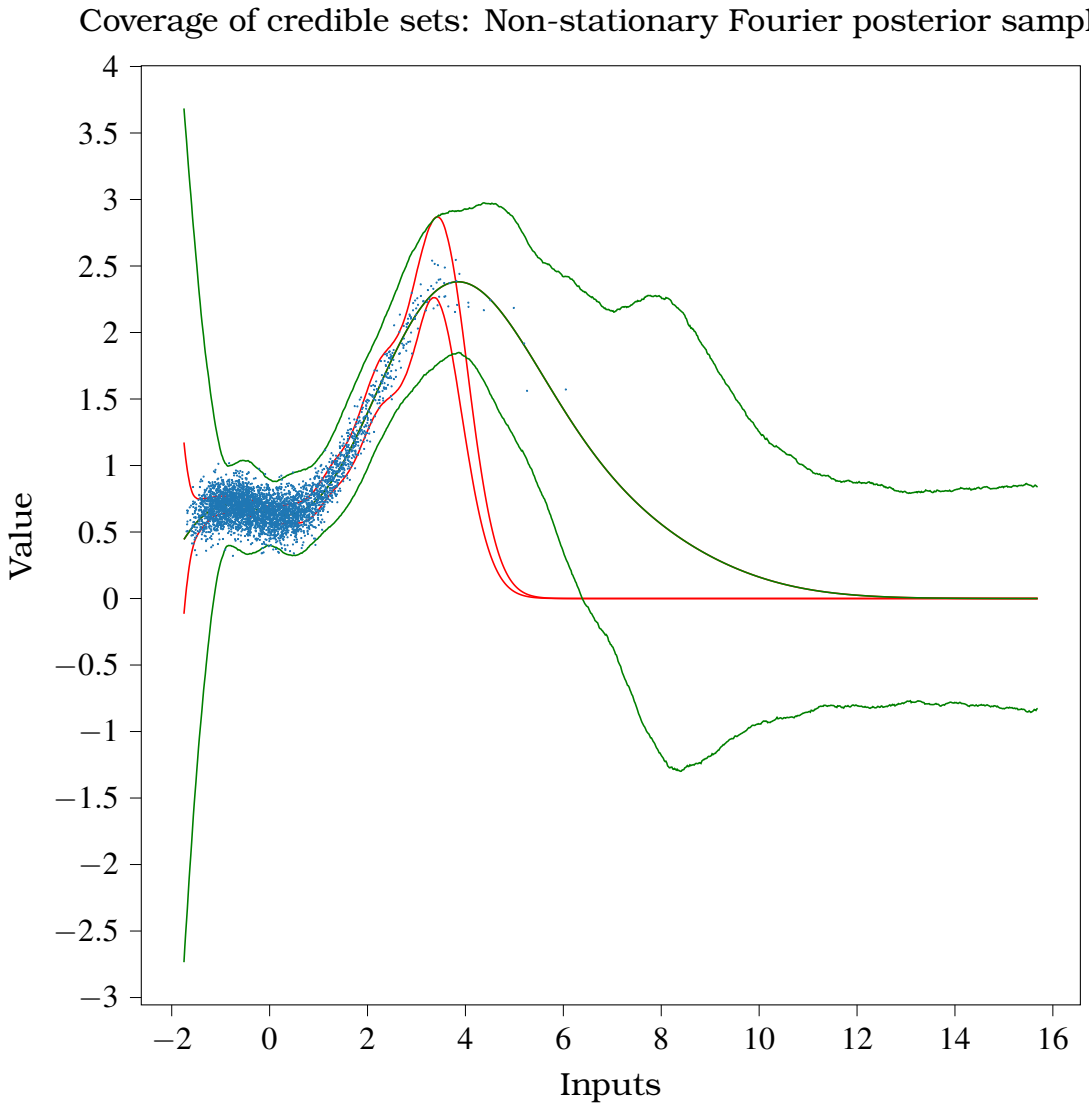

Figure 3.11: Comparison of the uncertainty in the Mercer case and the Favard case with Fourier posterior sampling as in Section 3.5 with Gaussian mixture approximation to the spectral distribution. The inputs are sampled from a Gamma(3,3) distribution; the outputs are generated by a test function with added Gaussian noise, where the function is $1.5\sin(x/2)+0.5\cos(2x)+x/8$ and the noise is distributed according to $N(0,0.1)$. The weight function on the Favard generated basis is $\exp(-x^2/4)$; the eigenvalues are the same between the two cases. Presented are 5% and 95% quantiles of random GP samples for the Mercer(in red) and Favard cases (in green), where posterior samples have been generated as in Section 3.5 as well as the posterior mean for the Favard case (black).

3.6.2 Real data comparisons

In this section we present the application of the method to subsets of two publicly available datasets; the Wine data set at the UCI Machine Learning Repository, (Frank and Asuncion, 2010) and a Formula 1 dataset (see `https://ergast.com/mrd/db`). All experiments were run on an Arch Linux system with a Ryzen 3900X, with 64Gb of RAM. No explicit graphics processing power was necessary. Both of these exhibit cases where the input distribution differs from a Normal distribution, and we utilise these examples to demonstrate the importance of appropriate choice of basis.

In both cases, we arbitrarily choose an order of $m = 10$, and train a Gaussian process using a standard "Mercer" kernel with Gaussian assumed input (as in the "Fasshauer" example in the paper), and a "Favard" kernel with constructed orthonormal basis, via the Gautschi modified moments (Gautschi, 1982) method, with weight function $w(x) = \exp(-x^2/4)$. Data is normalised for training; this leads to a trivial alteration to the recurrence coefficients, of the constructed orthogonal polynomials (Chihara, 2011), and allows one to keep m lower, since the basis functions are aligned over the data.

In each cases, the kernel hyperparameters are learned using the whole dataset, and then subsets of the dataset are used for validation. The log predictive densities in the "Favard" case are marked $\mathcal{F}$, and those in the "Mercer" case are marked $\mathcal{M}$.

3.6.2.1 UCI Wine Dataset

From the UCI Wine quality dataset, we present both a one-dimensional and two-dimensional regression example. For the one-dimensional example, we choose as input the "total sulphur", and predict "free

sulphur". For the two-dimensional example, we choose as inputs "total sulphur" and "free sulphur", and predict "residual sugar". We take a random subsample of size 600 of the input data as conditioned observations for the Gaussian process posterior. Using this Gaussian process conditioned on these observations, we take a random subset of size 50 of the remaining observations, as a test set, and calculate the log predictive density over function values at these points, evaluated at the true values. These give a measure of predictive success of the Gaussian process. We repeat this process for 1000 random subsamples, and the histogram of Figure 3.12 below displays the differences in log predictive density between the Mercer and Favard kernels at these random subsamples. Where the difference is positive, the Favard kernel predicts better than the Mercer kernel; this illustrates the better predictive ability of the use of an appropriate basis.

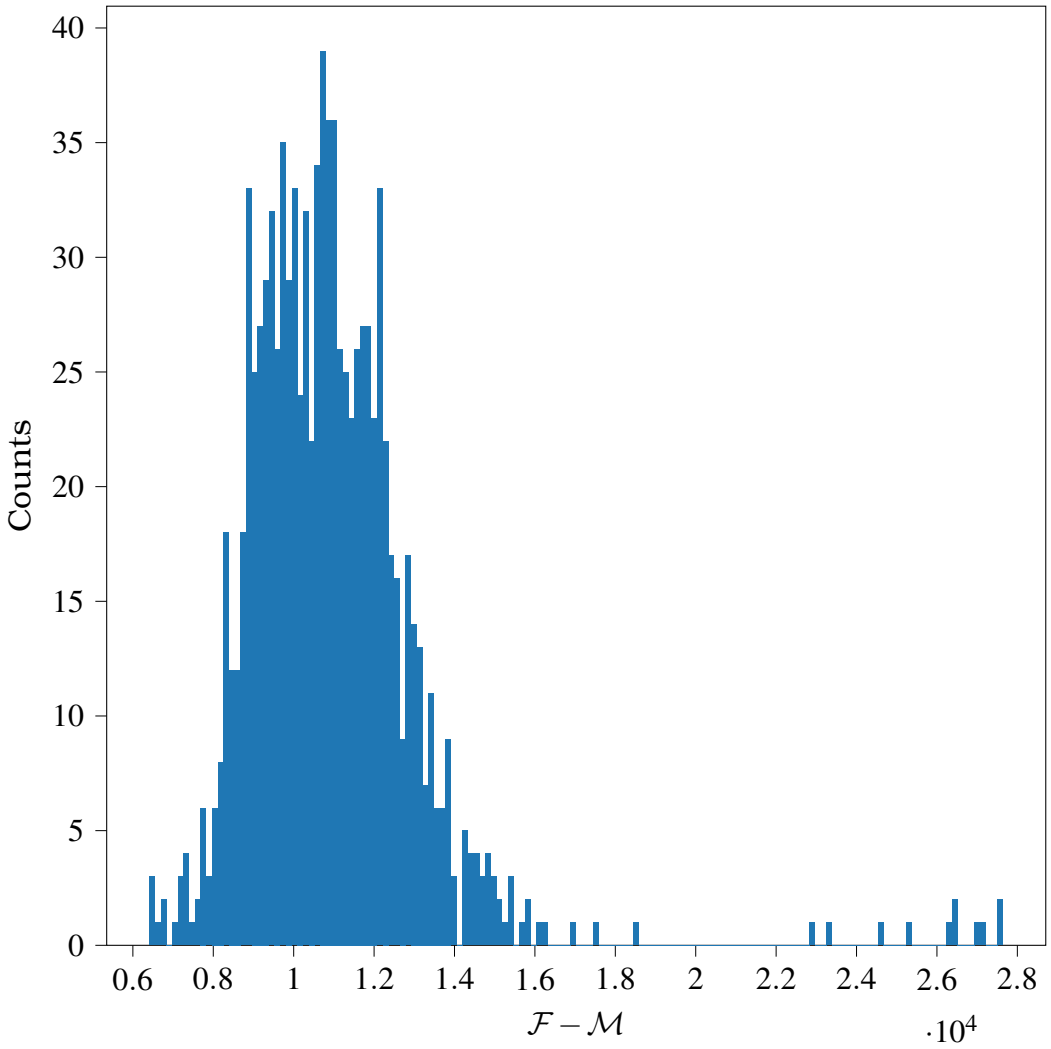

Figure 3.12: Log-predictive density differences between Favard and Mercer kernel approaches on the UCI Wine Quality Dataset (see text for the details of the regression). The histogram shows the differences in log predictive density between the Favard and Mercer kernels at random subsamples of the data. Positive values represent subsamples at which the Favard kernel predicts better than the Mercer kernel.

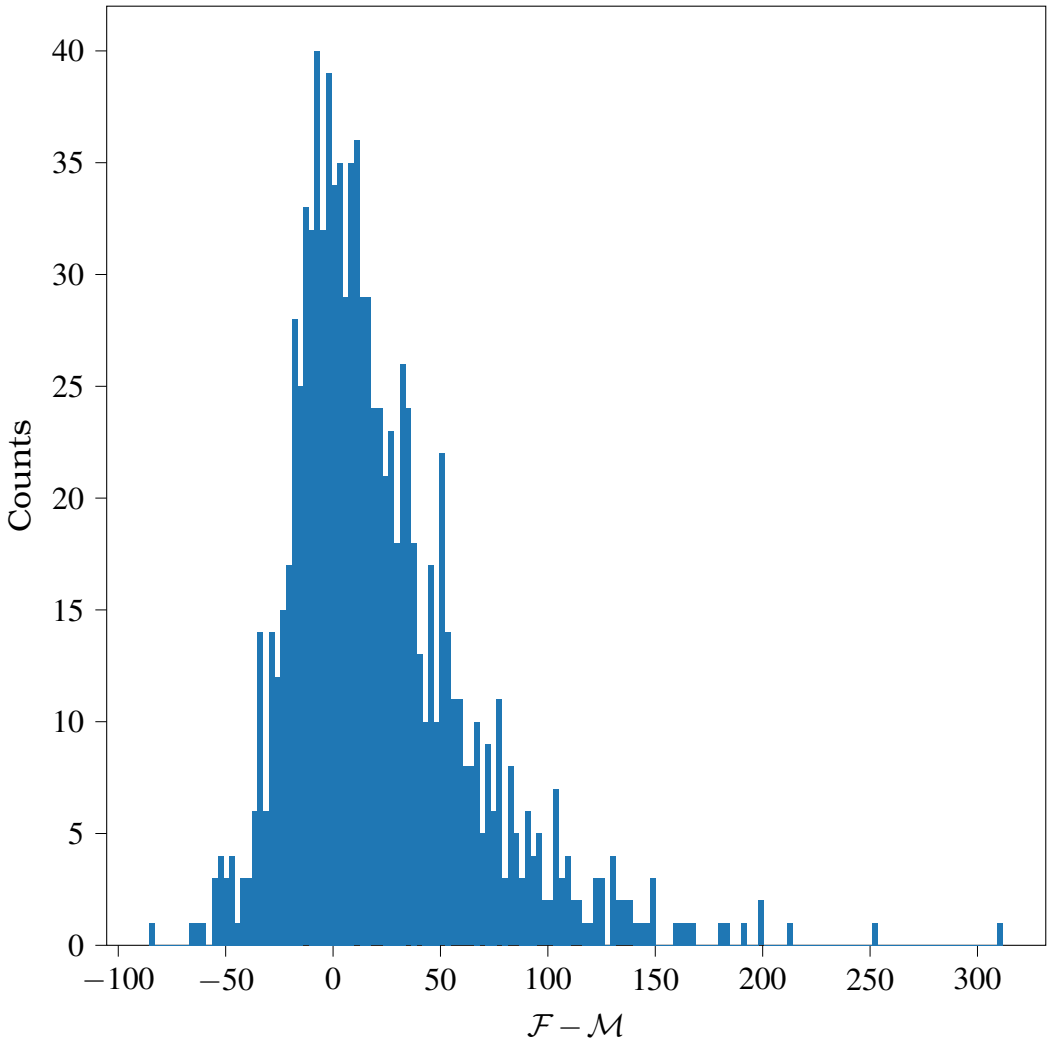

Figure 3.13: Two-dimensional Log-predictive density differences between Favard and Mercer kernel approaches on the UCI Wine Quality Dataset (see text for the details of the regression). The histogram shows the differences in log predictive density between the Favard and Mercer kernels at random subsamples of the data. Positive values represent subsamples at which the Favard kernel predicts better than the Mercer kernel.

3.6.2.2 Formula 1 Dataset

The Formula One data contains very non-Gaussian input distributions, so we considered it to be a good example for exhibition of the benefits of the Favard kernel approach. Here we take as inputs the pit stop times, and as output the count of pit stops. The pit stop

times exhibit a strong multimodal distribution. For this dataset, we take a random subsample of size 800 of the input data as conditioned observations for the Gaussian process posterior. Using this Gaussian process conditioned on these observations, we take a random subset of size 50 of the remaining observations, and calculate the log predictive density over function values at these points, evaluated at the true values. We repeat this process for 1000 random subsamples, and the histogram below displays the difference in log predictive density between the Mercer and Favard kernels.

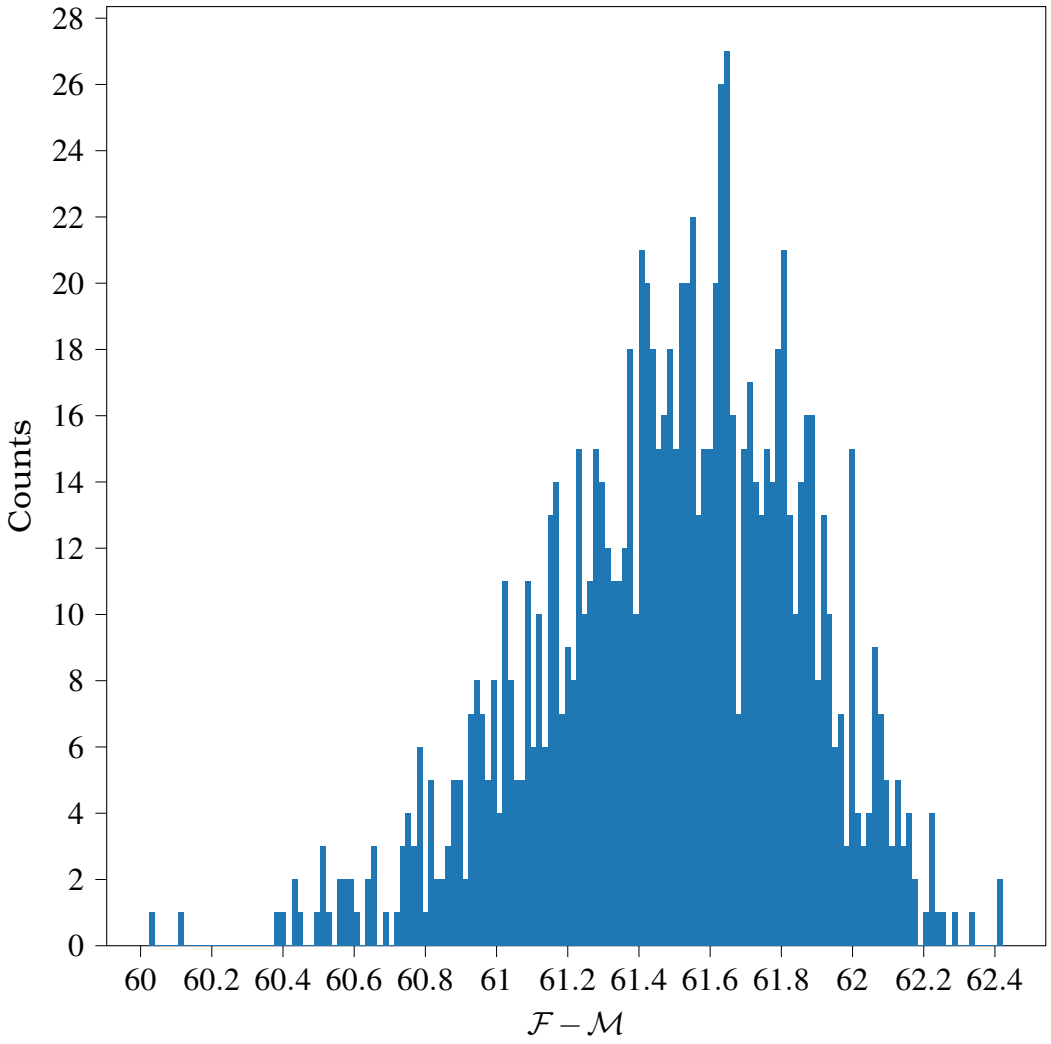

Figure 3.14: Log-predictive density differences between Favard and Mercer kernel approaches on the Ergast Formula 1 Dataset (see text for the details of the regression). The histogram shows the differences in log predictive density between the Favard and Mercer kernels at random subsamples of the data. Positive values represent subsamples at which the Favard kernel predicts better than the Mercer kernel.

3.7 Conclusion

In this chapter we have presented a method for constructing Gaussian process models from orthonormal sequences of basis functions in sampling regimes of iid inputs, and proposed a solution to the problem

of appropriate posterior sampling in the non-stationary setup. This approach unfortunately does not lend itself directly to models for Bayesian optimisation (e.g. Thompson sampling), since the input distribution is the distribution that results from the sequence of activation functions.

As presented, the method only explicitly deals with 1-d problems, and implicitly with multidimensional problems where the input measure is a product measure. In that case, it is simple to construct orthonormal sequences via a tensor product on the one-dimensional basis, since the corresponding inner product calculations decompose into products of integrals.

In future work we aim to extend this to multidimensional models with general measures via the use of multivariate orthogonal polynomials. We also aim to investigate error bounds relating to the Gaussian process resulting from the use of an approximate orthonormal basis as described herein. We believe that our proposed method constitutes a valid and applicable technique to scaling Gaussian process models for input measures with non-finite support.

Data: A Mercer kernel $k(x,y) = \sum_{i=0}^{m} \lambda_i \phi_i(x)\phi_i(y)$, feature count R, frequency count N
Data: A sequence of left-input points: $\mathbf{x} = \{x_1, x_2, ..., x_N\}$.
Data: A sequence of right-input points: $\mathbf{y} = \{y_1, y_2, ..., y_N\}$.
Result: A Gaussian process sample $\mathbf{h}$ s.t. $\mathbb{E}[\mathbf{h}(x)\mathbf{h}(y)] = k(x,y)$
$\Psi_1 = FFT(\phi_i(\mathbf{x}))$;
$\Psi_2 = FFT(\phi_i(\mathbf{y}))$;
$\Lambda = \text{diag}(\lambda_1, \lambda_2, ...\lambda_N)$;
$\overline{P} = \Psi_1 \Lambda \Psi_2'$;
Flatten $\overline{P}$ to a vector $\overline{\mathbf{p}}$;
Normalise elements of $\overline{\mathbf{p}}$ by $\sum_{i=0}^{N^2} \mathbf{p}_i$ to create probability vector $\mathbf{p}$;
for $r \leftarrow 0$ **to** *to* R **do**
  Sample integer i from probability distribution $\mathbf{p}$;
  Sample noise (e_{1r}, e_{2r}) uniformly from $[-0.5, 0.5]^2$;
  $\omega_{r1} \leftarrow (i \bmod N)$;
  $\omega_{r2} \leftarrow (i \text{ divfloor } N)$;
  $\omega_{r1} \leftarrow \omega_{r1} + e_{1r}$;
  $\omega_{r2} \leftarrow \omega_{r2} + e_{2r}$;
end
for $r = 0$ ***to*** $2R$ **do**
  $b_r \sim Unif[0, 2\pi]$;
  $\hat{\theta}_r' \sim \mathcal{N}(0,1)$;
  if $r \leq R$ **then**
    $\hat{\theta}_r \sim \mathcal{N}(0,1)$;
    $\hat{\theta}_{R+r} = \hat{\theta}_r$;
    $\hat{\phi}_r^{NSRFF}(z) = \frac{1}{\sqrt{R}} \cos(\omega_{r1} x + b)$;
  end
  else
    $\hat{\phi}_r^{NSRFF}(z) = \frac{1}{\sqrt{R}} \cos(\omega_{r2} x + b)$;
  end
  $\theta_r = \hat{\theta}_r + j\hat{\theta}_r'$;
end
return $\sum_{r=0}^{2R} \theta_r \hat{\phi}_r^{\text{NSRFF}}(z)$

Algorithm 1: Fourier feature prior GP sampling from non-stationary Mercer form kernels

Chapter 4

Feature Construction for Anomaly Detection in Dynamic Graphs

4.1 Introduction

In many modern data science problems, there are examples of datasets whose relations can best be described using a graph; Google search famously relies on a graph representation of the internet to rank webpages (Page et al., 1999), and social networks such as Facebook and Twitter can be represented as graphs of users and their connections.

In certain cases, *dynamic* graphs are an appropriate model for a given data set. By this we mean a sequence of graphs, where each graph represents a snapshot of the data at a given time. For example, in a social network, the connections between users may change over time, and so a dynamic graph is a natural model for such a dataset.

In this chapter, we present a method for constructing features that can be used for clustering and anomaly detection in dynamic sequences of weighted or unweighted, undirected graphs.

In general, the anomaly detection problem consists of identifying examples in a dataset that can be considered to be *abnormal* or *anomalous* in some sense; such outliers can be considered detrimental to training data or can represent a problem in e.g. some manufacturing process.

One approach to anomaly detection in graphs is the *feature-based* approach (Akoglu, Tong, and Koutra, 2015); in this approach, features are constructed from the graph data, and then anomalies are identified by considering the values of these features.

A downside to the "feature-based" approach is that the features constructed are often not able to capture structural properties of graphs. For example, there are node-level properties, such as the degree of a graph's nodes; edge-level properties, such as summary statistics on weights, or global properties, such as the number of connected components in a graph. On the other hand, features that capture the structure of a graph can be hard to interpret; an example is the characteristic polynomial of the graph adjacency matrix, and corresponding eigenvalues and eigenvectors. Whilst this can capture structural properties of a graph, it is not always clear how to interpret such features; the interpretation of a graph adjacency as an operator on vectors is not clear, so its spectrum, which describes the behaviour of the matrix on vectors in its invariant spaces, are an even more abstract concept for discussing the structural properties of a graph.

The key feature of the approach laid out in this chapter is that it provides an unsupervised method for feature construction that can capture the general structure of a graph. Furthermore, in certain circumstances it is capable of capturing anomalies that spectral methods are incapable of capturing. It relies on the the graph matching

polynomial, which in its coefficients captures the general structure behaviour of the graph. A connection between this polynomial and a corresponding reproducing kernel Hilbert space produces a natural embdding for graphs into the set of $\mathcal{L}^2(v)$ spaces.

4.2 Related Work

4.2.1 Anomaly Detection

Anomaly detection has a long history in the machine learning literature. Applied to graphs, various anomaly detection algorithms have arisen, with different applications requiring different approaches. For relevant work, we focus on methods pertaining to dynamic graphs, as that is the natural setting for the method we propose. However, a comperehensive review of anomaly detection for graphs can be found in Akoglu, Tong, and Koutra (2014). Density-based methods (Steinwart, Hush, and Scovel, 2005) attempt to construct a probabilistic model over graphs, and ascribe a density value to a given graph or some feature. Graphs with low density according to the model are considered anomalous. This approach requires an explicit model, but can also be done using some form of empirical density estimation. This method is often used on some latent space constructed from graph features. However, this can be hard to interpret if the latent-space mapping is learnt, as in the case of Goyal and Ferrara (2018) as opposed to hand-constructed.

Feature-based methods extract features from graphs and construct time series of feature values; general-purpose anomaly detection algorithms can be applied to these feature values in order to capture anomalous graphs. Standard features might include the dis-

tribution of node degrees or the eigenvalues of the adjacency matrix. However, as noted by Akoglu, Tong, and Koutra (2014), better methods appear to utilise more graph structure and often are not computable in polynomial time. The implication is that simpler features (such as some metric of node connectivity) may not capture enough information to correctly distinguish anomalousness graphs.

General-purpose graph anomaly detection methods may then compare either consecutive graphs, or use a "landmark" strategy that compares all constructed features to a key baseline graph feature value. Then, graphs that diverge in distance from such a baseline are considered anomalous. Many such feature approaches have been developed; but the main ones that are pertinent for comparison are parameter-free methods for weighted and unweighted dynamic graphs. To present these alternatives, we focus on the examples from the literature review by Akoglu, Tong, and Koutra (2015), which provides a comprehensive look at the relevant methods.

Although our feature construction method can be used in general regression tasks, we focus on anomaly detection in dynamic graphs. The setting is that the practitioner observes a sequence of graphs, usually on a fixed set of nodes, and wants to detect periods in which anomalous graphs occur. As noted by Akoglu, Tong, and Koutra (2015), the main approaches can be divided into four categories; feature-based, decomposition-based, community-based and window-based methods.

Two relatively computationally complex methods are the maximum common subgraph (MCS) distance and the graph edit distance (GED). The MCS distance between two graphs is defined as $d(G,H) = 1 - m(G,H)/M(G,H)$ where $m(G,H)$ is the size of the maximum

common subgraph between graphs G and H, and $M(G,H)$ is the size of the union of the graphs. It is known that calculation of the maximum common subgraph is NP-hard(Barrow and Burstall, 1976), by its relation to the maximum clique problem. These measure the size of the maximum common subgraph between two graphs.

The graph edit distance measures the number of edit operations to get from one graph to another; close graphs will have low cost of transformation between them. In general, this approach is not computationally efficient; in certain cases the compuation of the graph edit distance is equivalent to the maximum common subgraph problem (Bunke, 1997).

A quintessential spectral approach to the problem is known as the λ-distance (Shoubridge et al., 2002). This method compares graphs by the distance between the vectors of top-k eigenvalues of the adjacency matrix. Whilst it is generally simpler to calculate the distance between e.g. vectors of eigenvalues of the adjacency matrix, it is not clear that this is a good measure of graph similarity. Firstly, we do not consider that this approach yields generally interpretable comparisons. Admittedly, calculation of the eigenvalues of this matrix is likely to be computationally efficient, but there is a lack of interpretability. Naturally, the use of matrix eigenvalues places this approach in the category of spectral graph theory methods.

We consider such approaches to complement our method, as the characteristic polynomial is another graph invariant (Shi et al., 2016). However it is not always simple to interpret the spectral properties of the adjacency matrix (or the Laplacian) as an operator on vectors. Furthermore, graphs may be cospectral but between them anomalous; spectral methods will therefore not be able to capture these

differences.

Diameter distance (Gaston, Kraetzl, and Wallis, 2006) can also be used to compare graphs. This approach calculates a metric of graph diameter constructed as the average eccentricity over the graph's vertices. Then, graph diameter distance is the absolute value of the difference between the graph diameter of two graphs. As noted by the authors, this approach aims to capture structural properties of graphs. However it is not clear how well it can differentiate between similar graphs as the constructed graph diameter metric includes an averaging over vertices; this may lead to aggregation of relevant information for anomaly or change detection.

Finally, GraphScope (Sun et al., 2007) operates by constructing partitions of a sequence of graphs on-line; the decision function for whether a graph should be included in a given partition is based on the encoding cost of the new graph given the graphs in a given segment. This provides a change-point detection method. According to the authors, the results agree with intuition; this is taken to justify the approach to cluster (partition, in the language of the paper) construction.

The approach outlined in this chapter to feature construction shares properties with each of these methods. We rely on a relatively intuitive argument for the feature construction and graph comparison method, in that our approach compares graphs to graphs with idealised structure. We intepret anomalous graphs as differing from non-anomalous graphs according to the extent that they deviate from this idealised structure. Whilst this is not formally rigorous, the empirical experiments bear out the intuition for the method. Our method also has relatively high computational complexity. This is

an unfortunate side effect of the approach to capturing whole-graph structure, and methods such as the GED and MCS also suffer from this problem.

4.2.2 Matching Polynomials

Our work relies on the use of the matching polynomial, a useful graph invariant. The concept originated in chemistry where it has been used to model the placement of oxygen molecules on lattices (Heilmann and Lieb, 1972). The connection between certain graph configurations and specific orthogonal polynomials was an early observation by Heilmann and Lieb (1972). An early form of the Christoffel graph kernel we present here was introduced in that paper. Further work analysing other properties of the matching polynomial has also been carried out (Farrell, 1979; Farrell, 1980; Farrell and Wahid, 1986), including generative methods that invert the mapping from graph to matching polynomial.

Useful theoretical results have been developed by Christopher. D. Godsil (1981a), relating to the Hermite polynomials and the so-called Complement theorem; and asymptotic statistical properties of matching polynomials (Christopher. D. Godsil, 1981b). It is important to note that the characteristic polynomial and matching polynomial of a graph are identical if and only if the graph is a forest (Shi et al., 2016, see Theorem 5.3.1). As a result, there exist cospectral graphs, i.e. having the same characteristic polynomial, with different matching polynomials. Examples of such graphs are presented in Section 4.7.1.2.

The computational intractability of the matching polynomial was also an early observation (Jerrum, 1987). Specifically, computation

of the matching polynomial is #P-hard, which can informally be described as the counting problem equivalent of the NP-hard class of decision problems. Via connections to the computation of the matrix permanent, FPRAS (fully polynomial-time randomized approximation scheme) for this problem have been developed. However, the size of the constant terms in the polynomial complexity unfortunately are likely very large. Justification for this belief has been provided by Newman and Vardi (2020), who present an implementation of an FPRAS for the calculation of the permanent of a matrix, and show that the point at which the FPRAS beats the naïve approach, the time taken to calculate the permanent is infeasible (on the order of 400,000 years) in both cases. Since there is an equivalence between the matching polynomial of a bipartite graph and the calculating of the permanent of its adjacency matrix, this result is also applicable to the matching polynomial of a graph in certain cases, and without detailed analysis of this problem for matching polynomials, we expect our method to remain confined to cases of small dynamic graphs. However, we provide a relatively fast implementation in Rust (Matsakis and Klock, 2014) that utilises a CPU word-size representation of each graph node in order to best utilise hardware capabilities. We also provide a fast approximation method for calculation of the matching polynomial based on the Barvinok estimator (Barvinok, 1999).

Furthermore, the methods we outline are not only valid for matching polynomials as a graph invariant. As noted by Christopher. D. Godsil (1992), the characteristic polynomial also exhibits a Christoffel kernel property, and the resulting kernel could be used to construct a graph measure embedding as outlined below. However, the characteristic polynomial does not exhibit the same recurrence that yields

their interpretation as orthogonal polynomial sequences.

4.3 Preliminaries: Graph theory

We now present preliminary theory and definitions with the relevant graph theory concepts for the next chapter.

Definition 15 (Graph)**.** *A graph is a pair $\mathcal{G} = \langle V, E \rangle$ where V is a set of* nodes *(equiv.* vertices*) and E is a set of* edges*, that connect elements of V.*

In all cases, the graphs in this chapter will be *simple* graphs: graphs with no edges from nodes to themselves, and no sets of multiple edges between a given pair of nodes. Associated with a graph is the concept of a matching, or a dimer arrangement.

Definition 16 (Dimer)**.** *A dimer is a pair of nodes, connected by an edge.*

Definition 17 (Matching)**.** *Given a simple graph $\mathcal{G}$, a m-matching (or m-dimer arrangement) is a set of m dimers placed on the edges of $\mathcal{G}$ such that no two dimers share a node. A perfect matching on a graph $\mathcal{G}$ of $2k$ nodes is a k-matching on that graph; i.e. one that covers all the nodes.*

To clarify these concepts, we present in Figure 4.1 an example of a matching on a graph.

Definition 18 (Matching Polynomial)**.** *The matching polynomial of a simple graph $\mathcal{G}$ is the polynomial*

$$Q(\mathcal{G};x) = \sum_{m=0}^{\lfloor n/2 \rfloor} (-1)^m M_m x^{n-2m} \tag{4.1}$$

where M_m is the number of m-matchings in $\mathcal{G}$.

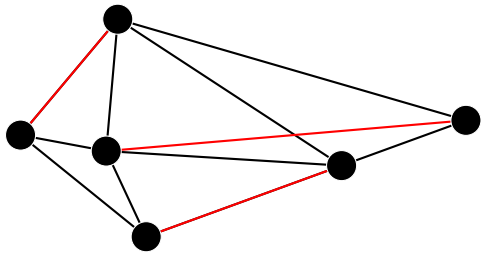

Figure 4.1: A 3-matching on a graph of 6 vertices. Red edges are in the matching; black edges are not.

The matching polynomial is a graph invariant (Cvetkovic et al., 1988); i.e. a property of a graph that does not change depending on the representation of the graph. They were originally defined by Heilmann and Lieb (1972) in the setting of monomer-dimer systems as an approach in chemistry to describe the behaviour of molecules laid on a surface. There they note some of the useful properties of certain matching polynomials, some of which we will utilise in the method described in this chapter.

4.4 Model

The method presented in this chapter relies on a connection between the matching polynomial of a graph, and sequences of orthogonal polynomials. For definitions and results relating to orthogonal polynomials, see Chapter 2. Firstly, we note the *vertex-deletion recurrence*, a property unique to the matching polynomial (Christopher David Godsil, 2017). Denoting the graph with node v removed by $\mathcal{G}-v$, the vertex removal recurrence is written:

$$Q(\mathcal{G};x) = xQ(\mathcal{G}-v;x) - \sum_{i=0}^{n} w_{i,v}Q(\mathcal{G}-v-i;x) \tag{4.2}$$

where $w_{i,v}$ is the weight of the edge between v and i. To illustrate the connection between matching polynomials and orthogonal polynomi-

als, we present a simple result which shows the relation between the structure of a graph and its matching polynomial. We will require the definition of a *complete node sequence*:

Definition 19 (Complete node sequence)**.** *A complete node sequence σ on a graph of n nodes is a sequence of nodes $v_1, v_2, \ldots, v_n$ such that v_i appears exactly once for each $i \in \{1, \ldots, n\}$. The i-th element of a complete node sequence σ is denoted by σ_i.*

Graphs for which the matching polynomial is an element of an orthogonal polynomial sequence often exhibit some specific structure. In this section we present a theorem that allows us to utilise this idea to compare the matching polynomial of a graph to a given orthogonal polynomial sequence, and this will allow for general description of graph structure and a method for graph comparison.

The classic examples of this (Heilmann and Lieb, 1972) are the matching polynomials of the cycle graph, and the complete graph. The cycle graph on n nodes, has matching polynomial recurrence:

$$Q(C_n;x) = xQ(C_{n-1};x) - 2Q(C_{n-2};x)$$

which is the recurrence relation for the following:

$$Q(C_n;x) = 2T_n(x/2)$$

where T_n is the n-th Chebyshev polynomial of the first kind, orthogonal with respect to a scaled Beta distribution.

Another classical example is that of the complete graph:

$$Q(K_n;x) = xQ(K_{n-1};x) - (n-1)Q(K_{n-2};x)$$

which is the recurrence relation for the Hermite polynomials, orthogonal with respect to the Gaussian distribution.

Favard's theorem, in fact, gives us a generalisation of these observations, which we present as the following theorem.

Theorem 13. *Given a simple graph $\mathcal{G}$, and a complete node sequence σ of $\mathcal{G}$, denote by $\mathcal{G}^\sigma$ the sequence of graphs $\mathcal{G}^\sigma = \{\mathcal{G}, \mathcal{G}_1, \mathcal{G}_2, ...\mathcal{G}_n\}$, where $\mathcal{G}_i$ is the graph obtained by removing nodes $\sigma_1, \sigma_2, \ldots \sigma_i$ from $\mathcal{G}$. If, in the matching polynomial vertex-deletion recurrence* (4.2)*, written out for $\mathcal{G}_i \in \mathcal{G}^\sigma$, the summands in the summation term are independent of the index of summation i, then the sequence of matching polynomials $\{Q(\mathcal{G}_i; x)\}_{i=0}^{n}$ is orthogonal with respect to some measure ν.*

Proof. Proof in Appendix B. □

In this chapter we will use these properties to build a method for evaluating the distance of a given graph from e.g. the fully connected graph, via the relation between a given matching polynomial and a corresponding sequence of orthogonal polynomials.

This relation is clarified in Theorem 14 below. Preliminary to this theorem however, we present some further definitions relating to orthogonal polynomial sequences.

Definition 20 (Christoffel-Darboux Kernel)**.** *Given a sequence of polynomials $\{P_i\}_{i=0}^{\infty}$, orthonormal with respect to a measure ν, the Christoffel-Darboux kernel of order m associated with $\{P_i\}_{i=0}^{\infty}$ is defined as*

$$k_m^\nu(x, y) = \sum_{i=1}^{m} P_i(x) P_i(y) \tag{4.3}$$

A useful property, dependent on the three-term recurrence associated with orthogonal polynomials, is that the Christoffel-Darboux

kernel can be rewritten in a useful form.

Lemma 1 (Christoffel-Darboux kernel formula)**.** *Given a sequence of orthonormal polynomials $\{P_i\}_{i=0}^{\infty}$ orthogonal with respect to a measure ν, the Christoffel-Darboux kernel of order m associated with $\{P_i\}_{i=0}^{\infty}$ can be written:*

$$k_m^{\nu}(x,y) = \frac{\gamma_m}{\gamma_{m+1}} \frac{P_{m+1}(x)P_m(y) - P_m(x)P_{m+1}(y)}{x-y}$$

where γ_i is the leading coefficient for P_i.

Proof can be found in (Chihara, 2011). Taking the limit as $x \to y$, yields the useful *confluent* form:

$$k_m^{\nu}(x,x) = \frac{\gamma_m}{\gamma_{m+1}} P'_{m+1}(x)P_m(x) - P'_m(x)P_{m+1}(x)$$

where $P'_i(x)$ is the derivative of $P_i(x)$.

Corresponding to this confluent form of the Christoffel-Darboux kernel is the Christoffel function:

Definition 21 (Christoffel function)**.** *Given a Christoffel-Darboux kernel $k_m^{\nu}(\cdot,\cdot)$, the associated Christoffel function $\Gamma_m^{\nu}(x)$ is the reciprocal of its diagonal:*

$$\Gamma_m^{\nu}(x) = \frac{1}{k_m^{\nu}(x,x)}.$$

This function provides a connection between a given orthogonal polynomial sequence and its measure of orthogonality (Lasserre, Pauwels, and Putinar, 2022). The method described in this chapter will exploit this connection to produce an feature of graphs that can be used to mark out anomalous examples in a given dataset, relative to a given landmark graph.

A final definition for this chapter is what we define as the graph

Christoffel kernel. This is connected to ideas even in the original work of Heilmann and Lieb (1972), and is a useful tool for the method described in this chapter.

Definition 22 (Graph Christoffel-Darboux Kernel)**.** *Given a simple graph $\mathcal{G}$, and its deck $D_{\mathcal{G}} = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_n\}$, where $\mathcal{G}_i$ is the graph $\mathcal{G}$ with vertex i removed, the graph Christoffel-Darboux kernels are defined:*

$$k^i_{\mathcal{G}}(x,y) = \frac{Q(\mathcal{G};x)Q(\mathcal{G}_i;y) - Q(\mathcal{G}_i;x)Q(\mathcal{G};y)}{x-y} \tag{4.4}$$

with the corresponding graph Christoffel functions defined $\Gamma^i_{\mathcal{G}}(x) = \frac{1}{k^i_{\mathcal{G}}(x,x)}$

We are now equipped to present the main result of this section, which is the following theorem. This clarifies the connection between a given graph matching polynomial and sequences of orthogonal polynomials. It relies essentially on the spectral theorem for l^2 sequences, and describes the ability to "rotate" a given sequence of matching polynomials into an orthogonal one.

Theorem 14. *Let $\mathcal{G}$ be a simple graph with n nodes, and σ a complete node sequence of $\mathcal{G}$. Denote by $k^{\sigma_1}_{\mathcal{G}}(x,y)$ the graph Christoffel-Darboux kernel corresponding to the graph $\mathcal{G}_{\sigma_1}$ as in Definition 22, where $\mathcal{G}_{\sigma_1}$ is the graph obtained by removing σ_1 from $\mathcal{G}$. Then, there exist a measure ν and a sequence of $n = m+1$ orthogonal polynomials $\{P_i\}_{i=0}^{m+1}$ orthonormal with respect to ν, such that $P_{m+1} = cQ(\mathcal{G};x)$ for some constant c and $k^i_{\mathcal{G}}(x,y) = k^{\nu}_m(x,y)$.*

Proof. Proof in Appendix B. □

The theorem essentially states that the matching polynomial of a graph is an element of *some* orthogonal polynomial sequence; and the

approach in this chapter is to compare graphs between themselves via comparison of their orthogonal polynomials. However, it is not necessarily clear how to compare orthogonal polynomials directly.

Work on perturbation of orthogonal polynomials is an active area of research in analysis (Deift, 2000; Ding and Trogdon, 2021). The approach developed in that vein of research is based on comparison of the Steiltjes transform of the measures of orthogonality. In this work we take a simpler approach that has not yielded specific guarantees, but allows empirically for a useful unsupervised method of comparing graphs. The key point that is noted by Ding and Trogdon (2021) and analysed previously by (Gautschi, 1986) is that the orthogonal polynomials themselves are sensitive to perturbations in the moments of the measure of orthogonality. This is described as the result of the poor conditioning of the mapping between moments and orthogonal polynomial coefficients. Small perturbations in the moments of the measure lead to large perturbations in the coefficients of the corresponding orthogonal polynomials; this naturally leads to difficulty in comparison between given orthogonal polynomals, as seemingly "distant" orthogonal polynomials, in terms of their coefficients, may represent close measures of orthogonality.

As a result it is necessary to capture this variation between measures using a more robust tool. This is provided by the Chebyshev-Markov-Steiltjes inequalities.

Lemma 2 (Chebyshev-Markov-Stieltjes Inequalities (Lasserre, Pauwels, and Putinar, 2022))**.** *Denote by k_m^ν the Christoffel-Darboux kernel of order m corresponding to a measure ν. Select a value z such that $P_{m-1}(z) \neq 0$ where P_i is the orthonormal polynomial of degree i with*

respect to ν. *Construct the* Christoffel numbers $\{\beta_1, \beta_2, \ldots, \beta_m\}$ *as the roots of the polynomial* $k_m^{\nu}(z,\cdot)$. *Then, the following inequalities hold:*

$$\sum_{i:\beta_i<x} \Gamma_{m-1}^{\nu}(\beta_i) \leq F_{\nu}(z) \leq \sum_{i:\beta_i\leq x} \Gamma_{m-1}^{\nu}(\beta_i) \tag{4.5}$$

where F_{ν} *is the distribution function associated with the measure* ν.

Note that the confluent form of the Christoffel kernel implies that the the m-th and $(m-1)$-th orthogonal polynomials are highly informative for the measure of orthogonality, because they capture all the information from the polynomials below. It is also known that the inequalities (2) are convergent (Lasserre, Pauwels, and Putinar, 2022), in the sense that the left and right hand sides converge to the same value as $m \to \infty$.

We can regulate the variation in orthogonal polynomial coefficients caused by variation of the moments of the measure of orthogonality by utilising the Chebyshev-Markov-Steiltjes inequalities; changes in the measure of orthogonality will be captured in variations in the corresponding graph Christoffel functions.

We can thus construct, given a graph $\mathcal{G}$, and a complete node sequence σ, an embedding into the space of measures by constructing the function:

$$\Gamma_{m-1}^{\mathcal{G}}(x) = \frac{1}{k_{\mathcal{G}}^{\sigma}(x,x) - Q(\mathcal{G}_{\sigma};x)^2}$$

and as a result the *measure estimate*:

$$\hat{\nu}_{\mathcal{G},\nu}(x) = \sum_{i:\beta_i\leq x} \Gamma_{m-1}^{\mathcal{G},\nu}(\beta_i) \tag{4.6}$$

As shown by Lasserre, Pauwels, and Putinar (2022), this is a consis-

tent estimator of the distribution function of the measure of orthogonality of the sequence of orthogonal polynomials. that has as n-th and $n-1$-th elements the matching polynomial of a given graph and the matching polynomial of the graph with one node removed.

4.5 Feature Construction

We can now describe the method for constructing feature embeddings for graphs, and present their application to unsupervised anomaly detection in graphs. Having constructed the Christoffel function (See Definition 22) for a given graph, and the corresponding measure estimate (4.6), we can compare a graph with another by calculating an appropriate distance between their measure estimates. Given that the measure estimate is a right-continuous, increasing step function, it is itself a valid distribution function for a measure, with support equal to $\{\beta_i\}_{i=0}^{n}$. In order to compare such measures, a distance function that does not require equality of support will be useful. A good example is the squared Maximum Mean Dispersion (Gretton et al., 2007).

Definition 23. *Denoting a pair of measures ν_1, ν_2, the maximum mean dispersion (MMD) between ν_1, ν_2, given a function set $\mathcal{F}$ is defined:*

$$MMD(\nu_1, \nu_2, \mathcal{F}) = \sup_{f \in \mathcal{F}} \left(\mathbb{E}_{\nu_1}[f] - \mathbb{E}_{\nu_2}[f] \right)$$

where $\mathcal{F}$ is a set of functions $\{f | ||f||_{\mathcal{H}} \leq 1\}$ and $\mathcal{H}$ is a Hilbert space of functions induced by a reproducing kernel k. The squared *MMD is then given by:*

$$MMD(\nu_1, \nu_2, \mathcal{F}) = \mathbb{E}_{\nu_1}[k(x,x)]^2 + \mathbb{E}_{\nu_2}[k(y,y)]^2 - 2\mathbb{E}_{\nu_1,\nu_2}[k(x,y)]$$

This allows us to construct a simple distance function on this embedding of graphs. Another valid metric, often used in the literature (see e.g. Gao and Kleywegt, 2016; Kolchinsky and Tracey, 2017) would be the Wasserstein distance, given that it can handle distributions that do not have common support.

Denoting the normalised measure estimate constructed for a graph $\mathcal{G}$ with n nodes, indexed by node i, by $\hat{\nu}^i_{\mathcal{G}}$, we denote a vector whose elements are the points in its discrete support by $\beta \in \mathbb{R}^n$, and its corresponding vector of probabilities $\mathbf{p}$. Similarly, denote by $\tilde{\beta}$ the support of a chosen base measure (such as the Christoffel function measure estimate for the Normal distribution) and by $\tilde{\mathbf{p}}$ the corresponding vector of probabilities. Then the squared Maximum Mean Dispersion between two measure estimates is given by:

$$\begin{aligned} \mathrm{MMD}^2(\hat{\nu}, \hat{\nu}') = &\sum \mathbf{p}\mathbf{p}' \odot k(\beta, \beta) \\ &+ \sum \tilde{\mathbf{p}}\tilde{\mathbf{p}}' \odot k\left(\tilde{\beta}, \tilde{\beta}\right) \\ &- 2\sum \mathbf{p}\tilde{\mathbf{p}}' \odot k\left(\beta, \tilde{\beta}\right). \end{aligned}$$

where $\odot$ denotes the Hadamard product; $\mathbf{p}'$ the transpose of the probability vector $\mathbf{p}$, and $\sum$ denotes summation over all elements of the matrix.

This value constitutes a useful, structure-capturing feature for general dynamic graph tasks, such as anomaly detection, clustering, and other general regression problems. We present how it can be used for anomaly detection, clarifying the general idea in the form of Algorithm 2.

Having constructed the representation described above, it is possible to compare graphs by comparing their (MMD) distances to a

Data: sequence of graphs $\{G_i\}$
Data: base comparison graph G_0
Function `get_measure_estimate(`G, v`)`**:**
  $\mu \leftarrow$ `get_matching_polynomial(`G`)` ;
  $\mu' \leftarrow$ `get_matching_polynomial(`G - v`)` ;
  $\mathbf{r} \leftarrow$ `calculate_roots(`μ`)` ;
  $\mathbf{k} \leftarrow$ `christoffel_kernel(`μ, μ'`)` ;
  $\mathbf{c} \leftarrow \frac{1}{\mathbf{k}}$;
  $\hat{\nu} \leftarrow$ `cum_sum(`$\mathbf{c}$ $(\mathbf{r})$`)` $/ \sum_i \mathbf{c}(\mathbf{r}_i)$;
  return $\hat{\nu}$;
Procedure `detect_anomalies(`$\{G_i\}$, G_0, ε`)`**:**
  a $\leftarrow [0] \times$ N ;
  $\hat{\nu}_0 \leftarrow$ `get_measure_estimate(`G_0`)` ;
  for $i \leftarrow 0$ **to** N **do**
    MMD $\leftarrow$ `MMD(`$\hat{\nu}_0$, $\hat{\nu}$`)` ;
    if MMD $\geq \varepsilon$ **then**
      a$[i] \leftarrow$ True ;
    end
  end
  return a;

Algorithm 2: Anomaly detection algorithm. Returns an array of indices of the graphs deemed anomalous.

given base graph with specific behaviour. Multivariate features can be constructed by comparing simultaneously to multiple base measure estimates.

4.6 Computation

4.6.1 Theory

We now discuss the computation of the Christoffel measure estimates as in (4.6). The computational complexity of their construction is dominated by the computation of the matching polynomials of the given graphs. The standard approach (Stein and Joyner, 2005) to computation of the matching polynomial follows the *edge-removal*

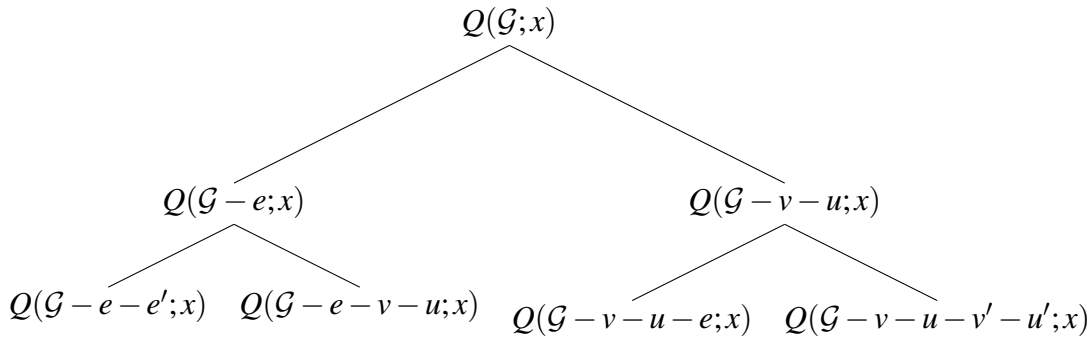

Figure 4.2: Example of a subset of the computational tree for the matching polynomial. At each node is a matching polynomial, and each polynomial is the sum of its children. The root of the tree is the matching polynomial of the graph $\mathcal{G}$; the leaves will be monomials. Computation of the matching polynomial can thus be achieved recursively, by travelling down the tree and incrementing the coefficient of the k-th order term in the matching polynomial by one for each leaf node whose corresponding graph has k nodes.

recurrence (Christopher David Godsil, 2017):

$$Q(\mathcal{G};x) = Q(\mathcal{G}-e;x) - Q(\mathcal{G}-v_1-v_2;x) \tag{4.7}$$

Specifically, this can be used to compute the matching polynomial as follows. Firstly, we alter (4.7) to read

$$Q(\mathcal{G};x) = Q(\mathcal{G}-e;x) + Q(\mathcal{G}-v_1-v_2;x).$$

This is equivalent to (4.7), but the signs of the coefficients have been flipped. Since the *signed* matching polynomial can be acquired from the signless matching polynomial by a simple flipping of some of the coefficient signs, we can recursively compute the matching polynomial by noting that the signless recurrence (4.6.1) implies a binary computational tree. We present a clarifying diagram in Figure 4.2. Each node represents a matching polynomial of some graph

$\mathcal{G}'$, and has children consisting of the matching polynomials of the graphs $\mathcal{G}' - e$ and $\mathcal{G}' - v - u$ where v, u are the nodes connected by the edge e. The leaves of this tree are graphs that have no edges, i.e. are disjoint unions of isolated nodes. The matching polynomial of a disjoint union of n unconnected nodes is simply the monomial x^n. This means that one can calculate the matching polynomial of a graph by recursively traversing this tree; when a leaf is reached, simply increment the corresponding monomial term in the polynomial by 1. Once this process is complete, we flip the corresponding terms (every fourth order, including terms that have zero coefficient) in the resulting polynomial to obtain the signed matching polynomial.

The resulting algorithm exhibits complexity of $\mathcal{O}(2^n)$, where again n is the number of nodes. However, we can take advantage of the Godsil complement theorem (Christopher David Godsil, 2017) to speed up calculation in dense graphs; this yields complexity on the order of $\mathcal{O}\left(2^{n/2}\right)$ as the depth of the relevant computational tree is at maximum $n/2$.

Theorem 15 (Godsil Complement Theorem, (Christopher David Godsil, 2017))**.** *Given a graph $\mathcal{G}$ with matching polynomial $Q(\mathcal{G};x)$, and the complement of $\mathcal{G}$ written as $\overline{\mathcal{G}}$, write the matching polynomial as*

$$Q(\mathcal{G};x) = \theta' \mathbf{x}$$

where θ denotes the vector coefficients of the polynomial in the monomial basis $\mathbf{x} = (1, x, x^2, \ldots)$.

Correspondingly write the matching polynomial for the graph comple-

ment $\overline{\mathcal{G}}$ *as*

$$Q(\overline{\mathcal{G}};x) = \overline{\theta}'\mathbf{x}$$

Denoting the Hermite basis by $\mathbf{h} \equiv (\mathbf{h}_0, \mathbf{h}_1, ...)$ *where* $\mathbf{h}_i$ *is the i-th Hermite polynomial.*
Then,

$$Q(\mathcal{G};x) = \overline{\theta}'\mathbf{h}.$$

Essentially, this means that, on a densely connected graph, we can calculate the matching polynomial of the complement graph, and then use the Godsil complement theorem to calculate the matching polynomial of the original graph by writing the same polynomial with respect to the basis formed by the Hermite polynomials.

The standard approach is to calculate the density of the graph, defined as $\frac{2|E|}{|V|(|V|-1)}$, and if it is above 0.5, to calculate the matching polynomial of the complement graph instead. This can in fact be applied recursively, so that reaching a node in the binary tree such that the density of the graph at that node is greater than 0.5, the resulting matching polynomial will be calculated as the Hermite polynomial of the complement graph.

4.6.1.1 Weighted Graphs

We can also extend the method to weighted graphs. In this case, the matching polynomial is *defined* via the edge removal recurrence (Cvetkovic et al., 1988) as:

$$Q(\mathcal{G};x) = Q(\mathcal{G}-e;x) - w(e)Q(\mathcal{G}-v_1-v_2;x) \tag{4.8}$$

where $w_{i,j}$ is the weight of the edge between nodes i and j. Note that this yields the same matching polynomial as in the vertex-removal recurrence (4.2).

In Section 4.7 we present an example of application of the weighted matching polynomial method to the real Wikipedia visitation data.

4.6.1.2 Large Graphs

The method described above exhibits exponential computational complexity in the number of nodes. We have found graphs that have suboptimal density (i.e., most nodes are connected to about half of all the nodes in a graph) to be very slow even at around $n \approx 30$. As a result it is necessary to find another approach to calculation of the matching polynomial that can be applied to larger graphs.

First, we present Lemma 3, originally presented without proof by Christopher David Godsil (2017):

Lemma 3 ((Christopher David Godsil, 2017)). *Let $\mathcal{G}$ be a graph with n nodes. Define an augmented graph $B = \overline{\bar{\mathcal{G}} \cup K_r}$, where $\bar{\mathcal{G}}$ is the complement of $\mathcal{G}$, and K_r is a complete graph on r vertices, for $r < n$. The number of perfect matchings on B is equal to $r! M_{(n-r)/2}$, where $M_{(n-r)/2}$ is the number of $(n-r)/2$-matchings on $\mathcal{G}$.*

Proof of Lemma 3. In the notation of the lemma, B is a graph that is $\mathcal{G}$ with r additional vertices, each of which is connected to every vertex in $\mathcal{G}$. Any perfect matching covers all the vertices of B by definition, and therefore must include a set of r edges, each covering one of the r additional vertices. The other side of each of these edges covers one of r of the vertices in $\mathcal{G}$. Hence, the remaining $n-r$ vertices are covered by a set of $\frac{n-r}{2}$ edges, which comprises a $\frac{n-r}{2}$-matching on $\mathcal{G}$. Fixing this $\frac{n-r}{2}$-matching on $\mathcal{G}$, there are $r!$ ways to permute the edges that

connect the new r vertices to the remaining r vertices in $\mathcal{G}$. Hence, there are $r!M_{(n-r)/2}$ perfect matchings on B. □

This hints at an approach to calculating the matching polynomial of a graph. As explained by Rudelson, Samorodnitsky, and Zeitouni (2016), the number of perfect matchings of an undirected graph can be calculated by taking the *hafnian* of its adjacency matrix:

Definition 24 (Hafnian of a matrix)**.** *The hafnian of a $2k \times 2k$ matrix A can be defined as*

$$haf(A) = \sum_{\sigma \in \tau} \prod_{i=1}^{2k} A_{i,\sigma(i)}$$

where τ is the set of derangements σ of $2k$ elements such that σ^2 is the identity.

The hafnian of a matrix is, similarly to the calculation of the permanent, not tractable (Valiant, 1979). However, we propose an approximation to this using the Barvinok estimator of the hafnian (Barvinok, 1999).

The Barvinok estimator is a Monte Carlo method that utilises the properties of the determinant to construct an unbiased estimator of the hafnian of a matrix. We present the unbiasedness of this estimator as a theorem:

Theorem 16. *Denote by A a matrix, and by Z a random, skew-symmetric matrix such that the upper triangular entries of Z are independent Gaussian random variables with mean 0 and variance 1. Define A_2 to be the matrix that contains the element-wise square root of the matrix A.*

Construct the estimator $\alpha(A) = det(Z \odot A_2)$. Then, $\mathbb{E}[\alpha(A)] = haf(A)$.

Proof. Proof in Appendix B. □

The Barvinok estimator of the hafnian is presented originally by Barvinok (1999), without proof of unbiasedness. We include a proof in order to clarify the approach to the construction of the estimator; see Appendix B.

4.6.1.3 Control variates

In testing, however, it was noticed that for more densely connected graphs, the resulting estimator was correspondingly more noisy. This arguably can lead to failure to maintain sensitivity for anomaly detection.

One approach is to again use Theorem 15 to calculate the hafnian of the complement graph, which will be less dense and therefore subject to less noise. However, this may require use of very large-order Hermite polynomials, whose coefficients become large enough to cause overflow problems even on modern 64-bit systems. Specifically, the coefficients of the Hermite polynomials are related to the double factorial, numbers (A001147 in the OEIS) which naturally grow factorially. The largest factorial that can be represented in a 64-bit integer is $20!$, so the Hermite polynomials of high order can end up with overflows on coefficients. As a result, we propose a control variate approach to the construction of the Barvinok estimator of the hafnian.

When constructing Monte Carlo estimators, it is often possible to improve the efficiency of the estimator by using a control variate. A control variate is a random variable that is correlated with the random variable of interest, and whose expectation is known. Such control variates can then be used to reduce the variance of the estimator

(Dellaportas and Kontoyiannis, 2012).

Assume we have a random variable $z \in \mathbb{R}^d$, where $z \sim F$ for some dimension d, and distribution function F. Suppose an arbitrary continuous function $g : \mathbb{R}^d \to \mathbb{R}$ whose expectation we would like to calculate. The standard Monte Carlo estimator would be to construct samples $z_i \sim F$ and take the sample mean to get an estimator $\hat{\mu}$:

$$\hat{\mu} = \frac{1}{N} \sum_{i=0}^{N} g(z_i)$$

If we take a different function $f : \mathbb{R}^d \to \mathbb{R}$ such that we know the expectation $\mathbb{E}_F[f(z)]$, we can construct a new estimator that uses the control variate f to create an improved estimator $\hat{\mu}^*$:

$$\hat{\mu}^* = \frac{1}{N} \sum_{i=0}^{N} (g(z_i) - a(f(z_i) - \mathbb{E}_F[f(z)])$$

where a is a coefficient chosen to minimise the variance of the estimator.

Essentially this coefficient is chosen to orthogonalise the control variate with respect to the random variable of interest. The variance of the estimator is then minimised; this is a property of least-squares estimators (Hayashi, 2000). Essentially, it replicates the result of applying the Gram-Schmidt process (Mayers, Golub, and Loan, 1986) to the random variable of interest and the control variate; such that all the variance in the estimator is from factors not accounted for in the control variate. Naturally, we can add as many control variates for which we have the exact expectation, to further improve the variance of the final estimator. A natural control variate for the estimator is the matrix permanent of the random matrix Z. The Barvinok estimator

control variate is written

$$\beta(Z) = \det(Z) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} Z_{i,\sigma(i)}$$

and by taking its expectation:

$$\begin{aligned}
\mathbb{E}[\beta(Z)] &= \mathbb{E}[\det(Z)] \\
&= \sum_{\sigma \in S_n} \mathbb{I}[\sigma \in \tau] \\
&= \#\{\text{permutations in } \tau\}
\end{aligned}$$

To see which permutations are in this set, we can use cyclic notation. The number of permutations of $2k$ elements is $2k!$. Writing a given permutation out, we note that the permutations σ such that $\sigma^2 = \mathrm{Id}$ can be written as a set of k composed 2-cycles; i.e. by wrapping pairs of elements in k pairs of brackets. Each of the $k!$ rearrangements of the brackets is equivalent; as well as the in-pair rearrangements of the paired elements. Thus, the resulting count of the permutations in τ is $\frac{(2k)!}{k!2^k}$.

4.6.2 Implementation

In order to facilitate the use of the matching polynomial in general graph machine learning tasks, we have a library, written in Rust (Matsakis and Klock, 2014) with Python bindings to allow for ease of use. The software is designed to better utilise the behaviour of modern CPUs to achieve high efficiency. The software is available at `https://github.com/wegreenall/matching_poly_lib`.

4.6.2.1 Binary representation of graphs

A standard way to conduct operations on graphs is via the adjacency matrix. However, the adjacency matrix in essence contains redundant information. Furthermore, it requires storage on the order of n^2 integers, each of which captures only a zero or one. In order to allow the CPU to conduct the necessary calculations as fast possible, the library represents graphs in a different format. Initial implementations using ready-made graph representations such as from the NetworkX library (Hagberg, Schult, and Swart, 2008) would regularly lead to memory overflows on my computer.

We represent a graph adjacency as a set of up to 64 64-bit integers. This is because the CPU operates directly on registers that are 64 bits in length. This means that the operations on the matrix rows or columns can be reduced to simple bitshifts or masks, utilising far fewer clock cycles. Specifically, we represent a graph as a set of integers, with leading bit set to 1. For example, the graph whose adjacency matrix is written

$$\begin{matrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{matrix}$$

is represented as the set of integers $(29, 14, 7, 2, 1)$.

As standard in construction of adjacency representations, the existence of a 1 in an off diagonal position (i, j) denotes an edge between the node i and the node j. In our representation, a bit set on

the "diagonal" (or as the leading bit) denotes the existence of a node in the graph.

Continuing our clarifying example, note that the binary representation of the integers $(29, 14, 7, 2, 1)$ is:

$$\begin{matrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{matrix}$$

This is the upper-triangle of the adjacency matrix above, with leading bits set to 1, marking that the node on that row is still in the graph. Removal of the i-th node requires zeroing of the i-th row, and AND-ing of all integers with $FFFF$ XOR $(1 << i)$, where $<<$ represents the left shift operator. Removal of an edge between nodes i and j requires just the latter operation on the i-th row, masked with $FFFF$ XOR $(1 << j)$.

In this way we can speed up the operations required for the calculation of the matching polynomial using the edge-removal recurrence (4.7) greatly; avoiding the overhead of maintaining a more complex graph representation as in e.g. the NetworkX Library (Hagberg, Schult, and Swart, 2008). This means our implementation is likely the state-of-the-art in calculating the matching polynomial of a graph.

4.7 Experiments

4.7.1 Synthetic data

To present the feature construction method for anomaly detection, we have both synthetic and real-world datasets. We generate random graphs using several "standard" models, selecting a fixed number of these graphs to be anomalous. The aim is to identify the anomalous graphs; i.e. the ones that have different structure to the others. Since the method is unsupervised, we do not require any labels for the anomalous graphs, and there is no formal "training" phase. The features for comparison are constructed from the Christoffel measure estimate for each graph. We compare the Christoffel measure estimate for each graph with a specific base measure estimate, and use the resulting distance to decide whether a graph is "anomalous" or not.

To create Figure 4.3(a), 400 graphs were generated, where each graph is a *complete* (i.e., fully connected) graph with a single random edge removed from each node; this is the "standard" graph. We then select 40 of these graphs to be anomalous, and set these to be complete graphs with 4 edges removed at random from each node. We use as a base measure the fully connected graph with no edges removed. An *increase* in the distance between the base measure estimate (as marked by an increased value of the MMD value) indicates that the graph is "further" from the fully connected graph.

In Figure 4.3(b), we use the same approaches to generation of the standard graphs as well as the anomalous graphs. The only conceptual difference is that now we compare the measure estimate for each graph with that of the *path* graph. This provides the interpretation that, when the MMD values *decrease*, the graph can be described as

"more path-like". It is clear from the diagram, comparing the locations of the true anomalous graphs and the spikes in the MMD values, that the negative spikes mark the graphs that are anomalously more path-like in the data set.

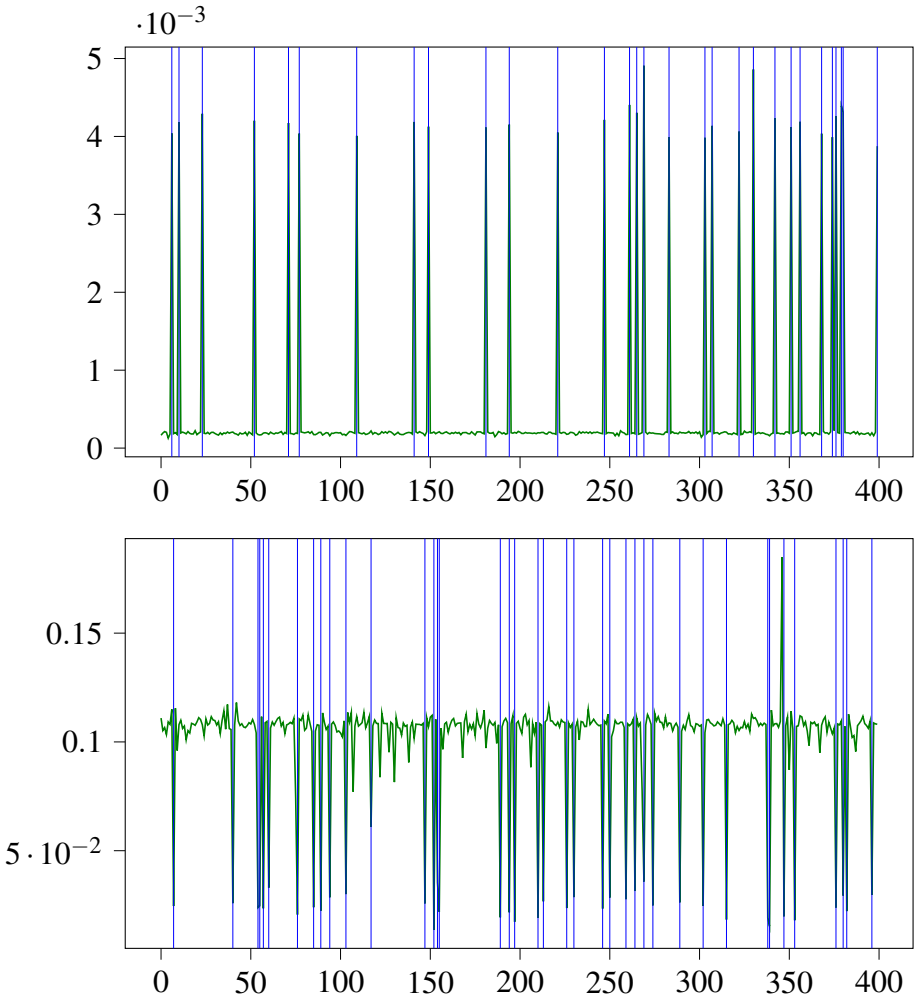

Figure 4.3: Application to a synthetic sequence of graphs. Standard graphs are generated as complete graphs, with a single, random edge removed from each node. Anomalous graphs are complete graphs with 4 edges removed at random from each node. Anomalous graphs are marked as blue vertical lines at the anomalous "indices", while green lines denote MMD values. Top: The base measure is the complete graph with no edges removed, so we interpret anomalies to be less "fully connected" than the standard graphs, since the MMD *increases* at anomalies. Bottom: The base measure is the path graph; this yields the interpretation that the anomalous graphs are more "path-like" than the standard graphs, since the MMD *decreases* at the anomalies.

4.7.1.1 Large Graphs

As described in Section 4.6.1.2, we require a special approach to larger graphs to be able to tame the complexity through appropriately constructed approximations. We present an example of this in Figure 4.4.

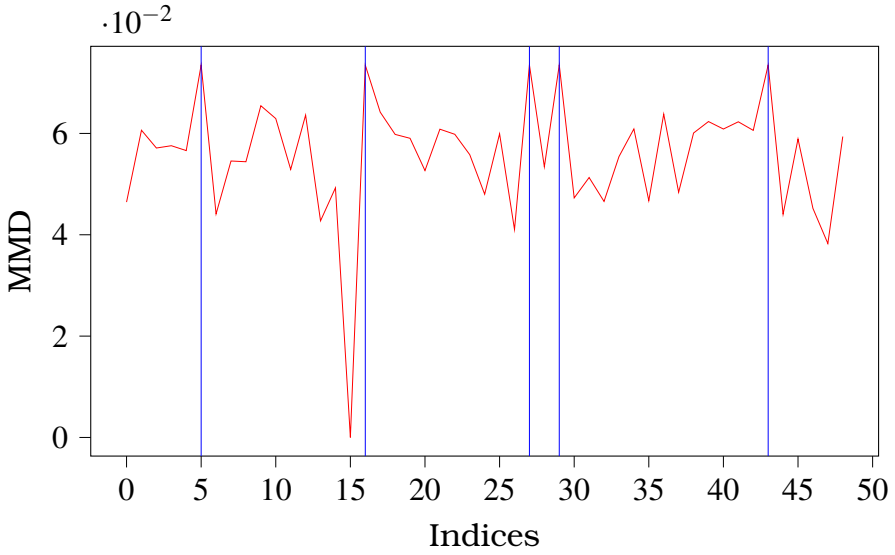

Figure 4.4: Application of the Barvinok estimator described in Section 4.6.1.2. The base measure is that for the fully connected graph; the base graphs are fully connected of size 80, and the anomalous graphs are path graphs of size 80 with 40 edges removed at random.

4.7.1.2 Cospectral Graphs

A further benefit to our approach is that it is in many cases able to capture anomalies that other methods cannot. To present this, we have an example of a constructed data set that exhibits an ability to observe anomalies in cospectral graphs. We use the approach taken by Christopher. D. Godsil and McKay (1982)(see example "a" in that paper) to construct cospectral graphs. The standard graph is a polygon graph of size $2k$ with an extra node added; this node is connected to k of the nodes of the polygon, in clockwise order. The

base comparison measure is the Christoffel measure estimate for this non-anomalous graph. The anomalous graphs are constructed in the same way, but the extra node is connected to k *random* nodes. The graphs constructed in this way are by construction cospectral (Christopher. D. Godsil and McKay, 1982). Since the graphs are cospectral, spectral methods are incapable of differentiating between any of these graphs. Furthermore, each of the graphs in the dataset exhibit the same connectedness, so averaging over node properties to summarise the graph also fails to detect anomalies. Whilst this example is deliberately constructed, it highlights the ability of the approach outlined in this chapter to capture certain anomalies that other methods will fail to capture.

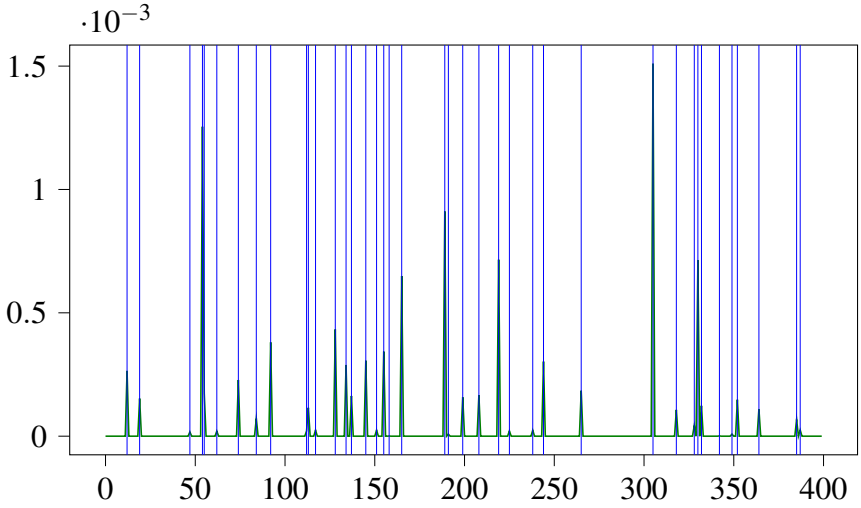

Figure 4.5: Application of the method to a sequence of cospectral graphs. All graphs are a polygon of size $2k$ with an added node. The added node is connected to k nodes in clockwise order (for standard graphs) or random order for anomalous graphs. Anomalous graphs are marked as blue vertical lines at the anomalous "indices", while green lines denote MMD values. All graphs are cospectral, so spectral methods are unable to differentiate between them. The base measure is the measure estimate for an element from the non-anomalous graph set.

4.7.2 Weighted Graphs

To show how the method can also be applied to weighted graphs, we use the wikivital mathematics dataset (UCI, 2021). This dataset records the daily visitor numbers to a large number of Wikipedia pages, with the general theme of topics in mathematics. In order to test this approach, a subset relating to statistics topics was selected, containing 29 nodes, and the number of visitors to each page was recorded for each day in the period from 16th March 2019 to 15th March 2021. The resulting dataset therefore is representable as a sequence of node-weighted graphs, where the weight of each node is the number of visitors to the corresponding Wikipedia page. To capture this weighting in a format that matches the requirements of the method, the weight of the nodes adjacent to a given edge are averaged, and the edge is ascribed the resulting value. Because these numbers can be large, we take the logarithm of the weight of each edge. The result is a sequence of edge-weighted graphs, to which we apply the feature construction method.

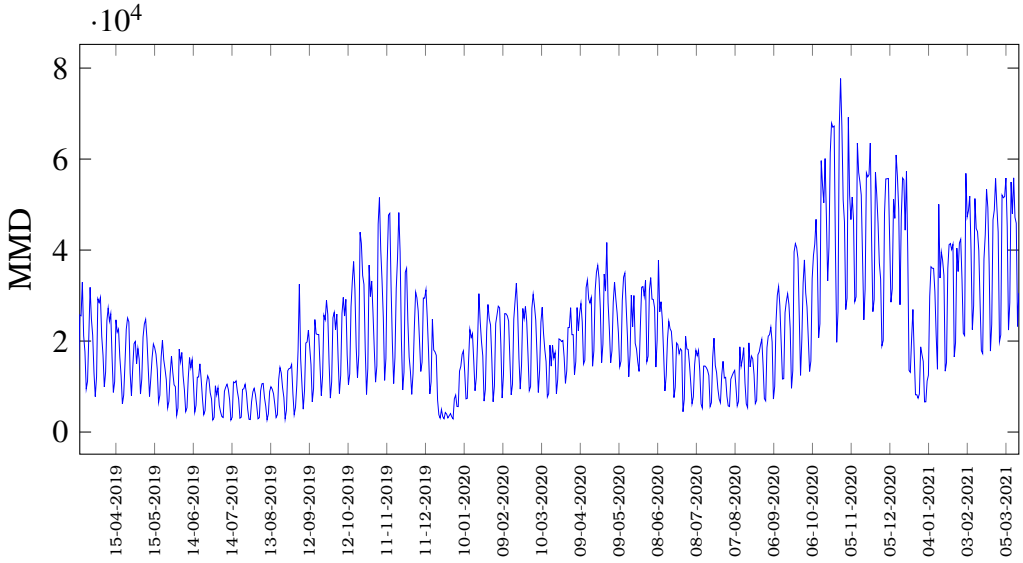

Figure 4.6: Wikipedia statistics article graph embeddings

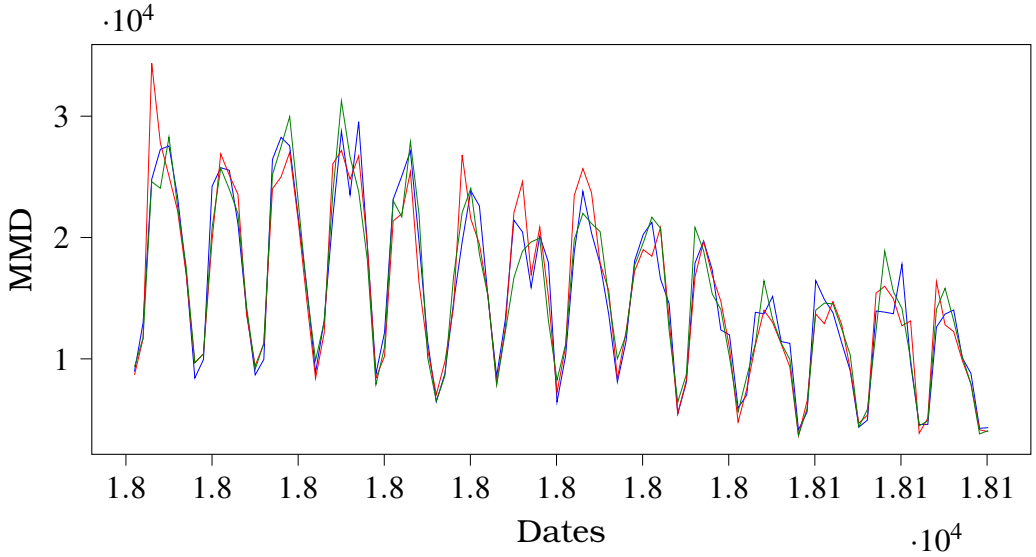

Figure 4.7: Comparison of base measure estimates for Wikivital Stats articles graph data. Blue: The base measure used is the standard base measure for the Wikipedia data; i.e. the implicit measure estimate for the connectivity graph, with no weights. as a result it is equivalent to the Wikipedia graph with e visitors, since we use log weights as the metric to avoid excessive weight values. Red: The base measure used is the measure estimate corresponding to a path graph on the same nodes. We can thus interpret larger "red" values as being more path-like. Green The base measure used is the measure estimate corresponding to a complete graph on the same nodes. We can thus interpret larger green values as being like more "fully connected" graphs.

The result can be seen in Figure 4.6. The base measure is the Christoffel measure estimate for the graph with no visitors. As a result, higher MMD values indicate that the graph is "further" from the graph with no visitors; this provides the interpretation that the graphs have more visits when this value is higher. The weekly periodic behaviour in visitor numbers is clearly visible, as well as less busy periods in summer and at Christmas. Furthermore, different choices of base measure will yield different interpretations of the MMD values. Examples of this are presented in Figure 4.7.

4.8 Conclusion

We have presented a novel method for intepretable feature construction and anomaly detection on sequences of dynamic graphs. We have shown that this method is able to detect anomalies in cases that spectral methods cannot be used, and also flexibly captures various different forms of anomaly, depending on the chosen base measure. Furthermore, the method allows one to test different "base cases" for a given sequence of graphs without recomputing some new aspect of the graphs in the dataset.

One of the downsides to this method is that it is that the computational complexity is quite high. This is endemic to methods that attempt to utilise the structure of the graph on the whole (Akoglu, Tong, and Koutra, 2014), and our Barvinok estimator method attempts to go someway towards alleviating this problem. It is likely that further work in this direction could lead to more effective expansion for larger graphs, or perhaps applying the method herein after a dimension reduction step.

Another downside is the lack of rigour in the arguments behind interpretability of the method. Specifically, description of graphs as more e.g."path-like" is somewhat vague, and its evidence is empirical rather than theoretical. Without formal theory of the perturbation of orthogonal polynomials, it is questionable whether a graph truly is "path-like". However, the theory on perturbations of orthogonal polynomials is extremely niche and not very accessible. Future work will aim to bring these theoretical tools to bear on the problem.

Further work should also focus on understanding of other graphs with orthogonal polynomial measures. A natural direction of research

would be to look into generating graphs with a given (orthogonal) matching polynomial. Favard's theorem means that there are a continuum of orthogonal polynomials, and it seems reasonable that there should be a corresponding continuum of graphs with orthogonal matching polynomials, allowing for arbitrary "landmark" association; i.e. deliberate construction of graphs with a given base measure.

Chapter 5

Superposition Gaussian Cox Processes

5.1 Introduction

Stochastic point process models such as the Gaussian Cox process (Cox, 1955) suffer from an intractability in the likelihood function. This comes from the fact that the intensity is modelled as a stochastic Gaussian process. The Poisson process, given the realisation of the stochastic intensity, has an integral in its likelihood that integrates out uncertainty induced by the region of space observed but not containing points. To get a valid marginal likelihood function, the practitioner must integrate out both this uncertainty and the uncertainty resulting from the (Gaussian) stochastic process sample. As a result, it requires a double integration; first integrating over the intensity function, and then over the uncertainty in the intensity function.

This intractability has yielded much literature that aims to overcome it. Many approaches centre on the utilisation of specific properties of the link function to yield a tractable version of the corresponding

integrals. The link function is used to transform the latent Gaussian process to a positive-valued function as required by the standard definition of the Gaussian Cox process.

If the link function is the square function $\ell(x) = x^2$, the process is known as a *permanental* process (McCullagh and Møller, 2006); this is because its k-point correlation function is the permanent of an appropriately defined matrix. Such a process exploits Hilbert space methods which redefine the corresponding integral as the norm of the latent function (Flaxman, Teh, and Sejdinovic, 2017; Walder and Bishop, 2017) and yields tractability via the representer theorem (Kimeldorf and Wahba, 1970; Schölkopf, Herbrich, and Smola, 2001) which provides a way to express the sought-after function as a specific sum of a sequence of kernel functions.

The other classic approach is the log-Gaussian Cox process (Møller, Syversveen, and Waagepetersen, 1998), which uses the exponential function $\ell(x) = \exp(x)$ as its link function. The resulting process yields a very simple form for the k- point correlation function. However, estimation focuses on discrete approximation of the likelihood integrals, construction of gradient estimates (Choiruddin et al., 2020) or more complex MCMC approaches (Peter J. Diggle et al., 2013; Taylor and Peter J Diggle, 2014).

Other approaches propose different link functions, such as the sigmoidal Gaussian Cox process (Adams, Murray, and MacKay, 2009; Donner and Opper, 2018), focusing on Bayesian MCMC approaches that require e.g. a Laplace approximation of the posterior; or utilise variational inference to approximate the posterior (Lloyd et al., 2015; Aglietti et al., 2019).

The methods described above can also induce certain unsatisfac-

tory artefacts in regions of the the estimated Cox process intensity model; see John and Hensman (2018) for details. A clear example of such bias is that of *nodal lines* (John and Hensman, 2018) which are regions (curves, when mapping spatial data) where the intensity is zero along arbitrary lines. This does not map to specific information about the behaviour of the point process. We aim to circumvent these problems by a slight redefinition of the Cox process that provides such artefacts with interpretable meaning.

A further problem that has not been adequately handled in the literature is the natural lack of identifiability in the standard Gaussian Cox process model. Specifically, the intensity function is modelled as a stochastic process, and the data are viewed as points from an inhomogeneous Poisson process conditional on a realisation of this stochastic intensity function. Since the process is latent, and ostensibly a single sample has been observed, there is nothing in the data that allows the practitioner to differentiate between a highly random, zero-mean intensity realisation, or a realisation that has a strong prior mean component with a small random component. Our approach is able to handle this explicitly, via selection of a prior parameter that regulates the prior weight ascribed to either variation or mean, since only problem or domain knowledge can help to distinguish between these settings unless multiple realisations have been observed.

Finally, the approaches above have been utilised to construct stochastic classification models (McCullagh and Jie Yang, 2006a; J. Yang, Miescke, and Mccullagh, 2012; Matthews and Ghahramani, 2014). The approach taken in this small but concise literature is to model the classification problem as viewing realisations of different marked point processes. The items to be classified (e.g., images of dogs

or cats) can be modelled as events in an appropriately defined space, to which point process models can be applied. The benefit of such an approach is that the classification model yields exact probabilistic predictions, in contrast to standard approaches where uncertainty quantification is either done by mapping predictions to the simplex to yield predictions that have the *form* of probability distributions; or more recently through conformal prediction (Lei and Wasserman, 2014). In our case, uncertainty quantification is inherent to the output of the classifier. As we show in this chapter, it is also capable of capturing highly non-linear behaviour in classification problems.

We define the superposition Gaussian Cox process model by assuming that the intensity function is a superposition of point processes defined on regions over which the intensity is positive. The great advantage of such an approach is that by avoiding the non-linear link functions of the log-Gaussian and permanental Cox processes we can directly exploit the rich theoretical developments of the Gaussian processes. In particular, we can approximate the latent intensity process via a linear combination of basis functions in a space of square-integrable functions avoiding the double-intractability problem. This allows us to solve the identifiability of the Gaussian Cox process models by viewing it as a Bayesian inference problem with informative prior specifications. A further important advantage is that by achieving fast and reliable inference of the intensity process, we can also construct a stochastic classification model.

In summary, our new modelling and inferential strategy offers a new approach to inference for non-homogeneous point processes that: (i) sidesteps entirely the intractability of the likelihood; (ii) produces an extremely computationally efficient inference procedure; (iii) requires

no training or expensive MCMC inference phase; (iv) directly handles the lack of identification in point process models by allowing a prior specification of the weighting between mean and variance in the latent modulating Gaussian Cox process; and (v) yields a stochastic classification model that produces direct probabilistic predictions avoiding *ex post* uncertainty quantification.

At the time of writing, the material in this chapter forms the base of a paper currently under submission at a machine learning conference. Code for generation of the diagrams comparing the approach outlined in this chapter to other methods was written by my co-author on the paper submitted to said conference, Apostolis Kapetis. All other parts are my own work.

5.2 Motivation

We assume that we have available data $\{x_i\}_{i=0}^N$ consisting of N points in $\mathcal{X} \subseteq \mathbb{R}^d$, and that the data are generated according to an inhomogeneous point process, and we aim to model uncertainty over the corresponding intensity by placing a prior distribution over the intensity function using a Gaussian Cox process.

The standard approach, as mentioned above, is to use the following model for the data-generating process.

Definition 25 (Gaussian Cox Process)**.** *A Gaussian Cox Process is a stochastic Poisson process, for which the density of the intensity measure is a transformed sample function from a Gaussian process*

(GP), i.e.

$$\mathbb{P}\{N(S) = n|\psi\} = \frac{\Psi(S)^n}{n!}e^{-\Psi(S)},$$
$$\ell(\psi(s))^{-1} \sim \mathcal{GP}(s(\cdot), k(\cdot,\cdot))$$

where $\ell(\cdot)$ is an invertible link function; ψ is the intensity function; Ψ is the intensity measure; $s(\cdot)$ represents the prior mean function of the Gaussian process; and $k(\cdot,\cdot)$ its covariance function or kernel; $S \subset \mathcal{X}$ the set containing the observed points; and $N(S)$ denotes the number of points found in S.

The choice of link function naturally has major ramifications regarding the behaviour of samples from this model, and induces certain biases that may be undesirable. For example, the standard link functions, namely the exponential function $\exp(\cdot)$ and the square function, both enforce positivity of the resulting intensity function. This does yield a valid intensity function, but it also induces a bias in the sense that points at which the latent function f goes negative induces arbitrary zeroes in the intensity function.

Given such a model, the practitioner aims to select appropriate hyperparameters of the kernel in order to get reasonable behaviour from the model. This is usually achieved by maximising the marginal likelihood of the data, having integrated out the latent function f. The log-likelihood function of an inhomogeneous Poisson process is given by

$$\log \mathcal{L}(x_1, x_2, ..., x_N) = \sum_{i=0}^{N} \log \psi(x_i) - \int_S \psi(s) d\nu(s)$$

where ν is usually taken to be the Lebesgue measure. Since the

marginal likelihood also requires a second integration over the stochastic intensity ψ, the marginal likelihood in such models is often referred to as *doubly*-intractable. This motivates the need for a new inferential strategy.

5.3 Method

We propose a modification of the standard Gaussian Cox process that allows us to utilise alternative methods to estimate the hyperparameters. We call our method for estimation the Orthogonal Series Gaussian Cox Process (OSGCP), and we call the model the superposition Gaussian Cox process.

Definition 26 (Superposition Gaussian Cox Process)**.** *Assume a compact measure space $(\mathcal{X}, \mathcal{F}, \nu)$, with measure ν. Denote by f a sample function from a Gaussian process with Mercer kernel k such that $k(0,0)'' > 0$, and a mean function $s(\cdot)$. Define the sets $\mathcal{S} = \{S_1, S_2, ..., S_p\}$ where $S_i \subseteq \mathcal{X}$ are open and disjoint and are defined such that $f(x) > 0$ for all $x \in S_i$. Since f is continuously differentiable, we can define a series of point processes on each of S_i by an intensity function $\psi_i(x) = f(x)\mathbb{I}[x \in S_i]$ with corresponding intensity measure $\Psi_i(A) = \int_A \psi_i(x) d\nu$. Therefore, there are now p disjoint point processes. A superposition Cox process is the superposition of these disjoint point processes, and the superposition theorem (J. F. C. Kingman, 1975) states that this superposition is itself a Poisson process with intensity measure $\Psi(A) = \sum_{i=1}^{p} \Psi_i(A)$ for $A \in \mathcal{X}$.*

A clarifying example can be seen in Figure 5.1. The main benefit of the superposition Gaussian Cox process formulation is that it allows us to sidestep the issue of the doubly-intractable likelihood.

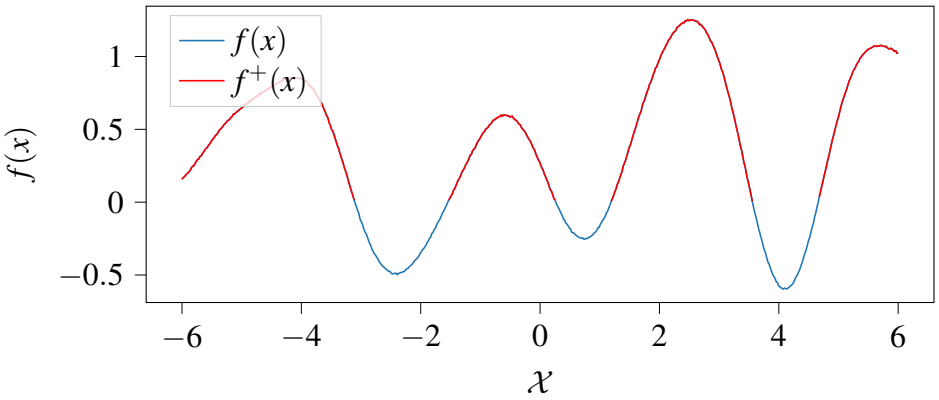

Figure 5.1: Examples of the Cox process model outlined in this chapter. Note the difference between this and using $\max(0,\cdot)$ as a link function. We denote the Gaussian process sample as $f(x)$ and the superposition sample as $f^+(x)$.

The aim is to treat the problem as inference over the properties of the full latent function f. Furthermore, negative values of the latent function, when sampled, are not forced to positive values as they are not transformed by a link function as in the usual approach to Cox processes; instead they directly represent the information that the point process realisation yielded no points in that region, and avoid the artificial "nodal lines" phenomenon (John and Hensman, 2018).

A consequence of this is that the inference can be sped up greatly. Just as the permanental process allows for a specific interpretation of the RKHS norm of the intensity as the integral component in the model likelihood, the superposition Gaussian Cox process allows for an alternative interpretation of the model likelihood. We use this to construct an approximate likelihood model that offers a conjugate Bayesian approach to the problem. As a result, estimation and inference can be performed in a fraction of the time, and it is this computation benefit that motivates the use of the superposition Gaussian Cox process; details of this speedup can be seen in Section 5.6.

Associated with the kernel k is a reproducing kernel Hilbert space

of functions

$$\mathcal{H}_k = \left\{ g : g = \sum_{i=0}^{\infty} g_i \phi_i; \sum_{i=0}^{\infty} \frac{g_i^2}{\lambda_i} < \infty \right\},$$

where ϕ_i are the orthonormal eigenfunctions of k with corresponding eigenvalues λ_i (Rasmussen and C. K. I. Williams, 2018). Assuming that $\{\phi_i\}_{i=0}^{\infty}$ form a basis in $\mathcal{L}^2(\nu)$, the space of square-integrable functions with measure ν, we can define

$$\psi_i(\cdot) = \sum_{j=0}^{\infty} \psi_j^{(i)} \phi_j(\cdot), i \in \{0, 1, ..., p\}, \tag{5.1}$$

for $\psi_j^{(i)} \in \mathbb{R}$, by the Karhuenen-Loève theorem (Kanagawa et al., 2018). Thus, the intensity measure Ψ has an intensity function

$$\begin{aligned} \psi(\cdot) = \sum_{i=0}^{p} \psi_i(\cdot) &= \sum_{i=0}^{p} \sum_{j=0}^{\infty} \psi_j^{(i)} \phi_j(\cdot) \\ &= \sum_{j=0}^{\infty} \sum_{i=0}^{p} \psi_j^{(i)} \phi_j(\cdot) \\ &= \sum_{j=0}^{\infty} \xi_j \phi_j(\cdot), \end{aligned} \tag{5.2}$$

where in the last sum, $\xi_j = \sum_{i=0}^{p} \psi_j^{(i)}$ is the sum of the coefficients for the given eigenfunction ϕ_j in each of the disjoint intensities.

Definition 27 (Orthogonal coefficient estimator)**.** *Suppose that we have a sequence of orthonormal functions $\{\phi_i\}_{i=0}^{\infty}$, $\phi_i : \mathcal{X} \to \mathbb{R}$, that form an orthonormal basis in $\mathcal{L}^2(\nu)$. Assume that there is a sample $\{x_i\}_{i=0}^{N}$ from a point process available, with intensity function f. An estimator We define the* orthogonal coefficient estimator *for basis function j in the representation of the intensity function f with respect to the basis*

functions $\{\phi_i\}_{i=0}^{\infty}$ *as*

$$\hat{\xi}_j = \sum_{i=0}^{N} \phi_j(x_i). \tag{5.3}$$

We first present Campbell's theorem (J. Kingman, 2005) as a lemma. Notation is taken almost directly from (J. Kingman, 2005).

Lemma 4 (Campbell's Theorem, (J. Kingman, 2005))**.** *Suppose a Poisson point process on* $\mathcal{X}$ *with intensity measure* Ψ*, and with a sample of points* $\{x_i\}_{i=1}^{N}$*. Let* g *be a measurable function on* $\mathcal{X}$*. Then, the sum* $\sum_{i=0}^{N} g(x_i)$ *converges if and only if*

$$\int_{\mathcal{X}} \min(|g(x)|, 1) d\Psi(x) < \infty.$$

Then,

$$\mathbb{E}\left[\sum_{i=0}^{N} g(x_i)\right] = \int_{\mathcal{X}} g(x) d\Psi(x). \tag{5.4}$$

and

$$\mathbb{V}\left[\sum_{i=0}^{N} g(x_i)\right] = \int_{\mathcal{X}} g(x)^2 d\Psi(x).$$

We now show how Campbell's theorem allows us to construct an unbiased estimator for the intensity function coefficients.

Theorem 17. *Assume* Ψ *is the intensity measure for a superposition Gaussian Cox process, with intensity function* ψ*, and denote by* $\{\phi_i\}_{i=0}$ *the sequence of basis functions* ϕ_i *as in Definition 27. Then,*

$$\psi(x) = \sum_{j=0}^{\infty} \xi_j \phi_j(x).$$

Denote by $\hat{\xi}_j$ the orthogonal coefficient estimator for ξ_j, as in Definition 27. Then,

$$\mathbb{E}\left[\hat{\xi}_j\right] = \xi_j.$$

Theorem 17 shows that the superposition Gaussian Cox process yields a valid orthogonal series estimator (Kimeldorf and Wahba, 1970; Kronmal and Tarter, 1968), and that the resulting estimator can be written as a sum of the eigenfunctions associated with the kernel k.

The key to our proposed method is the switch of summation order in (5.2), valid by the Karhuenen-Loève theorem. Its relevance is that in our case, there is no need to construct or identify the sets S_i, $i = 1, 2, ...$ as would be necessary if applying our method to density estimation. In that case the corresponding component supported on S_i would need to be weighted by the number of points in S_i; different choices of S would yield different estimators. In our case the inference is the same regardless of the specific structure of the sets $\{S_i\}_{i=0}^{p}$.

5.3.1 Model Setup

We now present a method for learning a superposition Gaussian Cox process model. For a given kernel k, the method above requires the construction of the basis functions of k with respect to the implicit measure ν on the measure space $(\mathcal{X}, \mathcal{F}, \nu)$. As noted elsewhere in the literature (Zhu et al., 1998; Fasshauer, 2012b), this is in general intractable. Normally, one selects a kernel, and retrieves the appropriate basis functions and eigenvalues given the measure ν. However, rather than pre-select a kernel and aim to find its eigenfunctions and eigenvalues, we select an appropriate orthonormal basis and allow

the model to learn the eigenvalues freely. This is a similar approach to that presented in Chapter 3.

We assume that the data are a finite subset of $\mathcal{X}$, and we consider the $\mathcal{GP}(s(\cdot), k(\cdot,\cdot))$ where the kernel has Mercer decomposition $k(x,y) = \sum_{i=0}^{\infty} \lambda_i \phi_i(x)\phi_i(y)$. Since the superposition Cox process does not require the Hilbert space to be infinite dimensional, we will restrict to a finite order m and work in the truncated Hilbert space. Assume that we have available the functions $\{\phi_i\}_{i=0}^{m}$. Following Theorem 17 and (5.1), we can construct $\hat{\psi}$, an orthogonal series estimator for the intensity function.

Here it is important to note that the Gaussian Cox process modelling problem is ill-posed. Specifically, it is not generally possible to differentiate points generated by a point process with a random intensity that has a specific mean function $s(\cdot)$ from points generated by a point process with a zero-mean Gaussian process generating its intensity function. To clarify this, suppose a Gaussian process is such that its mean function can also be written as a sum of the basis functions. Then, by the linearity of Gaussian random variables and the Karhunen-Loève theorem, the latent Gaussian process sample can be written as

$$f(x) = \sum_{i=0}^{\infty} \theta_i \phi_i(x) = \sum_{i=0}^{\infty} (s_i + \sqrt{\lambda_i} z_i)\phi_i(x) \tag{5.5}$$

for mean coefficients s_i and kernel eigenvalues λ_i. Here, θ_i denotes a random coefficient for basis function ϕ_i, z_i is a standard-normally distributed random variable, and we assume the coefficients are such that all sums converge.

Given a single realisation of θ_i, it is not possible to separately

identify s_i from $\sqrt{\lambda_i} z_i$. This corresponds to the intuitive fact that, for a single realisation of a Gaussian process, we cannot tell whether this realisation was the result of small-valued random coefficients and large mean coefficients, or large-valued random coefficients and small mean values. The former case implies that the realisation generating the observations is representative in its *location*; the latter implies that the realisation is representative in its *length-scale*. It is naturally impossible to distinguish between these two extremes given only a single realisation, and this must be put down to prior understanding of the problem setting.

The natural choice to handling this lack of identifiability is a conjugate Bayesian approach, which will weight values according to prior belief between the "large random coefficient" view and the "small random coefficient" view.

5.3.2 A Bayesian approach

To construct a Bayesian approach to this problem, we first note that, by Theorem 17, the orthogonal coefficient estimators are unbiased for the coefficients of each basis function in the given realisation of f. They also exhibit observation noise as a result of the variance in the specific locations of the point process; for a given realisation of f, there are many different possible realisations of the inhomogeneous Poisson process. We assume that the data generating mechanism that produced $\hat{\xi}_i$ is a set of linear models of the form

$$\hat{\xi}_i = s_i + \sqrt{\lambda_i} z_i + \sigma_i \varepsilon_i \; i = 0, 1, ..., m, \tag{5.6}$$

where $\sigma_i\varepsilon_i$ is mean-zero observation noise with standard deviation σ_i and the unknown vector of parameters is $(s_i, \lambda_i, \sigma_i)$. Clearly, this is a non-identifiable model but it will produce proper posterior distributions if we place informative priors on the parameter vector.

The distribution of $\sigma_i\varepsilon_i$ is not available since it depends on the unknown f. We assume that this distribution is Gaussian. We tested this normality assumption with a small experiment in which we generated data from the ground truth intensities exhibited in Figures 5.2, 5.3, and 5.4. Kolmogorov-Smirnov tests against normality were performed for 20 basis functions from 5000 realisations of the Poisson process and only one sample rejected the null at 5%. See Section 5.4 for more details.

The assumed Gaussianity of the observation noise allows us to construct the likelihood function of (5.6) as a product of Gaussians since each $\hat{\xi}_i$ follows a Gaussian with mean s_i and variance $\sigma^2 + \lambda_i$. The conjugate prior for the mean and variance in a standard Gaussian linear regression model is a Gaussian-inverse Gamma distribution $\mathcal{N} - \Gamma^{-1}(\mu, \eta, \alpha, \beta)$; see, e.g. Bernardo and Smith (2009). This approach means that inference is essentially instant, given that it merely requires evaluation of the corresponding posterior mean values given the observation data. For each of the orthogonal coefficient estimators we have one observation. The corresponding hyperparameter updates are therefore $\mu_i' = \frac{\eta_i\mu_i + \hat{\xi}_i}{\eta_i + 1}$, $\eta_i' = \eta_i + 1$, $\alpha_i' = \alpha_i + \frac{1}{2}$, and $\beta_i' = \beta_i + \frac{\eta_i}{\eta_i + 1}(\hat{\xi}_i - \mu_i)^2$, with posterior means given by

$$\begin{aligned} \mathbb{E}[s_i] &= \frac{\eta_i\mu_i + \hat{\xi}_i}{\eta_i + 1} \\ \mathbb{E}\left[\lambda_i + \sigma_i^2\right] &= \frac{\beta_i'}{\alpha_i' - 1}. \end{aligned} \tag{5.7}$$

All that remains is selection of the parameters of the prior density. We set the prior mean parameter μ_i to zero as an uninformative choice. The prior interpretation of α is that 2α is the number of prior observations, which exhibited non-normalised sample variance equal to 2β. For identification of the three parameters s_i, λ_i and σ_i, we assume at minimum three observations, so we set $\alpha_i = \frac{3}{2}$ following the minimum necessary sample principle of Novick and W. J. Hall (1965). For β_i we can utilise the fact that we have an estimator of the observation noise as a result of Campbell's theorem (Lemma 4) and provide an empirical Bayes estimator. Thus, we can construct an unbiased estimate of the variance of the orthogonal coefficient estimator with $\hat{V}(\hat{\xi}_i) = \sum_{j=0}^{N} \phi_i^2(x_j)$, and we set $\beta = \hat{V}(\hat{\xi}_i)$. The unbiasedness of this estimate is a direct result of Campbell's theorem being applied to ϕ^2. Under this description of the prior specification, this is equivalent to posterior values with $\beta = 0.0$. Shifting the posterior mean estimator for the variance to get an estimator of the eigenvalues yields

$$\hat{\lambda}_i = \frac{\beta_i'}{\alpha_i' - 1} - \sum_{j=0}^{N} \phi_i^2(x_j).$$

Substituting in the hyperparameter updates yields the following elegant equations for posterior mean estimates of the mean coefficients and the eigenvalues:

$$\hat{s}_i = \frac{\hat{\xi}_i}{\eta_i + 1}, \; \hat{\lambda}_i = \frac{\eta_i}{\eta_i + 1} \left(\hat{\xi}_i\right)^2 \tag{5.8}$$

which clarify the roles of the hyperparameter η_i as weighting between the two possible prior settings. In our numerical examples we chose a fixed $\eta_i = \eta$ for all i. The "posterior mean" presented in Figure 5.2 is

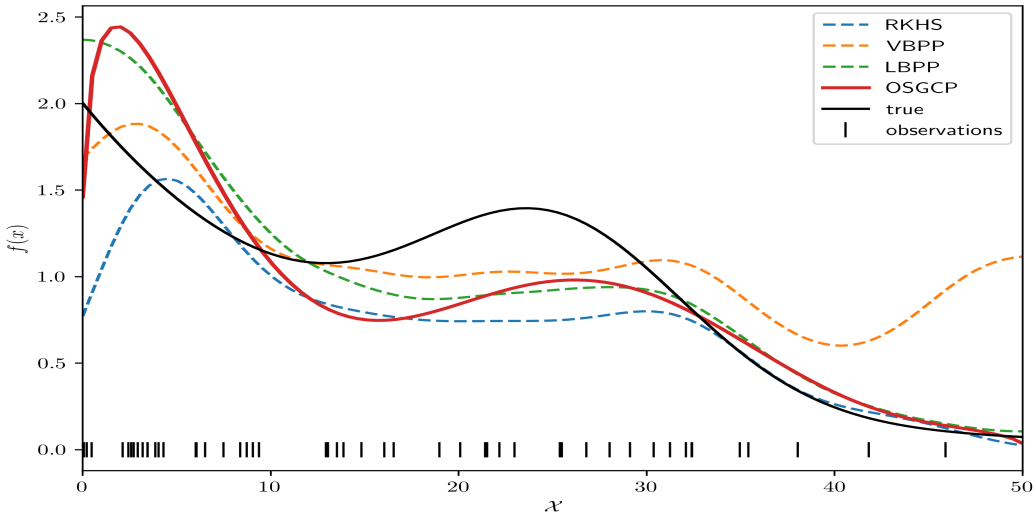

Figure 5.2: Comparison of 4 methods on synthetic function $\lambda_1(x)$ in Adams, Murray, and MacKay (2009). The presented curve is the posterior mean estimate in each case. OSGCP is constructed using $\eta = 0.12$, $\alpha = 1.5$, $\beta = 2.0$, $\mu = 0$, and $m = 8$. See Section 5.6 for details on how this figure was constructed.

constructed as a sum of the basis functions using equation (5.7) as the coefficients.

Turning now to the interpretation of the η_i parameters, we note that the relevance of the mean component and the eigenvalue (or variance) component of the latent process is likely to depend on the problem setting. For example, in looking at forestry data, it is likely that one aims to learn how the latent intensity changes over a given region of space, and take this to inform the change one can expect elsewhere. It does not seem of interest to learn the posterior mean of where the trees actually are, unless for example the forester is of the belief that the trees will grow back in roughly the same locations. On the other hand, in a classification setting as described in Section 5.7, the object of interest is the location of the intensity, and it is assumed that the corresponding intensity likely has low variance and resides close to its mean, so that the information learnt from the classifier is useful for labelling new observed items.

5.4 Basis function coefficient Gaussianity

We first present a simple experiment to justify our approach to modelling the basis function coefficient observation noise using a Gaussian distribution. In order to justify our approach to modelling the basis function coefficient observation noise in this way, we used Kolmogorov-Smirnov tests against the normal distribution. For each of the synthetic functions, we generate 5000 samples from the corresponding intensity function with the standard thinning approach. We then compare the distribution of the coefficients for each of 20 basis functions to see if they are statistically different from the Gaussian distribution. The basis functions are of the form $\phi_i = c_i P_i(2x-(a+b)/(b-a))w^{1/2}(x)$, where P_i denotes the i-th Chebyshev polynomial of the second kind; $w(x) = \sqrt{1-x^2}$ denotes the weight function; and c_i is a normalising coefficient based on the moments of the Wigner semicircle distribution, so that the orthonormality condition $\int_a^b \phi_i(x)\phi_j(x)dx = \delta_{ij}$ is satisfied. This makes $\{\phi_i\}_{i=0}^m$ an orthonormal set on $(a,b) \subset \mathbb{R}$. The results are found in Table 5.1, for the three synthetic intensity functions described in the text; λ_1, λ_2 and λ_3 respectively. Each row of each table presents the KS-statistic and the p-value against a normal distribution for the value of the orthogonal coefficient estimator, calculated for 5000 different realisations of the Poisson process using the respective table's corresponding intensity function.

5.5 Experiments

In this section we present both synthetic- and real-data experiments exhibiting the method.

Table 5.1: KS-test results for synthetic function basis coefficients (see Section 5.6 for explanation and functional form.)

	λ_1		λ_2		λ_3	
I	KS-STATISTIC	P-VALUE	KS-STATISTIC	P-VALUE	KS-STATISTIC	P-VALUE
0	0.0160	0.1553	0.0079	0.9145	0.0216	**0.0184**
1	0.0108	0.5986	0.0100	0.7003	0.0087	0.8397
2	0.0072	0.9554	0.0108	0.6013	0.0128	0.3823
3	0.0093	0.7765	0.0070	0.9643	0.0089	0.8239
4	0.0130	0.3672	0.0187	0.0597	0.0097	0.7303
5	0.0110	0.5808	0.0101	0.6779	0.0075	0.9422
6	0.0075	0.9404	0.0084	0.8651	0.0090	0.8085
7	0.0068	0.9719	0.0065	0.9826	0.0080	0.9026
8	0.0104	0.6514	0.0116	0.5118	0.0128	0.3861
9	0.0063	0.9883	0.0096	0.7430	0.0099	0.7039
10	0.0084	0.8676	0.0113	0.5410	0.0139	0.2889
11	0.0084	0.8732	0.0084	0.8678	0.0090	0.8117
12	0.0103	0.6569	0.0108	0.5953	0.0050	0.9996
13	0.0075	0.9411	0.0066	0.9817	0.0071	0.9602
14	0.0083	0.8795	0.0098	0.7203	0.0048	0.9998
15	0.0094	0.7648	0.0073	0.9518	0.0080	0.9035
16	0.0072	0.9541	0.0069	0.9703	0.0092	0.7878
17	0.0070	0.9647	0.0063	0.9886	0.0106	0.6264
18	0.0077	0.9294	0.0089	0.8168	0.0063	0.9888
19	0.0074	0.9469	0.0163	0.1378	0.0093	0.7778

5.5.1 Synthetic Data

In the following experiments, we run three different one-dimensional models, and present the effect of the η parameter on the model. In Figures 5.2, 5.3, and 5.4, we present three examples of application of the model to synthetic data. The examples are taken from the literature (Flaxman, Teh, and Sejdinovic, 2017; Walder and Bishop, 2017).

5.6 Comparison

We compare our method to Variational Bayesian Point Process (VBPP) (Lloyd et al., 2015), Laplace Bayesian Point Process (LBPP) (Walder and Bishop, 2017) and Reproducing Kernel Hilbert Space method

(RKHS) (Flaxman, Teh, and Sejdinovic, 2017). For the methods relying on the use of inducing points (LBPP and VBPP), we use 32 inducing points. For both VBPP and RKHS we use the squared exponential kernel, while for the LBPP we use the cosine based kernel, provided by the authors. Since both LBPP and VBPP infer kernel's hyperparameters via optimization, we choose initialization points matching the properties of the ground truth intensity function i.e the scale and the smoothness. For LBPP, following Rasmussen and C. K. I. Williams (2018), we update the posterior mean at each optimization step for the kernel hyperparameters. For the RKHS method, following the authors, hyperparameters are selected via cross validation.

In all cases, our basis functions are of the form $\phi_i(z) = c_i P_i(z) w^{1/2}(z)$, where $z = (2x - (a+b))/(b-a))$, for domain lower and upper bounds (a,b), where P_i denotes the i-th Chebyshev polynomial of the second kind; $w(x) = \sqrt{1-x^2}$ is a weight function; and c_i a normalising coefficient derived from the inner product on the Chebyshev polynomials. Multi-dimensional bases are constructed from the tensor product of these functions $\phi_i, i = 0,1,2,...m$ with $m = 15^d$ for dimension d.

5.6.1 Synthetic Datasets

For generating the synthetic datasets we use three different intensities proposed by Adams, Murray, and MacKay (2009) as follows. $\lambda_1(x) = 2e^{-(x/15)} + e^{-((x-25)/10)^2}, \mathcal{X} = [0,50]$; $\lambda_2(x) = 5\sin(x^2) + 6, \mathcal{X} = [0,5]$, and $\lambda_3(x)$ is the piecewise linear shown in Figure 5.4 over the domain $\mathcal{X} = [0,100]$. Performance comparisons were based on (i) mean squared error (MSE) calculated as the sum of squared differences between vectors constructed by evaluating the true intensity and the appropriate predictive function for each model evaluated at an evenly-

spaced set of inputs, (ii) computational time-to-train in seconds and (iii) and empirical coverage (EC) (Leininger and Gelfand, 2015) which is a likelihood-free method for comparing point process models. To calculate EC, we first generate a data sample for each of the intensities by thinning. For each Bayesian method we generate 100 sample intensities, and for each of these we generate a point process sample. For the RKHS method we use only the learnt mean function.We then generate 5000 random subsets of the domain, and count the number of points ascribed to each of the sets by the data and the samples. This difference between counts is called the predictive residual (Leininger and Gelfand, 2015). We square these numbers and take their mean to get the EC.

Figure 5.2 depicts estimates of $\lambda_1(x)$ with different estimation methods. The true intensity is constructed by generating a point process realisation on the domain $[0,50]$ via thinning (J. F. C. Kingman, 1975). In Figures 5.3 and 5.4, we present $1-d$ examples using 16 and 8 basis functions respectively. It is clear that our method captures higher frequency behaviour than the other methods. The presented function in these cases is the estimate of the latent sample f. The corresponding intensity, from the superposition Gaussian Cox process, is implicit as its positive component.

Table 5.2 presents results for each intensity measure which demonstrate that OSGCP clearly outperforms other methods in terms of computational efficiency, whereas it performs at least as well in terms of MSE and EC.

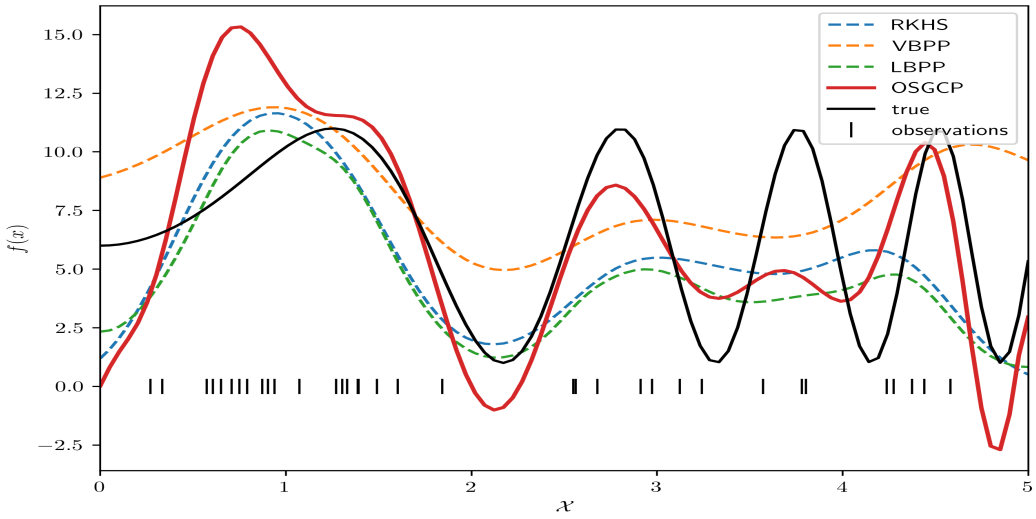

Figure 5.3: The presented curve is the posterior mean estimate in each case for the synthetic function λ_3 as described in the text. Ours (OSGCP) is constructed using $\eta = 0.12$, $\alpha = 1.5$, $\beta = 2.0$, $\mu = 0$, and $m = 8$.

Table 5.2: Metrics for synthetic data.

	$\lambda_1(x)$			$\lambda_2(x)$			$\lambda_3(x)$		
	MSE	EC	TIME(S)	MSE	EC	TIME(S)	MSE	EC	TIME(S)
OSGCP	0.099	**8.845**	**0.002**	**9.610**	**4.9454**	**0.004**	0.167	95.96	**0.002**
VBPP	0.165	13.509	3.81	11.006	6.612	3.86	0.354	59.44	3.82
LBPP	**0.083**	10.558	0.36	10.873	17.160	0.05	**0.151**	106.416	0.18
RKHS	0.129	16.1604	27.32	10.149	12.905	17.31	0.206	**29.167**	35.37

5.6.2 Real World Datasets

We present application of our method to the Redwood and the White Oak datasets (Baddeley, Rubak, and Turner, 2016). Figure 5.5 shows the learnt posterior mean function using OSGCP on the Redwood Dataset. RKHS required expensive cross-validation, which was aided only by the relatively low number of data points. The LBPP method required a slow optimisation step with relatively complex gradient construction and we found non-trivial to get the optimisation to converge properly with lots of tuning. In Figure 5.6, we present comparison of our method applied to the Redwood dataset. We use only a few

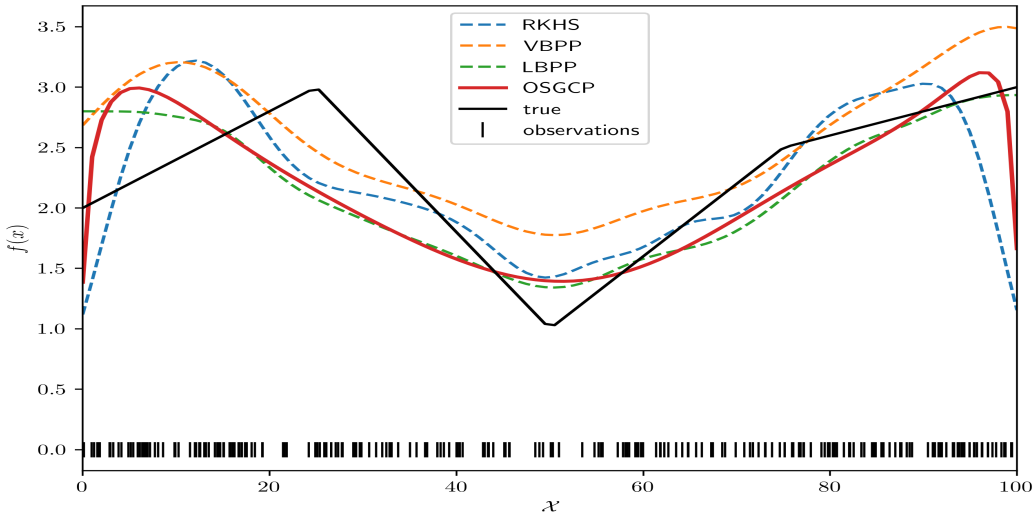

Figure 5.4: The presented curve is the posterior mean estimate in each case for the synthetic function λ_2 as described in the text. Ours (OSGCP) is constructed using $\eta = 0.12$, $\alpha = 1.5$, $\beta = 2.0$, $\mu = 0$i, and $m = 16$. See text for details on other methods.

Table 5.3: Metrics for real world datasets

	Redwood		White Oak	
	EC	time(s)	EC	time(s)
OSGCP	**5.85**	**0.006**	**12.8094**	**0.006**
VBPP	6.079	6.34	16.301	8.75
LBPP	418.225	1.24	1758.123	1.98
RKHS	10.103	961.4	186.602	1043.5

basis functions $m = 8^2$ to show that our method is able to capture finer structure at similar smoothness to the other methods. We would tend to expect a higher order in order to better capture high-frequency behaviour. The same phenomenon can be observed in Figure 5.7. Whilst these diagrams provide a subjective view of method performance, we feel that our method is better able to capture nuanced behaviour in the point process, and at a fraction of the computational cost.

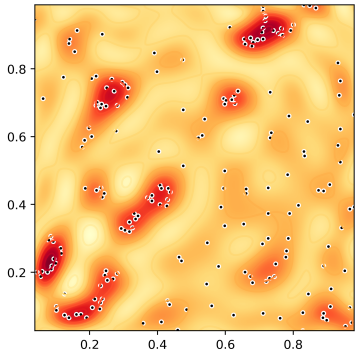

Figure 5.5: The learnt posterior mean estimate on the Redwood dataset (Adams, Murray, and MacKay, 2009). Parameters for generating this figure were $\mu = 0.0, \alpha = 1.5, \beta = 2.0, \eta = 0.12$, and $m = 20^2$.

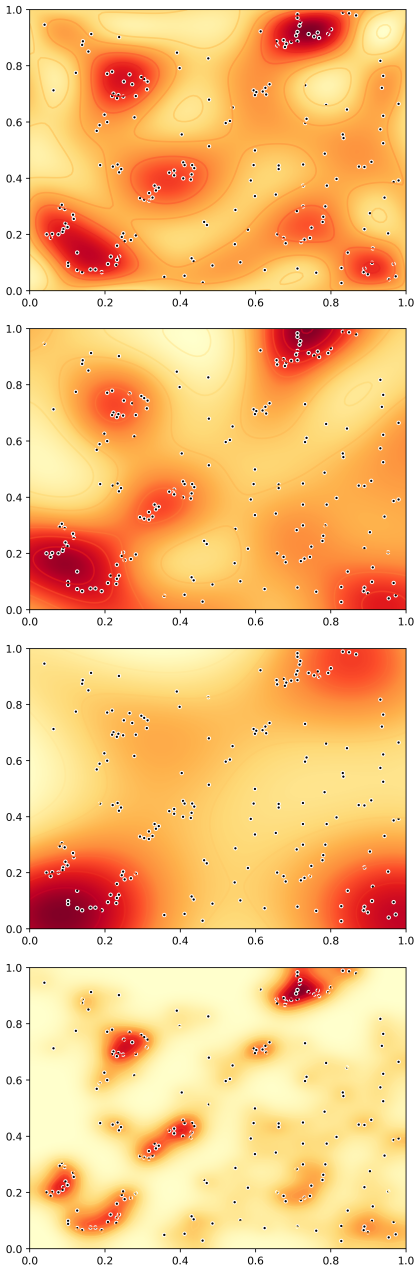

Figure 5.6: Redwood dataset method comparison. In each case we present the posterior mean or predictive mean function as prescribed in each paper. From top to bottom: Our method; VBPP Method (Lloyd et al., 2015); RKHS Method (Flaxman, Teh, and Sejdinovic, 2017); and LBPP Method (Walder and Bishop, 2017).

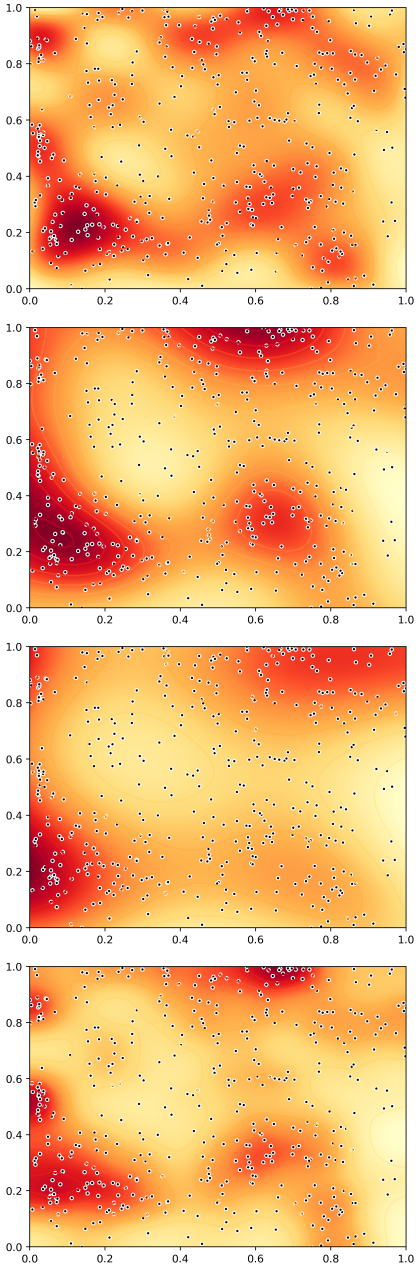

Figure 5.7: White Oak dataset method comparison. In each case we present the posterior mean or predictive mean function as prescribed in each paper. From top to bottom: Our method; VBPP Method (Lloyd et al., 2015); RKHS Method (Flaxman, Teh, and Sejdinovic, 2017); and LBPP Method (Walder and Bishop, 2017). Note that we are able to capture finer structure, at order $m = 10^2$. The other methods do not reflect high-frequency behaviour. We are able to retrieve more higher frequency behaviour by increasing the order.

5.7 Classification

We now present how the above model can be used to construct a stochastic classification model as considered by (McCullagh and Jie Yang, 2006a). The key idea is that in classification problems, one can think of the feature vector realisations as points on an appropriately defined space, such that the classification problem essentially becomes an intensity estimation problem. This interpretation is made simple by the concept of the marked Poisson process (J. Kingman, 2005), and the colouring theorem.

This interpretation of point processes as a classification model has been developed in a small but concise literature (McCullagh and Jie Yang, 2006a; J. Yang, Miescke, and Mccullagh, 2012; Matthews and Ghahramani, 2014). In this section we present an extension of that approach that takes advantage of the simplicity of our superposition Gaussian process model to avoid some of the computational issues present in those approaches.

First, we present the classification framework based on marked Point processes as presented by McCullagh and Jie Yang (2006a). We then present our extension to this via our construction that allows for the calculation of the relevant conditional probabilities.

Definition 28 (Classification)**.** *A classification problem is a tuple*

$$\langle \mathcal{U}, \mathcal{X}, \mathcal{Y} \rangle$$

where $\mathcal{U}$ *is a space of* units u *which are examples or events that can be described as belonging to different classes; a feature function* $x : \mathcal{U} \to \mathcal{X}$*, which maps units to* features*, and a labelling function* $y : \mathcal{X} \to \mathcal{Y}$*, which maps features to* labels*. The set of labels* $\mathcal{Y}$ *is a finite set* $\{y_1, y_2, \ldots, y_m\}$*.*

The aim of the practitioner is to find a rule for deciding, given feature values $x(u)$ of some new unit u, which class y it belongs to.

The approach to this problem, as outlined by McCullagh and Jie Yang (2006a), is to consider the observed data $\mathbf{x} \in \mathcal{S} \subset \mathcal{X}$ as a realisation of a set of point processes in $\mathcal{X}$. Specifically, we associate with each class y a Gaussian Cox process $\mathcal{P}_y$ on $\mathcal{S}$. This can be interpreted as a marked point process; i.e. a point process where at each point $x \in \mathcal{X}$ there is a mark $y \in \mathcal{Y}$.

We can thus, given such a dataset and the assumption regarding the point process data generating process, calculate conditional distributions of classes given the data. We first introduce the moment measure of a point process. This is also referred to as the n-point product density.

Definition 29. *Suppose $\mathcal{P}$ a Cox process on $\mathcal{S}$ with intensity function ψ. The moment measure of $\mathcal{P}$ is the measure μ on $\mathcal{S}^n$, evaluated at a point $\mathbf{x} = \{x_1, x_2, ..., x_n\}$, is written:*

$$\mu(\mathbf{x}) = \mathbb{E}\left[\prod_{i=1}^{n} \psi(x_i)\right]$$

This function, evaluated at $\mathbf{x}$, is the expected number of events in a volume $d^n x$ around $\mathbf{x}$.

Following McCullagh and Jie Yang (2006a), (J. Yang, Miescke, and Mccullagh, 2012), the probability, of a new unit u^* being in class y given the observed data is written:

$$p(y(u^*) = y|\mathbf{x}) = \frac{\mu_y(\mathbf{x} \cup \{x(u^*)\})}{\mu_y(\mathbf{x})} \tag{5.9}$$

where μ_y denotes the moment measure for the point process associated

with the class y.

The above expression is relatively simple in the case of the log-Gaussian Cox process (Matthews and Ghahramani, 2014); and less so for the permanental process (J. Yang, Miescke, and Mccullagh, 2012). In our case, however, we treat the sample function f at the input points directly at the intensity, since at the points where the data is found, the intensity is identical to the sample function. In order to calculate the appropriate class probabilities, we need to calculate the numerator and denominator of the expression (5.9). To do this, we need to calculate the moment measure of the process at the union of the data points and the test points, as well as at the data points. This latter expression is given by

$$\mu(\mathbf{x}) = \mathbb{E}\left[\psi(x_1)\psi(x_2)...\psi(x_n)\right]. \tag{5.10}$$

This is the expectation of the product of the intensity function at the observed points. For a zero-mean Gaussian process, this expression is given by Isserlis' theorem (Isserlis, 1918), known to physicists as the Wick formula; we state it here.

Theorem 18 (Isserlis' Theorem (Isserlis, 1918))**.** *Suppose $Z \sim \mathcal{N}(0,\Sigma)$ a multivariate, n-dimensional Gaussian random variable. Then,*

$$\mathbb{E}_{\mathcal{N}}[Z_1 Z_2 ... Z_n] = \sum_{\sigma \in P_n^2} \prod_{B \in \sigma} \Sigma_{i,\sigma i}$$

where P_n^2 is the set of derangements σ such that $\sigma^2 = Id$, and Z_i is the i-th component of the random vector Z.

Note that the expression in the theorem is the hafnian of the covariance matrix Σ (see Chapter 4 for more on the hafnian).

In our case, however, the intensity is not a zero-mean Gaussian process; we are interested in using the posterior GP to calculate the moment measure, and the posterior mean is informed by our orthogonal series estimate; i.e. it is not zero. As a result, we must adapt the above to our case. There exists an extension to the above theorem, due to Withers (1985). First however we define the loop hafnian (Björklund, Gupt, and Nicolás Quesada, 2019; Nicolas Quesada, 2019).

Definition 30 (Loop hafnian of a matrix)**.** *The loop hafnian of a $k \times k$ matrix A can be defined as*

$$lhaf(A) = \sum_{\sigma \in \pi} \prod_{i=1}^{k} A_{i,\sigma(i)}$$

where π is the set of involutions on k elements. The involutions are the permutations σ such that σ^2 is the identity.

Note the difference with the hafnian in that the loop hafnian includes permutations that can be written as a product of both 1-cycles and 2-cycles; the corresponding set in the definition of the hafnian includes only permutations that can be written as 2-cycles (i.e., derangements whose square is the identity).

We now present a theorem which will allow us to calculate the moment measure of the posterior GP intensity.

Theorem 19. *Suppose $Z \sim \mathcal{N}(\mu, \Sigma)$ a multivariate, n-dimensional Gaussian random variable. Construct the matrix $\hat{\Sigma}$ with entries:*

$$\hat{\Sigma}_{i,j} = \begin{cases} \mu_i & i = j \\ \Sigma_{i,j} & i \neq j \end{cases}$$

That is, the covariance matrix Σ with its diagonal having been replaced by the mean vector μ. Then,

$$\mathbb{E}_{\mathcal{N}}[Z_1 Z_2 ... Z_n] = lhaf(\hat{\Sigma}).$$

Proof. Proof in Appendix B. □

We can thus calculate the moment measure of the posterior GP intensity, by constructing $\hat{\Sigma}$ as in Theorem 19, with the mean vector calculated from the posterior mean, and the covariance matrix calculated from the posterior covariance.

As in the case of the hafnian presented in Chapter 4, computation of the loop hafnian is intractable for large matrices. The state-of-the-art algorithm for calculating the loop hafnian is due to Björklund, Gupt, and Nicolás Quesada (2019). Quoting from that paper: "Despite our highly optimized algorithm, numerical benchmarks on the Titan supercomputer with matrices up to size 56 × 56 indicate that one would require the 288000 CPUs of this machine for about a month and a half to compute the hafnian of a 100 × 100 matrix."

In order to save the practitioner the expense of such a computation for the purposes of our classification algorithm, we propose a Barvinok-type estimator of the loop hafnian in order to calculate the moment measure. We construct the estimator and show its unbiasedness in the following theorem.

Theorem 20. *Denote by A a $k \times k$ matrix, and by Z a random, skew-symmetric matrix such that its upper triangular entries are independent Gaussian random variables with mean 0 and variance 1. Define the*

estimator:

$$\alpha(A) = det((Z + I_k) \odot A_2)$$

where A_2 *is the matrix with entries defined*

$$\begin{cases} \sqrt{A_{ij}} & i \neq j \\ A_{ii} & i = j, \end{cases}$$

and where I_k *denotes the* $k \times k$ *identity matrix.*

Then, $\mathbb{E}[\alpha(A)] = lhaf(A)$.

Proof. Proof in Appendix B. □

Given the above, we can construct an estimator to the moment measure by generating matrices as in Theorem 19. Specifically, given the sample $\mathbf{x} = \{x_i\}_{i=0}^N$ and corresponding the intensity function estimate $\hat{\psi}$, we construct the matrix $\hat{K}$ as follows:

$$\hat{K}_{ij}[\mathbf{x}] = \begin{cases} \hat{\psi}(x_i) & \text{if } i = j \\ \sqrt{k\left(x_i, x_j\right)} & i \neq j \end{cases}$$

Then, applying Theorems 19 and 20, we can calculate the moment measure of the posterior GP intensity as

$$\hat{\mu}(\mathbf{x}) = \alpha(\hat{K}). \tag{5.11}$$

In practice, we generate for example $10,000$ iterations of $\alpha(\hat{K})$, and construct an estimate with improved variance by taking the average of these determinants; this can be achieved rapidly on modern GPUs. Furthermore, we can apply the control variate approach outlined in

Chapter 4 to reduce the variance of the estimator, since the control variate on the random component of Z is the same.

The resulting estimator of the class probability (5.9) at a new point u^* is then given by:

$$\hat{p}(y(u^*) = y|\mathbf{x}) = \frac{\hat{\mu}_y(\mathbf{x} \cup \{x(u^*)\})}{\hat{\mu}_y(\mathbf{x})} \tag{5.12}$$

5.8 Experiments: Classification

In this section we present some simple experiments to demonstrate the operation of the proposed method for classification problems. In Figure 5.8 we show a 1-dimensional example that allows us to visualize the effect of changing η on the resulting classification probabilities.

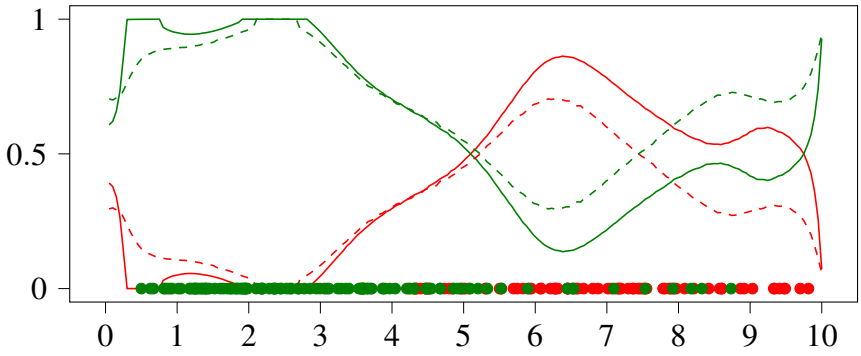

Figure 5.8: One-dimensional example of a classification problem. Curves shown are probability estimates as constructed using the Barvinok estimator outlined in this chapter. Intensities are given by scaled Gamma distributions with different parameters; class 1 has intensity $100 \times Gamma(8,1)$; class 2 has intensity $100 \times Gamma(3,1)$. Solid line: Prior hyperparameters are $\mu_i = 0.0$, $\alpha = 1.5$, $\beta = 0.0$, η=0.01. Dashed line: Prior hyperparameters are $\mu_i = 0.0$, $\alpha = 1.5$, $\beta = 0.0$, η=0.02. Note the effect of increasing the weighting parameter η leads to less certain estimates.

We also present a 2-dimensional example in Figure 5.9 which highlights the natural ability of the method to capture non-linearities without specific feature construction nor especial kernel design. The

idea for the checkerboard also found in the original papers by McCullagh and Jie Yang (2006b). For methods applied to *density* estimation, see (Ghalebikesabi et al., 2023), where it is pointed out that popular neural network based approaches, such as masked autoregressive flows (Papamakarios, Murray, and Pavlakou, 2017) and rational quadratic neural spline flows (Durkan et al., 2019) can struggle in such small-data density estimation problems.

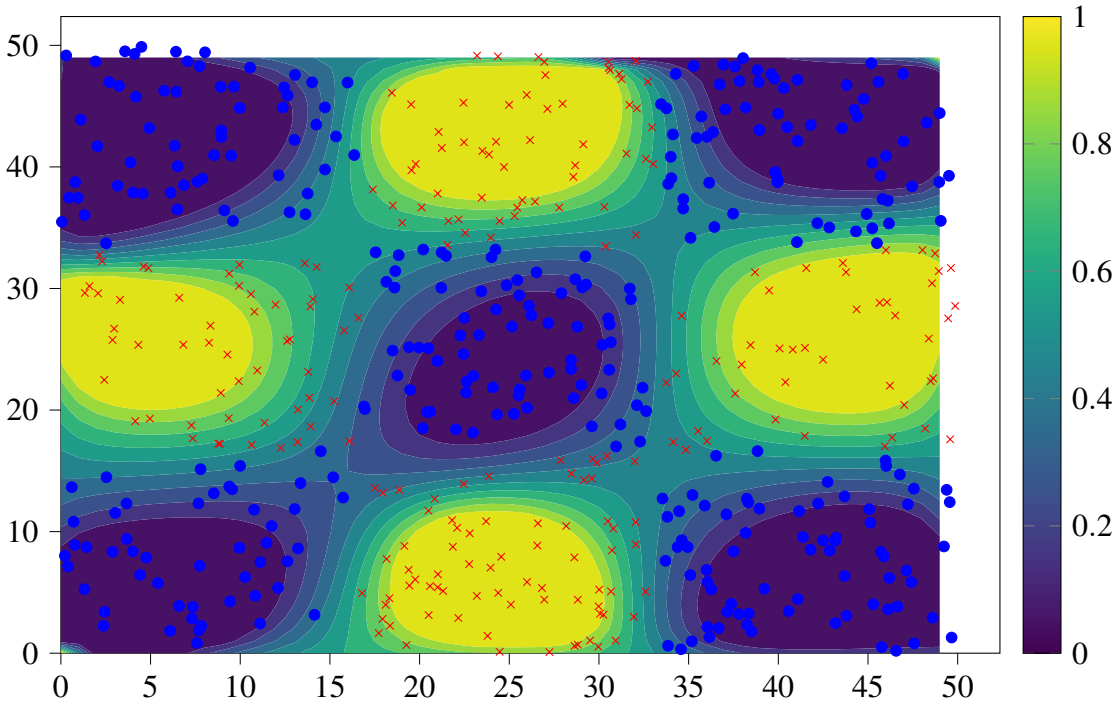

Figure 5.9: Application of the proposed point process classification method to a two-dimensional example. The presented function, presented as a contour plot, is the normalised probability ascribed to the class here whose observations are denoted with red crosses.

5.9 Conclusion

In this chapter we have presented a novel method for modelling point process data. This was achieved by definition of a new type of Gaussian Cox process, which allowed us to utilise the properties of the in-

duced inhomogeneous Poisson process in combination with the properties of orthogonal series estimators to directly construct a Bayesian prior on the weight-space view of the latent Gaussian process, which is not feasible in the standard permanental or log-Gaussian Cox process models.

This yields a method for estimation that allowed for direct expression of prior weighting between mean-function and kernel eigenvalues, which map directly to the prior weighting for standard Gaussian random variables in Bayesian analysis. This yields a method that requires no training stage, and deals directly with the heretofore underexplored problem of identification of the latent Gaussian process in the Cox process model. For example if the aim is to learn how the Cox process *varies* in space, then prior weighting can lean towards emphasizing the role of the eigenvalues; if the aim is to learn how the data behave in the specific region of observation, then prior weighting can lean towards emphasizing the role of the mean function. It is not possible to identify these two separate aspects with a single Cox process sample, a fact that has not apparently been explicitly dealt with in the relevant literature.

We also extended this to classification problems, following the stochastic classification approach of McCullagh and Jie Yang (2006b). Normally this would be intractable, but our Barvinok-type estimator approach, connecting the loop hafnian to the higher moments of Gaussian random variables, allowed us to construct a tractable stochastic classifier to sit alongside the permanental process classifier and the log Gaussian Cox process classifier, which yield actual probabilistic predictions rather than merely generating output values on the simplex. As a result we have a classification method that is

able to flexibly capture strongly non-linear decision boundaries with no tuning required.

Chapter 6

Conclusion

We have presented three applications of orthogonality to machine learning problems. First, we highlighted the importance of orthonormality in the choice of basis for constructing sparse Gaussian process models. Gaussian process models use the properties of the Gaussian distribution to represent information about infinite dimensional operators using finite dimensional matrices. Many applications of Gaussian processes rely on approximations to the operator, but ignore that extent to which this operator remains well-approximated by a finite-dimensional counterpart. We showed that, in one case of such sparse approximation, if the basis functions used to represent the behaviour of the operator are not orthonormal, the finite-dimensional approximation will be poor. We then presented a way to construct asymptotically orthonormal basis functions for the Gaussian process, and showed that this yields a sparse approximation that is asymptotically exact. This yields a novel approach to feature construction and sparse Gaussian process regression.

Future work on this topic should focus on application of the multivariate orthogonal polynomial literature to the construction of these basis functions. The standard approach to multivariate

problems involves the construction of the tensor product of univariate basis functions. However, as Theorem 7 this is only really valid for cases where the input variables are independent. Future work should extend the present work to utilise the literature on multivariate orthogonal polynomials (Xu, 1994a; Xu, 1994b).

Secondly, we showed an application of orthogonality to graph embeddings. Associated with every graph is a sequence of polynomials, and we showed that there exists a measure on the input space of these polynomials such that the given sequence is orthogonal. By comparing these measures, we can compare the graphs in a given sequence, and we showed how this could be applied this to an anomaly detection problem.

Future work in this area could improve the computation of matching polynomials for weighted graphs by acquiring e.g. a complement theorem for weighted graphs. This would allow for improved computation of the matching polynomial for denser weighted graphs. Further work should also look into improved estimation for the matching polynomial coefficients for larger graphs. The noise in the Barvinok estimator approach we outlined in Section 4.6.1.2 is quite high, and it would be beneficial to look at more stable estimators that either use better control variates, or look into equivalent formulations that yield the same information without the limitations on the graph size due to computational complexity. Finally, long-term work should look into applying the theory of perturbation of orthogonal polynomials to the graph matching polynomial, in order to better formalise the approach, whose effectiveness we have only demonstrated empirically.

Finally in Chapter 5, we presented a use of orthonormal bases to construct rapid estimators for Gaussian Cox process models. This

sidestepped the usual computational complexity of these models, and provided a method to construct a classification model that produces probabilistic predictions. This contrasts it with standard approaches to classification, which typically merely construct classification boundaries.

The approach we have demonstrated is valid for Poisson process data; specifically, data that does not exhibit interaction between points. This means it is not valid for a general range of spatial point process data. Future work should look into extending the approach to more general point process data that includes point interactions in the form of repulsion or clustering. This maybe achieved by noting that the determinantal point process and permanental point process (McCullagh and Møller, 2006; Kulesza and Taskar, 2012; Hough et al., 2009), which yield repulsion and clustering respectively, may benefit either from kernel sparsification methods, or combined modelling of the determinantal and permanental point processes. using the Barvinok estimators utilised in this thesis.

Appendix A

Lemmas

Lemma 5 (Favard's Theorem (Favard, 1935)). *Let $\{\beta_n\}_{n=0}$ be an arbitrary real sequence, and $\{\gamma_n\}_{n=0}$ be a sequence of positive real numbers. Let $\{P_n(x)\}_{n=0}^{\infty}$ be a polynomial sequence such that $P_{-1}(x) = 0$; $P_0(x) = 1$, and following the recurrence:*

$$P_n(x) = (x - \beta_n)P_{n-1}(x) - \gamma_n P_{n-2}(x).$$

Then $\{P_n(x)\}_{n=0}^{\infty}$ is an orthogonal polynomial sequence (OPS); and there is a unique moment functional $\mathcal{L}$ s.t. $\langle \mathcal{L}, 1\rangle = \gamma_1$ and $\langle \mathcal{L}, P_j(x)P_k(x)\rangle = B_i\delta_{jk}$ for some constant B_i depending on the order of the polynomial.

We can get an equivalent form written in the language of operators on sequences as follows. Rearranging the three-term recurrence above, there exist a_n, b_n such that

$$xP_n(x) = a_n P_n(x) + b_n P_{n+1}(x) + b_{n-1}P_{n-1}(x).$$

We can rewrite this as

$$AP = xP \tag{A.1}$$

where A is the infinite-dimensional tridiagonal matrix:

$$\begin{bmatrix} a_0 & b_0 & 0 & 0 & 0 & 0 \\ b_0 & a_1 & b_1 & 0 & 0 & 0 \\ 0 & b_1 & a_2 & b_2 & 0 & 0 \\ 0 & 0 & b_2 & a_3 & b_3 & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

and P is the infinite-dimensional column vector:

$$\begin{bmatrix} P_0(x) \\ P_1(x) \\ P_2(x) \\ P_3(x) \\ \vdots \end{bmatrix}$$

The content of Favard's theorem is then that any polynomial sequence that fulfills (A.1) *is an orthogonal polynomial sequence with respect to some measure ν.*

Lemma 6 (Ostrowski's Theorem (Braun, 2006))**.** *Denote by $\lambda_i(A)$ the operator that returns the i-th descending ordered eigenvalue of a square matrix A. Let Λ be a symmetric $m \times m$ matrix and Φ a non-singular $n \times m$ matrix. For $1 \leq i \leq m$, there exists some $\delta_i \geq 0$ such that:*

$$\lambda_m(\Phi'\Phi) \leq \delta_i \leq \lambda_1(\Phi'\Phi)$$

and:

$$\lambda_i(\Phi\Lambda\Phi') = \delta_i\lambda_i(\Lambda)$$

Lemma 7 (von-Neumann trace inequality (Carlsson, 2021))**.** *For any $N \times N$ matrices A, B, with decreasingly ordered singular values $a_1, a_2, ..., a_N$, $b_1, b_2, ..., b_N$ respectively,*

$$|tr(AB)| \leq \sum_{i=1}^{N} a_i b_i$$

with equality if and only if A, B share the same singular vectors.

Appendix B

Proofs

B.1 Proofs: Chapter 3

Proof of Theorem 7. For the forward direction, assume that $\{\phi_i\}$ are orthonormal. Note that a result of Ostrowski's theorem (Lemma 6) is that

$$\lambda_1\left(\frac{1}{N}\Phi'\Phi\right)\lambda_i(\Lambda) \geq \lambda_i\left(\frac{1}{N}\Phi\Lambda\Phi'\right) \geq \lambda_m\left(\frac{1}{N}\Phi'\Phi\right)\lambda_i(\Lambda)$$
$$\Rightarrow \left(\lambda_1\left(\frac{1}{N}\Phi'\Phi\right)-1\right)\lambda_i(\Lambda) \geq \lambda_i\left(\frac{1}{N}\Phi\Lambda\Phi'\right)-\lambda_i \geq \left(\lambda_m\left(\frac{1}{N}\Phi'\Phi\right)-1\right)\lambda_i(\Lambda)$$

Since $\{\phi_i\}$ are orthonormal w.r.t to ν, $\frac{1}{N}[\Phi'\Phi]_{ij} \to \int_{\mathcal{X}} \phi_i(x)\phi_j(x)d\nu = \delta_{ij}$ by the law of large numbers. This means the eigenvalues $\lambda_i(\frac{1}{N}\Phi'\Phi)$ converge to 1, and the eigenvalues $\lambda(\frac{1}{N}\Phi\Lambda\Phi')$ converge to λ_i by the squeeze theorem.

For the reverse direction, it is sufficient to show that at least one eigenvalue fails to converge if the functions $\{\phi_i\}_{i=0}^m$ are not orthonormal. First, we show that orthogonality alone does not achieve convergence. Then, we extend to the case when the basis functions are not orthogonal. Suppose that the $\{\phi_i\}_{i=0}^m$ are orthogonal but not

orthonormal. To simplify notation, define $\hat{\Phi}'\hat{\Phi} = \lim_{N\to\infty} \frac{1}{N}\Phi'\Phi$. Orthogonality of the basis functions implies that $\hat{\Phi}'\hat{\Phi}$ is diagonal. The result is trivial if $\lambda_1(\hat{\Phi}'\hat{\Phi}) < 1$ or $\lambda_m(\hat{\Phi}'\hat{\Phi}) > 1$, by the application of the last inequality above. To consider the case when $\lambda_1(\hat{\Phi}'\hat{\Phi}) > 1 > \lambda_m(\hat{\Phi}'\hat{\Phi})$, by the properties of the determinant (via the Sylvester determinant identity), the non-zero eigenvalues of the product AB of two matrices A, B, are equal to those of BA; so that $\lim_{N\to\infty} \lambda_i(\frac{1}{N}\Phi\Lambda\Phi') = \lambda_i(\Lambda^{\frac{1}{2}}\hat{\Phi}'\hat{\Phi}\Lambda^{\frac{1}{2}})$. The matrix $\Lambda^{\frac{1}{2}}\hat{\Phi}'\hat{\Phi}\Lambda^{\frac{1}{2}}$ is diagonal, because it is the matrix product of diagonal matrices. This means that we have $\lambda_i(\frac{1}{N}\Phi\Lambda\Phi') = \lambda_i(\Lambda)$ only if $(\hat{\Phi}'\hat{\Phi}) = I.$, i.e. the basis functions are orthonormal.

Now, suppose that the $\{\phi_i\}_{i=0}^m$ are not orthogonal. Showing that the trace of $\frac{1}{N}\Phi\Lambda\Phi'$ fails to converge to $\mathrm{tr}(\Lambda)$ is sufficient to show that at least one eigenvalue fails to converge, *a fortiori*. Note that since the basis functions are not orthogonal, $\hat{\Phi}'\hat{\Phi}$ is not diagonal. By the properties of the trace, $\mathrm{tr}\left(\frac{1}{N}\Phi\Lambda\Phi'\right) = \mathrm{tr}\left(\frac{1}{N}\Lambda\Phi'\Phi\right)$. It is thus sufficient to show that $\mathrm{tr}\left(\frac{1}{N}\Lambda\Phi'\Phi\right) \to \mathrm{tr}(\Lambda)$. We proceed by contradiction, and use a recent extension to the implications of the von-Neumann trace inequality (Carlsson, 2021). Suppose $\mathrm{tr}\left(\Lambda\hat{\Phi}'\hat{\Phi}\right) = \mathrm{tr}(\Lambda)$. Then by Lemma 7, we have:

$$\mathrm{tr}(\Lambda) = \mathrm{tr}\left(\Lambda\hat{\Phi}'\hat{\Phi}\right) \tag{B.1}$$

$$\leq |\,\mathrm{tr}\left(\Lambda\hat{\Phi}'\hat{\Phi}\right)\,| \leq \sum_{i=0}^{m} \lambda_i\lambda_i(\hat{\Phi}'\hat{\Phi}). \tag{B.2}$$

where we have (B.1) by assumption, and (B.2) by the von-Neumann trace inequality. We also have

$$\mathrm{tr}(\Lambda I_m) = \sum_{i=0}^{m} \lambda_i. \tag{B.3}$$

By Lemma 7, we have equality in (B.2) if and only if Λ, I_m share the same eigenvectors. Since Λ and I_m are both diagonal, (B.2) and (B.3) imply that I_m and $\hat{\Phi}'\hat{\Phi}$ share the same eigenvectors, by the assumption that $\mathrm{tr}\left(\Lambda\hat{\Phi}'\hat{\Phi}\right) = \mathrm{tr}\left(\Lambda\right)$. By assumption the basis functions are not orthogonal, so the matrix $\hat{\Phi}'\hat{\Phi}$ is not diagonal, and thus its eigenvectors do not point along the axes.

We therefore have a contradiction. As a result, the eigenvalues $\lambda_i(\frac{1}{N}\Phi'\Lambda\Phi) \to \lambda_i$ if and only if the basis functions are orthonormal. □

Proof of Theorem 8. For a measure ν with moments $\{\mu_i\}_{i=0}^{\infty}$, we refer to the sum $\sum_{i=0}^{\infty}\mu_{2i}^{-2i} = \infty$ as the Carleman sum for measure ν. Since, by assumption, the moments $\{\mu_i\}_{i=0}^{\infty}$ fulfill Carleman's condition, the measure ν is the solution to a determinate moment problem (Chihara, 2011). By the Radon-Nikodym theorem, we denote by $wd\nu$ the measure ζ such that $\zeta(A) = \int_A w(x)d\nu$. where $w = \frac{d\zeta}{d\nu}$. By the fact that $w \leq 1$, the Carleman sum for $wd\nu$ upper bounds that of ν term-by-term. As a result, $wd\nu$ is also the solution to a determinate moment problem.

By Theorem 6.1 composed with Theorem 3.3 of (Chihara, 2011), the corresponding moment functional is therefore positive definite, so there exists an orthogonal polynomial sequence $\{P_i\}_{i=0}^{\infty}$ orthogonal with respect to the measure $wd\nu$. By the determinacy of $wd\nu$, and by Deift (2000, Corollary 2.50), the polynomials are dense in $\mathcal{L}^2[wd\nu]$. Suppose now that the support of ν is compact. For a function $f : \mathcal{X} \to \mathbb{R}$, such that f can be written $f = gw^{1/2}$ for $g \in \mathcal{L}^2[wd\nu]$ and $\varepsilon > 0$, by the Stone-Weierstrass theorem there exists a polynomial g^*, of order m^*, such that $\sup_{x\in\mathrm{supp}(\nu)}|g(x) - g_{m^*}(x)| < \varepsilon$. Then there is way to write it as $g^* = \sum_{i=0}^{m^*} g_i d_i P_i$ for some coefficients g_i and normalising constants c_i, which means there is way to write $f_{m^*} = \sum_{i=0}^{m^*} g_i c_i P_i w^{1/2} = \sum_{i=0}^{m^*} f_i \phi_i$. Hence,

the functions $\{\phi_i\}_{i=0}^{\infty}$ form a basis in $\mathcal{L}^2[wdv]$. □

Proof of Theorem 9. Convergence of $\hat{P}_i(x)$ to $P_i(x)$ is equivalent to convergence of the corresponding recurrence coefficients. Since the sample $\mathcal{D}$ is iid, $\hat{\mu}_j \to_p \mu_j$ by the law of large numbers. $\hat{H}_n$ is a continuous polynomial function of the sample moments $\hat{\mu}_j, \{j = 1,2,...,2n\}$. By the continuous mapping theorem (Hayashi, 2000, Lemma 2.3), $\hat{H}_n \to_p H_n$. Since this polynomial is not identically zero, its set of zeroes has Lebesgue measure zero (Mityagin, 2020). As a result, the functions for the coefficients (3.8), (3.9). are defined with probability 1, since the denominators are zero with probability 0. As a result, $\hat{\beta}_n \to_p \beta_n, \hat{\gamma}_n \to_p \gamma_n$, so $\hat{P}_i(x) \to P_i(x)$. □

Proof of Theorem 10. Since $\hat{f}$ is in the projection $\mathcal{H}_k^{m^*}$, it can be written $\hat{f} = \sum_{i=0}^{m^*} \hat{\theta}_i \phi_i$. We can write a consistent estimator for the noise parameter:

$$\begin{aligned}\hat{\sigma}^2 =& \frac{1}{N}\sum_{i=1}^{N}\left(\left(\mathbf{y}_i - \sum_{j=0}^{\hat{m}} \hat{\theta}_j \phi_j(\mathbf{x}_i)\right)^2\right) \\ =& \frac{1}{N}\sum_{i=1}^{N}\left(\left(\sum_{j=0}^{m^*} \theta_j \phi_j(\mathbf{x}_i) + \varepsilon_i - \sum_{j=0}^{\hat{m}} \theta_j \phi_j(\mathbf{x}_i)\right)^2\right)\end{aligned}$$

which has the same limit in probability as any other consistent estimator for the noise parameter. If $\hat{m} < m^*$, we can write this as:

$$= \frac{1}{N}\sum_{i=1}^{N}\left(\left(\sum_{j=0}^{\hat{m}} (\theta_j - \hat{\theta}_j)\phi_j(\mathbf{x}_i) + \varepsilon_i + \sum_{j=\hat{m}+1}^{m^*} \theta_j \phi_j(\mathbf{x}_i))\right)^2\right) \tag{B.4}$$

or, if $\hat{m} \geq m^*$, we can write this as:

$$= \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{j=0}^{m^*} (\theta_j - \hat{\theta}_j)\phi_j(\mathbf{x}_i) + \varepsilon_i - \sum_{j=m^*+1}^{\hat{m}} \hat{\theta}_j \phi_j(\mathbf{x}_i)) \right)^2 \tag{B.5}$$

where by $\hat{\theta}$ we mean the vector of posterior mean coefficients. We begin with the case when $\hat{m} < m^*$. Expanding the square and taking expectations with respect to ε_i we get:

$$\begin{aligned}
\mathbb{E}_\varepsilon \left[\hat{\sigma}^2\right] = & \frac{1}{N} \sum_{i=1}^{N} \left(\left(\sum_{j=0}^{\hat{m}} (\theta_j - \hat{\theta}_j)\phi_j(\mathbf{x}_i) \right)^2 \right. \\
& + \sigma^2 \\
& + \sum_{j=\hat{m}+1}^{m^*} \left(\theta_j \phi_j(\mathbf{x}_i)\right)^2 \\
& \left. - 2 \sum_{j=0}^{m^*} (\theta_j - \hat{\theta}_j)\phi_j \sum_{j=\hat{m}+1} \theta_j \phi_j(\mathbf{x}_i) \right) \\
= & \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{k=0}^{\hat{m}} \sum_{j=0}^{\hat{m}} (\theta_j - \hat{\theta}_j)(\theta_k - \hat{\theta}_k)\phi_j(\mathbf{x}_i)\phi_k(\mathbf{x}_i) \right) \qquad \text{(B.6)} \\
& + \sigma^2 \\
& + \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{j=\hat{m}+1}^{m^*} \theta_j \phi_j(\mathbf{x}_i) \right)^2 \\
& - 2 \frac{1}{N} \sum_{i=1}^{N} \sum_{j=0}^{\hat{m}} (\theta_j - \hat{\theta}_j)\phi_j(\mathbf{x}_i) \sum_{j=\hat{m}+1}^{m^*} \theta_j \phi_j(\mathbf{x}_i) \qquad \text{(B.7)}
\end{aligned}$$

By orthonormality of ϕ_j, and by consistency of Bayesian estimators, for any δ there exists a sample size N_1^* such that the term (B.6) is less than $\delta/2$ for all $N \geq N_1^*$. Similarly, there exists a sample size N_2^* such that (B.7) is less than $\delta/2$ for all $N \geq N_2^*$, since none of the terms in the product of the sums shares a ϕ_j.

Define $N^* = \max\{N_1^*, N_2^*\}$. Then, for all $N \geq N^*$, we can write

$$\begin{aligned}\mathbb{E}_\varepsilon\left[\hat{\sigma}^2\right] =& \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{j=0}^{\hat{m}}(\theta_j - \hat{\theta}_j)\phi_j(\mathbf{x}_i)^2 + \sigma^2 + \sum_{j=\hat{m}+1}^{m^*}\theta_j^2\phi_j(\mathbf{x}_i)^2\right.\\ &\left. - 2\sum_{j=0}^{m^*}(\theta_j - \hat{\theta}_j)\phi_j(\mathbf{x}_i)\sum_{j=\hat{m}+1}\theta_j\phi_j(\mathbf{x}_i)\right)\\ &\leq \sigma^2 + \delta + \sum_{j=\hat{m}+1}^{m^*}\theta_j\phi_j(\mathbf{x}_i)^2 \,\forall\, N \geq N^*.\end{aligned}$$

Following a similar approach for when $\hat{m} \geq m^*$, we can write

$$\mathbb{E}_\varepsilon\left[\hat{\sigma}^2\right] \leq \sigma^2 + \delta + \frac{1}{N}\sum_{i=1}^{N}\sum_{j=0}^{m^*}\theta_j\phi_j(\mathbf{x}_i)^2 \,\forall\, N \geq N^*.$$

Define $\eta_{\hat{m}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=\hat{m}+1}^{m^*}\theta_j\phi_j(\mathbf{x}_i)^2$, when $\hat{m} < m^*$, and $\eta_{m^*} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=m^*+1}^{\hat{m}}\theta_j\phi_j(\mathbf{x}_i)^2$, when $\hat{m} \geq m^*$. Define further:

$$\eta_m = \begin{cases}\eta_{m^*} & \hat{m} \geq m^*\\ \eta_{\hat{m}} & \hat{m} < m^*\end{cases} \tag{B.8}$$

Substituting this into (B.4), we have

$$\mathbb{E}_\varepsilon\left[\hat{\sigma}^2\right] \leq \left(\sigma^2 + \delta + \eta_m\right) \,\forall\, N \geq N^*.$$

□

Proof of Theorem 12. Denoting the imaginary unit by j, The covari-

ance between two values of the complex gp **h** is given by:

$$
\begin{aligned}
&\mathbb{E}\left[\mathbf{h}(x)\mathbf{h}(x')\right] \\
=&\mathbb{E}_{\omega}\left[\mathbb{E}_{b}\left[\mathbb{E}_{\theta}\left[\mathbf{h}(x)\mathbf{h}(x') \mid \omega, b\right] \omega\right]\right] \\
=&\mathbb{E}_{\omega}\left[\mathbb{E}_{b}\left[\mathbb{E}_{\theta}\left[(\mathbf{f}(x)+j\mathbf{g}(x))(\mathbf{f}(x')+j\mathbf{g}(x')) \mid \omega, b\right] \mid \omega\right]\right] \\
=&\mathbb{E}_{\omega}\left[\mathbb{E}_{b}\left[\mathbb{E}_{\theta}\left[\mathbf{f}(x)\mathbf{f}(x') \mid \omega, b\right] \mid \omega\right]\right] \\
+&j\mathbb{E}_{\omega}\left[\mathbb{E}_{b}\left[\mathbb{E}_{\theta}\left[\mathbf{g}(x)\mathbf{f}(x') \mid \omega, b\right] \mid \omega\right]\right] \\
+&j\mathbb{E}_{\omega}\left[\mathbb{E}_{b}\left[\mathbb{E}_{\theta}\left[\mathbf{g}(x')\mathbf{f}(x) \mid \omega, b\right] \mid \omega\right]\right] \\
-&\mathbb{E}_{\omega}\left[\mathbb{E}_{b}\left[\mathbb{E}_{\theta}\left[\mathbf{g}(x)\mathbf{g}(x')) \mid \omega, b\right] \mid \omega\right]\right] \\
=&\mathbb{E}_{\omega}\left[\mathbb{E}_{b}\left[\mathbb{E}_{\theta}\left[\mathbf{f}(x)\mathbf{f}(x') \mid \omega, b\right] \mid \omega\right]\right] \\
-&\mathbb{E}_{\omega}\left[\mathbb{E}_{b}\left[\mathbb{E}_{\theta}\left[\mathbf{g}(x)\mathbf{g}(x')) \mid \omega, b\right] \mid \omega\right]\right]
\end{aligned}
$$

where the complex terms are dropped in the last equation by the law of iterated expectations and the fact that the coefficients θ_i, θ_i' are uncorrelated.

Taking each of these terms separately,

$$
\begin{aligned}
&\mathbb{E}_\omega\left[\mathbb{E}_b\left[\mathbb{E}_\theta\left[\mathbf{f}(x)\mathbf{f}(x') \mid \omega, b\right] \mid \omega\right]\right] \\
=&\mathbb{E}_\omega\left[\mathbb{E}_b\left[\mathbb{E}_\theta\left[\sum_{i=0}^{R}\theta_i\phi_i^{NS}(x)\sum_{i=0}^{R}\theta_i\phi_i^{NS}(x') \mid \omega, b\right] \mid \omega\right]\right] \\
=&\mathbb{E}_\omega\left[\mathbb{E}_b\left[\sum_{i=0}^{R}\phi_i^{NS}(x)\phi_i^{NS}(x') \mid \omega\right]\right] \\
=&\mathbb{E}_\omega\left[\mathbb{E}_b\left[\sum_{i=0}^{R}\frac{1}{R}\cos(\omega_{i1}x+b_j)\cos(\omega_{i1}x'+b_j) \mid \omega\right]\right] \\
+&\mathbb{E}_\omega\left[\mathbb{E}_b\left[\sum_{i=0}^{R}\frac{1}{R}\cos(\omega_{i1}x+b_j)\cos(\omega_{i2}x'+b_j) \mid \omega\right]\right] \\
+&\mathbb{E}_\omega\left[\mathbb{E}_b\left[\sum_{i=0}^{R}\frac{1}{R}\cos(\omega_{i2}x+b_j)\cos(\omega_{i1}x'+b_j) \mid \omega\right]\right] \\
+&\mathbb{E}_\omega\left[\mathbb{E}_b\left[\sum_{i=0}^{R}\frac{1}{R}\cos(\omega_{i2}x+b_j)\cos(\omega_{i2}x'+b_j) \mid \omega\right]\right]
\end{aligned}
$$

By the properties of the cosine function,

$$
\begin{aligned}
&\mathbb{E}_\omega\left[\mathbb{E}_b\left[\cos(\omega_{ij}x+b_i)\cos(\omega_{ik}x'+b_i) \mid \omega\right]\right] \\
=&\mathbb{E}_\omega\left[\mathbb{E}_b\left[\frac{1}{2}\left(\cos(\omega_{ij}x+\omega_{ik}x'+2b_i)\right] \mid \omega\right]\right. \\
+&\mathbb{E}_\omega\left[\frac{1}{2}\left(\cos(\omega_{ij}x-\omega_{ik}x')\right)\right] \\
=&\mathbb{E}_\omega\left[\frac{1}{2}\left(\cos(\omega_{ij}x-\omega_{ik}x')\right)\right]
\end{aligned}
$$

for $j,k \in \{1,2\}$, $i \in \{1,...,R\}$, where the last equality holds because $b_i \sim \text{Unif}[0,2\pi]$ and, by the law of iterated expectations, the first term vanishes. Substituting this into the corresponding terms above, we

get:

$$\begin{aligned}
&\mathbb{E}_\omega\left[\mathbb{E}_b\left[\mathbb{E}_\theta\left[\mathbf{f}(x)\mathbf{f}(x') \mid \omega, b\right]\right]\right] \\
=&\mathbb{E}_\omega\left[\sum_{i=0}^{R}\frac{1}{2R}\cos(\omega_{i1}(x-x'))\right] && \text{(B.9)} \\
+&\mathbb{E}_\omega\left[\sum_{i=0}^{R}\frac{1}{2R}\cos(\omega_{i1}x-\omega_{i2}x')\right] && \text{(B.10)} \\
+&\mathbb{E}_\omega\left[\sum_{i=0}^{R}\frac{1}{2R}\cos(\omega_{i2}x-\omega_{i1}x')\right] && \text{(B.11)} \\
+&\mathbb{E}_\omega\left[\sum_{i=0}^{R}\frac{1}{2R}\cos(\omega_{i2}(x-x'))\right] && \text{(B.12)}
\end{aligned}$$

By Bochner's theorem (Rahimi and Recht, 2007), the terms (B.9) and (B.12) are stationary kernels, whose spectral densities are the *marginal* densities of ω_1, ω_2 respectively. By Yaglom's theorem (Theorem 11), the terms (B.10) and (B.11) are non-stationary kernels following the spectral density $F_{\Omega_1,\Omega_2}(\omega_1,\omega_2)$. As a result, we get a final expression:

$$\begin{aligned}
&\mathbb{E}_\omega\left[\mathbb{E}_b\left[\mathbb{E}_\theta\left[\mathbf{f}(x)\mathbf{f}(x') \mid \omega, b\right] \mid \omega\right]\right] \\
=&\frac{1}{2}k_{\omega_1}\left(x,x'\right)+\frac{1}{2}k\left(x,x'\right)+\frac{1}{2}k\left(x',x\right)+\frac{1}{2}k_{\omega_2}\left(x,x'\right) && \text{(B.13)}
\end{aligned}$$

where by $k_{\omega_j}(x,x')$ we denote a stationary kernel whose spectral density is the marginal density of the random variable ω_j. Using similar reasoning for $\mathbf{g}$, we get:

$$\begin{aligned}
&\mathbb{E}_\omega\left[\mathbb{E}_b\left[\mathbb{E}_\theta\left[\mathbf{g}(x)\mathbf{g}(x') \mid \omega, b\right] \mid \omega\right]\right] \\
=&\frac{1}{2}k_{\omega_1}\left(x,x'\right)+\frac{1}{2}k_{\omega_2}\left(x,x'\right) && \text{(B.14)}
\end{aligned}$$

Subtracting (B.14) from (B.13) gives:

$$\begin{aligned}
&\mathbb{E}_\omega\left[\mathbb{E}_b\left[\mathbb{E}_\theta\left[\mathbf{f}(x)\mathbf{f}(x') \mid \omega, b\right] \mid \omega\right]\right] \\
-&\mathbb{E}_\omega\left[\mathbb{E}_b\left[\mathbb{E}_\theta\left[\mathbf{g}(x)\mathbf{g}(x') \mid \omega, b\right] \mid \omega\right]\right] \\
=&\frac{1}{2}k_{\omega_1}\left(x,x'\right)+\frac{1}{2}k\left(x,x'\right)+\frac{1}{2}k\left(x',x\right)+\frac{1}{2}k_{\omega_2}\left(x,x'\right) \\
-&\left(\frac{1}{2}k_{\omega_1}\left(x,x'\right)+\frac{1}{2}k_{\omega_2}\left(x,x'\right)\right) \\
=&\frac{1}{2}k\left(x,x'\right)+\frac{1}{2}k\left(x',x\right) \\
=&k\left(x,x'\right)
\end{aligned}$$

where the last equality holds because since the kernel is symmetric. □

B.2 Proofs: Chapter 4

Proof of Theorem 13. If the summands in the summation term in (4.2) are independent of i, then the vertex removal recurrence can be written:

$$Q(\mathcal{G};x) = xQ(\mathcal{G}-v;x) - w_{v'}Q(\mathcal{G}-v-v';x)$$

for some weight w_v and arbitrary vertex $v' \in \mathcal{G}-v$. Now, apply Favard's theorem (Lemma 5). □

Proof of Theorem 14. We begin with the existence of the measure ν and its corresponding orthogonal polynomials. Using the notation in the theorem statement, the matching polynomial recurrence can be

written:

$$Q(\mathcal{G};x) = xQ(\mathcal{G}_{\sigma_1};x) - \sum_{v\in\mathcal{G}_{\sigma_1}} Q(\mathcal{G}_{\sigma_1} - v;x)$$

$$\Rightarrow Q(\mathcal{G};x) + \sum_{v\in\mathcal{G}_{\sigma_1}} Q(\mathcal{G}_{\sigma_1} - v;x) = xQ(\mathcal{G}_{\sigma_1};x)$$

The summation term is a polynomial of order $n-2$ since it is the sum of $n-1$ polynomials of order $n-2$. Since the polynomials are a basis for the space of polynomials of order $n-2$, we can write this system:

$$\tilde{H}\tilde{Q} = x\tilde{Q}$$

where $\tilde{H}$ is a lower Hessenberg matrix, and $\tilde{Q}$ is the vector of matching polynomials, corresponding to the graphs in the complete node sequence σ. In fact, we can extend this to an operator on l^2, the space of square-summable sequences, by appending an arbitrary sequence of polynomials to $\tilde{Q}$ and corresponding coefficients to $\tilde{H}$ to get:

$$HQ = xQ$$

for Q a sequence of polynomials such that the i-th polynomial is the matching polynomial $Q(\mathcal{G}_{\sigma_i};x)$ if $i < n$, the matching polynomial $Q(\mathcal{G};x)$ if i == n, and any arbitrary polynomial otherwise.

On this sequence, multiplication by the independent variable is thus equivalent to application of the operator H. The spectral theorem for multiplication operators Reed and Simon (1972)[see Theorem VII.3] states that multiplication operators are unitarily equivalent to self-

adjoint operators, so we can write:

$$HQ = UAU^{-1}Q = xQ \tag{B.15}$$

where U denotes a unitary operator, and A is a self-adjoint operator.

Note that for the space of polynomials, the operator of multiplication by the independent variable x trivially has a cyclic vector; that is, a vector ψ such that the span of $\psi, x\psi, x^2\psi...$ is complete for the space of polynomials. Therefore, by Stone (1932)[Theorem 7.13], the self adjoint operator A has a tridiagonal representation, and we can write:

$$AU^{-1}Q = xU^{-1}Q \tag{B.16}$$

$$\Rightarrow AP = xP \tag{B.17}$$

As a result, we can write $P = U^{-1}Q$. By Favard's theorem 5 is a sequence of orthogonal polynomials for some measure ν. We have therefore shown existence of the measure and corresponding orthogonal polynomials.

Given this, suppose the measure ν_σ is available. we can thus write the Christoffel-Darboux kernel $k_m^\nu(x,y)$ for the orthonormal polynomials $\{P_i\}_{i=0}^{m+1}$ as:

$$k_m^\nu(x,y) = \frac{d}{c}\frac{P_{m+1}(x)P_m(y) - P_m(x)P_{m+1}(y)}{x-y} \tag{B.18}$$

where c, d are the normalising constants of the polynomials P_m and P_{m+1} respectively. Apply the Gram-Schmidt process to the polynomials

$Q(\mathcal{G};x), Q(\mathcal{G}_\sigma;x)...$ etc, so that:

$$P_{m+1}(x) = cQ(\mathcal{G};x)$$
$$P_m(x) = d\left[Q(\mathcal{G}_{\sigma_1};x) - acQ(\mathcal{G};x)\right]$$

Substituting into (B.18) we obtain:

$$\begin{aligned}
k_m^\nu(x,y) &= \frac{d}{c}\frac{P_{m+1}(x)P_m(y) - P_m(x)P_{m+1}(y)}{x-y} \\
&= \frac{d}{c}\frac{cQ(\mathcal{G};x)d\left[Q(\mathcal{G}_\sigma;y) - acQ(\mathcal{G};y)\right]}{x-y} \\
&\quad - \frac{d\left[Q(\mathcal{G}_\sigma;x) - acQ(\mathcal{G};x)\right]cQ(\mathcal{G};y)}{x-y} \\
&= \frac{d}{c}\frac{cQ(\mathcal{G};x)dQ(\mathcal{G}_\sigma;y) - cQ(\mathcal{G};y)dQ(\mathcal{G}_\sigma;x)}{x-y} \\
&= d^2 k_\mathcal{G}^\sigma(x,y)
\end{aligned}$$

where a is the orthogonalising factor between $Q(\mathcal{G};x)$ and $Q(\mathcal{G}_\sigma;x)$, and normalise ν_σ such that $d = 1$. Therefore,

$$k_m^\nu(x,y) = k_\mathcal{G}^\sigma(x,y).$$

□

Proof of Theorem 16. The determinant $\det(\alpha(A))$ can be expanded as follows:

$$\begin{aligned}
\det(\alpha(A)) &= \sum_{\sigma\in S_n} \operatorname{sgn}(\sigma)\prod_{i=1}^n \alpha(A)_{i,\sigma(i)} \\
&= \sum_{\sigma\in S_n} \operatorname{sgn}(\sigma)\prod_{i=1}^n (-1)^{\mathbb{I}[i>\sigma(i)]} Z_{i,\sigma(i)} A_{2i,\sigma(i)}
\end{aligned}$$

where S_n is the set of permutations of $\{1,\ldots,n\}$ and the indicator function $\mathbb{I}[i > \sigma(i)]$ terms capture whether the value is above or below

the diagonal. Opening the terms in the sum, we get:

$$\det(\alpha(A)) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \left((-1)^{\mathbb{I}[1>\sigma(1)]} Z_{1\sigma(1)} \ldots (-1)^{\mathbb{I}[2k>\sigma(2k)]} Z_{2k\sigma(2k)} \right) \cdot \prod_{i=1}^{n} A_{2(i,\sigma(i))}$$

Firstly, note that the terms Z_{ij} have an extra negative sign if $j < i$, by the skew-symmetric property of Z; the number of extra signs is precisely the number of crossings of the permutation σ, so the $\operatorname{sgn}(\sigma)$ is cancelled. The determinant becomes:

$$\det(\alpha(A)) = \sum_{\sigma \in S_n} \left(Z_{1\sigma(1)} Z_{2\sigma(2)} \ldots Z_{2k\sigma(2k)} \right) \prod_{i=1}^{n} A_{2(i,\sigma(i))}$$

Since the diagonal of Z is 0, the elements Z_{ii} zero out the summand. The only permutations that remain are the derangements (permutations without fixed points). Taking the expectation, any permutation such that $Z_{i\sigma(i)}$ is not also matched with $Z_{\sigma(i)i}$ will have an expectation of 0, since the elements of Z are independent otherwise. This must be true of all the elements in the product $\left(Z_{1\sigma(1)} Z_{2\sigma(2)} \ldots Z_{2k\sigma(2k)} \right)$, since otherwise the expectation of the whole term is 0. The permutations corresponding to non-zero terms therefore contain both $A_{i\sigma(i)}$ and $A_{\sigma(i)i}$. Since the matrix A_2 is symmetric, $A_{2(i,\sigma(i))} = A_{2(\sigma(i),i)}$, and since the elements used are square roots, the resulting captured values are $A_{2(i,\sigma(i))} A_{2(\sigma(i),i)} = A_{i,\sigma(i)}$.

The above argument is equivalent to saying that any non-zero term derangement maps $i \to \sigma(i)$, and $\sigma(i) \to i$; or, stated otherwise, $\sigma^2 = \mathrm{Id}$. Hence, the set of kept permutations is precisely the set τ. $\square$

B.3 Proofs: Chapter 5

Proof of Theorem 17.

$$\begin{aligned}
\mathbb{E}\left[\hat{\xi}\right] &= \mathbb{E}\left[\sum_{i=0}^{N}\phi_j(x_i)\right] \\
&= \int_{\mathcal{X}}\phi_j(x)d\Psi(x) \qquad \text{(B.19)} \\
&= \int_{\mathcal{X}}\phi_j(x)\psi(x)d\nu(x) \\
&= \int_{\mathcal{X}}\phi_j(x)\sum_{k=0}^{\infty}\xi_k\phi_k(x)d\nu(x) \\
&= \sum_{k=0}^{\infty}\xi_k\int_{\mathcal{X}}\phi_j(x)\phi_k(x)d\nu(x) \\
&= \sum_{k=0}^{\infty}\xi_k\delta_{kj} \\
&= \xi_j.
\end{aligned}$$

where we have B.19 by Lemma 4. □

Proof of Theorem 19. By Theorem 1.1 in Withers (1985), the mean $\mathbb{E}_{\mathcal{N}}[Z_1Z_2...Z_n]$ can be written $\sum_{k=0}^{\lfloor n/2\rfloor}\sum_{\sigma\in\mathcal{I}_n}\mu_{\sigma(1)}\mu_{\sigma(2)}\cdots\mu_{\sigma(n-2k)}\Sigma_{\sigma(k)\sigma(k+1)}\Sigma_{\sigma(2k-1)\sigma(2k)}$. where $\mathcal{I}_n$ is the set of involutions on the set n, which is the set of permutations σ such that σ^2 is the identity. Equivalently, they can be written only as the composition of 1-cycles and 2-cycles. The term in the summand of the loop hafnian of a matrix A is written:

$$\prod_{i=1}^{n}A_{i,\sigma(i)}. \qquad \text{(B.20)}$$

Since the permutation σ in the loop hafnian definition can be written as a composition of 1-cycles and 2-cycles, the product (B.20) has terms either $A_{i,i}$ or $A_{i,\sigma(i)}$. Placing the mean vector μ on the diagonal of $A_{i,i}$ thus yields precisely the formula described by Withers (1985). □

Proof of Theorem 20. The determinant $\det(\alpha(A))$ can be expanded as follows:

$$\begin{aligned}\det(\alpha(A)) &= \sum_{\sigma\in S_n} \operatorname{sgn}(\sigma)\prod_{i=1}^{n}\alpha_{i,\sigma(i)} \\ &= \sum_{\sigma\in S_n} \operatorname{sgn}(\sigma)\prod_{i=1}^{n}(-1)^{\mathbb{I}[i>\sigma(i)]}Z_{i,\sigma(i)}A_{2i,\sigma(i)}\end{aligned}$$

where S_n is the set of permutations of $\{1,\ldots,n\}$ and the indicator function $\mathbb{I}[i>\sigma(i)]$ terms capture whether the value is above or below the diagonal. Opening the terms in the sum, we get:

$$\det(\alpha(A)) = \sum_{\sigma\in S_n} \operatorname{sgn}(\sigma)\left((-1)^{\mathbb{I}[1>\sigma(1)]}Z_{1\sigma(1)}\ldots(-1)^{\mathbb{I}[2k>\sigma(2k)]}Z_{2k\sigma(2k)}\right)\cdot\prod_{i=1}^{n}A_2 i,\sigma(i)$$

Firstly, note that the terms Z_{ij} have an extra negative sign if $j<i$, by the skew-symmetric property of Z; the number of extra signs is precisely the number of crossings of the permutation σ, so the $\operatorname{sgn}(\sigma)$ is cancelled. The determinant becomes:

$$\det(\alpha(A)) = \sum_{\sigma\in S_n}\left(Z_{1\sigma(1)}Z_{2\sigma(2)}\ldots Z_{2k\sigma(2k)}\right)\prod_{i=1}^{n}A_{2(i,\sigma(i))}$$

Since the diagonal of Z is 1, the elements Z_{ii} are captured in permutations where $i=\sigma(i)$. The permutations that remain are the derangements (permutations without fixed points). Taking the expectation, any permutation such that $Z_{i,\sigma(i)}$ is not also matched with $Z_{\sigma(i),i}$ will have an expectation of 0, since the elements of Z are independent otherwise. This must be true of all the elements in the

product $\left(Z_{1,\sigma(1)}Z_{2,\sigma(2)}...Z_{2k,\sigma(2k)}\right)$, since otherwise the expectation of the whole term is 0. The permutations corresponding to non-zero terms therefore contain both $A_{i,\sigma(i)}$ and $A_{\sigma(i),i}$, or terms $A_{i,i}$ as noted above. Since the matrix A_2 is symmetric, $A_{2(i,\sigma(i))} = A_{2(\sigma(i),i)}$, and since the elements used are square roots, the resulting captured values are $A_{2(i,\sigma(i))}A_{2(\sigma(i),i)} = A_{i,\sigma(i)}$

The above argument is equivalent to saying that any non-zero term derangement maps $i \to \sigma(i)$, and $\sigma(i) \to i$. This includes values where $i \to i$. Stated otherwise, $\sigma^2 = \mathrm{Id}$. Hence, the set of kept permutations is precisely the set of involutions π. □

Bibliography

Adams, Ryan Prescott, Iain Murray, and David J. C. MacKay (2009). "Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities". In: *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*. Ed. by Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman. Vol. 382. ACM International Conference Proceeding Series. ACM, pp. 9–16.

Aglietti, Virginia et al. (2019). "Structured Variational Inference in Continuous Cox Process Models". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach et al., pp. 12437–12447.

Akoglu, Leman, Hanghang Tong, and Danai Koutra (2014). "Graph-based Anomaly Detection and Description: A Survey". In: arXiv: 1404.4679.

— (2015). "Graph based anomaly detection and description: A survey". In: *Data Mining and Knowledge Discovery* 29.3, pp. 626–688. arXiv: 1404.4679.

Baddeley, Adrian, Ege Rubak, and Rolf Turner (2016). *Spatial point patterns : methodology and applications with R / Adrian Baddeley,*

Ege Rubak, Rolf Turner. Chapman & Hall/CRC interdisciplinary statistics series. Boca Raton: CRC Press.

Barrow, H G and R M Burstall (1976). "Subgraph isomorphism, matching relational structures and maximal cliques". In: *Information Processing Letters* 4.4, pp. 83–84.

Barvinok, Alexander (1999). "Polynomial Time Algorithms to Approximate Permanents and Mixed Discriminants Within a Simply Exponential Factor". In: *Random Structures & Algorithms* 14.1, pp. 29–61.

Bernardo, José M and Adrian F M Smith (2009). *Bayesian Theory*. 1. Aufl. Wiley Series in Probability and Statistics. Newark: Wiley.

Björklund, Andreas, Brajesh Gupt, and Nicolás Quesada (2019). "A Faster Hafnian Formula for Complex Matrices and Its Benchmarking on a Supercomputer". In: *ACM J. Exp. Algorithmics* 24.

Bochner, Salomon (1929). "Über Sturm-Liouvillesche Polynomsysteme." In: *Mathematische Zeitschrift* 29, pp. 730–736.

Braun, Mikio L. (2006). "Accurate error bounds for the eigenvalues of the kernel matrix". In: *Journal of Machine Learning Research* 7.82, pp. 2303–2328.

Brochu, Eric, Vlad M. Cora, and Nando de Freitas (2010). "A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning". In: *CoRR* abs/1012.2. arXiv: 1012.2599.

Bunke, H (1997). "On a relation between graph edit distance and maximum common subgraph". In: *Pattern Recognition Letters* 18.8, pp. 689–694.

Calandriello, Daniele et al. (2019). "Gaussian Process Optimization with Adaptive Sketching: Scalable and No Regret". In: *Conference*

on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA. Ed. by Alina Beygelzimer and Daniel Hsu. Vol. 99. Proceedings of Machine Learning Research. PMLR, pp. 533–557.

Carlsson, Marcus (2021). "von Neumann's trace inequality for Hilbert–Schmidt operators". In: *Expositiones Mathematicae* 39.1, pp. 149–157.

Chihara, Theodore (2011). *An Introduction to Orthogonal Polynomials*. Dover Books on Mathematics. Dover Publications.

Choiruddin, Achmad et al. (2020). "Regularized estimation for highly multivariate log Gaussian Cox processes". In: *Statistics and Computing* 30.3, pp. 649–662. arXiv: 1905.01455.

Clenshaw, C W (1955). "A note on the summation of Chebyshev series". In: *Mathematics of Computation* 9.51, pp. 118–120.

Cox, D. R. (1955). "Some Statistical Methods Connected with Series of Events". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 17.2, pp. 129–157.

Cunningham, Harry Jake et al. (2023). "Actually Sparse Variational Gaussian Processes". In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. Ed. by Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent. Vol. 206. Proceedings of Machine Learning Research. PMLR, pp. 10395–10408.

Cvetkovic, Dragos M. et al. (1988). *Recent Results in the Theory of Graph Spectra*. 1st ed. Vol. 36. Annals of Discrete Mathematics. San Diego: Elsevier Science.

Daskalakis, Constantinos, Petros Dellaportas, and Aristeidis Panos (2022). "How Good Are Low-Rank Approximations in Gaussian Process Regression?" In: *Thirty-Sixth AAAI Conference on Artificial*

Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022. AAAI Press, pp. 6463–6470.

Deift, Percy (2000). "Orthogonal polynomials and random matrix theory: a Riemann-Hilbert perspective". In: Volume 3 of Courant lecture notes in Mathematics: Courant Institute of Mathematical Sciences.

Deift, Percy et al. (1999). "Uniform asymptotics for polynomials orthogonal with respect to varying exponential weights and applications to universality questions in random matrix theory". In: *Communications on Pure and Applied Mathematics* 52.11, pp. 1335–1425.

Deisenroth, Marc Peter and Carl Edward Rasmussen (2011). "PILCO: A Model-Based and Data-Efficient Approach to Policy Search". In: *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*. Ed. by Lise Getoor and Tobias Scheffer. Omnipress, pp. 465–472.

Dellaportas, Petros and Ioannis Kontoyiannis (2012). "Control variates for estimation based on reversible Markov chain Monte Carlo samplers". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74.1, pp. 133–161.

Diggle, Peter J. et al. (2013). "Spatial and spatio-temporal log-gaussian cox processes: Extending the geostatistical paradigm". In: *Statistical Science* 28.4, pp. 542–563.

Ding, Xiucai and Thomas Trogdon (2021). "A Riemann–Hilbert approach to the perturbation theory for orthogonal polynomials:

Applications to numerical linear algebra and random matrix theory". In: *ArXiv preprint* abs/2112.12354.

Donner, Christian and Manfred Opper (2018). "Efficient Bayesian inference of sigmoidal Gaussian cox processes". In: *Journal of Machine Learning Research* 19.67, pp. 1–34. arXiv: 1808.00831.

Durkan, Conor et al. (2019). "Cubic-Spline Flows". In: *ArXiv preprint* abs/1906.02145.

Fan, Ying, Letian Chen, and Yizhou Wang (2018). "Efficient Model-Free Reinforcement Learning Using Gaussian Process". In: *ArXiv preprint* abs/1812.04359.

Farrell, Edward. J. (1979). "An introduction to matching polynomials". In: *Journal of Combinatorial Theory, Series B* 27.1, pp. 75–86.

— (1980). "The matching polynomial and its relation to the acyclic polynomial of a graph." In: *Ars Combinatoria* 9, pp. 221–228.

Farrell, Edward. J. and Shanaz. A. Wahid (1986). "Matching polynomials: A matrix approach and its applications". In: *Journal of the Franklin Institute* 322.1, pp. 13–21.

Fasshauer, Gregory E. (2012a). "Green's functions: Taking another look at kernel approximation, radial basis functions, and splines". In: *Springer Proceedings in Mathematics*. Vol. 13, pp. 37–93.

— (2012b). "Green's functions: Taking another look at kernel approximation, radial basis functions, and splines". In: *Springer Proceedings in Mathematics*. Ed. by Marian Neamtu and Larry Schumaker. Vol. 13. New York, NY: Springer New York, pp. 37–93.

Favard, Jean (1935). "Sur les polynomes de Tchebicheff". In: *C.R. Acad. Sci. Paris* 200, pp. 2052–2053.

Flaxman, Seth R., Yee Whye Teh, and Dino Sejdinovic (2017). "Poisson intensity estimation with reproducing kernels". In: *Proceedings*

of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA. Ed. by Aarti Singh and Xiaojin (Jerry) Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, pp. 270–279.

Frank, Andrew J. and Arthur Asuncion (2010). *UCI Machine Learning Repository*.

Gao, Rui and Anton Kleywegt (2016). "Distributionally Robust Stochastic Optimization with Wasserstein Distance". In: *ArXiv preprint* abs/1604.02199.

Gaston, M E, M Kraetzl, and W D Wallis (2006). "Using graph diameter for change detection in dynamic networks". In: *Australasian Journal of Combinatorics* 35, pp. 299–311.

Gautschi, Walter (1982). "On Generating Orthogonal Polynomials". In: *SIAM Journal on Scientific and Statistical Computing* 3.3, pp. 289–317.

— (1986). "On the sensitivity of orthogonal polynomials to perturbations in the moments". In: *Numerische Mathematik* 48.4, pp. 369–382.

— (2004). *Orthogonal Polynomials*. eng. Oxford: Oxford University Press, p. 312.

Genton, Marc G (2001). "Classes of Kernels for Machine Learning: A Statistics Perspective". In: *Journal of Machine Learning Research* 2, pp. 299–312.

Ghalebikesabi, Sahra et al. (2023). "Quasi-Bayesian nonparametric density estimation via autoregressive predictive updates". In: *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*. Ed. by Robin J Evans and Ilya Shpitser. Vol. 216. Proceedings of Machine Learning Research. PMLR, pp. 658–668.

Girolami, Mark (2002). “Orthogonal series density estimation and the kernel eigenvalue problem”. In: *Neural Computation* 14.3, pp. 669–688.

Godsil, Christopher David (2017). *Algebraic Combinatorics*. Chapman and Hall mathematics. Place of publication not identified: Routledge.

Godsil, Christopher. D. (1981a). “Hermite polynomials and a duality relation for matchings polynomials”. In: *Combinatorica* 1.3, pp. 257–262.

— (1981b). “Matching behaviour is asymptotically normal”. In: *Combinatorica* 1.4, pp. 369–376.

— (1992). “Walk Generating Functions, Christoffel-Darboux Identities and the Adjacency Matrix of a Graph”. In: *Combinatorics, Probability and Computing* 1.1, pp. 13–25.

Godsil, Christopher. D. and Brendan. D. McKay (1982). “Constructing cospectral graphs”. In: *Aequationes Mathematicae* 25.1, pp. 257–268.

Goyal, Palash and Emilio Ferrara (2018). “Graph embedding techniques, applications, and performance: A survey”. In: *Knowledge-Based Systems* 151, pp. 78–94. arXiv: 1705.02801.

Gretton, Arthur et al. (2007). “A kernel approach to comparing distributions”. In: *Proceedings of the National Conference on Artificial Intelligence*. Vol. 2, pp. 1637–1641.

Hagberg, Aric A, Daniel A Schult, and Pieter J Swart (2008). “Exploring Network Structure, Dynamics, and Function using NetworkX”. In: *Proceedings of the 7th Python in Science Conference*. Ed. by Gaël Varoquaux, Travis Vaught, and Jarrod Millman. Pasadena, CA USA, pp. 11–15.

Hall, Brian C. (2013). *Quantum Theory for Mathematicians*. Graduate texts in mathematics ; 267. New York ; Springer, pp. 1–566.

Hayashi, Fumio. (2000). *Econometrics*. Princeton, N.J. ; Princeton University Press.

Heilmann, Ole J. and Elliott H. Lieb (1972). "Theory of monomer-dimer systems". In: *Communications in Mathematical Physics* 25.3, pp. 190–232.

Hensman, James, Nicolas Durrande, and Arno Solin (2018). "Variational Fourier features for Gaussian processes". In: *Journal of Machine Learning Research* 18.151, pp. 1–52. arXiv: 1611.06740.

Hoang, Quang Minh et al. (2020). "Revisiting the Sample Complexity of Sparse Spectrum Approximation of Gaussian Processes". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al.

Hough, J. et al. (2009). *Determinantal point processes*. Ed. by Gernot Akemann, Jinho Baik, and Philippe Di Francesco.

Isserlis, L. (1918). "On a Formula for the Product-Moment Coefficient of Any Order of a Normal Frequency Distribution in Any Number of Variables". In: *Biometrika* 12.1-2, pp. 134–139.

Jerrum, Mark (1987). "Two-dimensional monomer-dimer systems are computationally intractable". In: *Journal of Statistical Physics* 48.1-2, pp. 121–134.

John, S. T. and James Hensman (2018). "Large-Scale Cox Process Inference using Variational Fourier Features". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*.

Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 2367–2375.

Kanagawa, Motonobu et al. (2018). "Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences". In: *ArXiv preprint* abs/1807.02582.

Karhunen, K (1947). *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*. Annales Academiae Scientiarum Fennicae: Ser. A 1. Kirjapaino oy. sana.

Kimeldorf, George S. and Grace Wahba (1970). "A Correspondence Between Bayesian Estimation on Stochastic Processes and Smoothing by Splines". In: *The Annals of Mathematical Statistics* 41.2, pp. 495–502.

Kingman, J. F. C. (1975). *Random Discrete Distributions*. Tech. rep. 1, pp. 1–15.

— (2005). *P oisson Processes*. Clarendon Press.

Kolchinsky, Artemy and Brendan D. Tracey (2017). "Estimating mixture entropy with pairwise distances". In: *Entropy* 19.7. arXiv: 1706.02419.

Kronmal, R. and M. Tarter (1968). "The Estimation of Probability Densities and Cumulatives by Fourier Series Methods". In: *Journal of the American Statistical Association* 63.323, pp. 925–952.

Kulesza, Alex and Ben Taskar (2012). "Determinantal point processes for machine learning". In: *Foundations and Trends in Machine Learning* 5.2-3, pp. 123–286. arXiv: 1207.6083.

Lang, Serge (2002). *Algebra*. Rev. 3rd e. Graduate texts in mathematics ; 211. New York: Springer.

Lasserre, Jean Bernard, Edouard Pauwels, and Mihai Putinar (2022). *The Christoffel–Darboux Kernel for Data Analysis*. Cambridge: Cambridge University Press.

Lei, Jing and Larry Wasserman (2014). "Distribution-free Prediction Bands for Non-parametric Regression". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 76.1, pp. 71–96.

Leininger, Thomas and Alan Gelfand (2015). "Bayesian Inference and Model Assessment for Spatial Point Patterns Using Posterior Predictive Samples". In: *Bayesian Analysis* 12.

Lloyd, Chris M. et al. (2015). "Variational Inference for Gaussian Process Modulated Poisson Processes". In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. Ed. by Francis R. Bach and David M. Blei. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 1814–1822.

Loève, Michel. (1977). *Probability Theory*. 4th ed. Graduate texts in mathematics ; 45-46. New York: Springer-Verlag.

Maroñas, Juan et al. (2021). "Transforming Gaussian Processes With Normalizing Flows". In: *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*. Ed. by Arindam Banerjee and Kenji Fukumizu. Vol. 130. Proceedings of Machine Learning Research. PMLR, pp. 1081–1089.

Matern, B (1960). *Spatial Variation*. Lecture Notes in Statistics. Springer New York.

Matsakis, Nicholas D and Felix S Klock (2014). "The Rust Language". In: *Proceedings of the 2014 ACM SIGAda Annual Conference on*

High Integrity Language Technology. HILT '14. New York, NY, USA: Association for Computing Machinery, pp. 103–104.

Matthews, Alexander G. de. G and Zoubin Ghahramani (2014). "Classification using log Gaussian Cox processes". In: arXiv: 1405.4141.

Mayers, David F., Gene H. Golub, and Charles F. van Loan (1986). *Matrix Computations.* 3rd ed. Vol. 47. Johns Hopkins studies in the mathematical sciences 175. Baltimore ; Johns Hopkins University Press, p. 376.

McCullagh, Peter and Jesper Møller (2006). "The permanental process". In: *Advances in Applied Probability* 38.4, pp. 873–888.

McCullagh, Peter and Jie Yang (2006a). "Stochastic classification models". In: *International Congress of Mathematicians, ICM 2006* 3, pp. 669–686.

— (2006b). "Stochastic classification models". In: *International Congress of Mathematicians, ICM 2006*. Vol. 3. 669-686, pp. 669–686.

Mercer, James (1909). "XVI. Functions of positive and negative type, and their connection the theory of integral equations". In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 209.441-458, pp. 415–446.

Mityagin, Boris. S. (2020). "The Zero Set of a Real Analytic Function". In: *Mathematical Notes* 107.3-4, pp. 529–530. arXiv: 1512.07276.

Møller, Jesper, Anne Randi Syversveen, and Rasmus Plenge Waagepetersen (1998). "Log Gaussian Cox processes". In: *Scandinavian Journal of Statistics* 25.3, pp. 451–482.

Mutny, Mojmir and Andreas Krause (2018). "Efficient High Dimensional Bayesian Optimization with Additivity and Quadrature

Fourier Features". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. Ed. by Samy Bengio et al., pp. 9019–9030.

Newman, James E. and Moshe Y. Vardi (2020). "FPRAS Approximation of the Matrix Permanent in Practice". In: *ArXiv preprint* abs/2012.03367.

Novick, Melvin R and W J Hall (1965). "A Bayesian Indifference Procedure". In: *Journal of the American Statistical Association* 60.312, pp. 1104–1117.

Page, Lawrence et al. (1999). "The PageRank Citation Ranking : Bringing Order to the Web". In: *The Web Conference*.

Papamakarios, George, Iain Murray, and Theo Pavlakou (2017). "Masked Autoregressive Flow for Density Estimation". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al., pp. 2338–2347.

Puckette, S. E. and Walter Rudin (1965). *Fourier Analysis on Groups*. reprint. Vol. 72. 6. New York: Wiley Classics Library, Wiley-Interscience, p. 686.

Quesada, Nicolas (2019). "Franck-Condon factors by counting perfect matchings of graphs with loops". In: *The Journal of Chemical Physics* 150, p. 164113.

Rahimi, Ali and Benjamin Recht (2007). "Random Features for Large-Scale Kernel Machines". In: *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British*

Columbia, Canada, December 3-6, 2007. Ed. by John C. Platt et al. Curran Associates, Inc., pp. 1177–1184.

Rasmussen, Carl Edward and Joaquin Quiñonero Candela (2005). "Healing the relevance vector machine through augmentation". In: *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*. Ed. by Luc De Raedt and Stefan Wrobel. Vol. 119. ACM International Conference Proceeding Series. ACM, pp. 689–696.

Rasmussen, Carl Edward and Christopher K. I. Williams (2018). *Gaussian Processes for Machine Learning*.

Reade, J. B. (1984). "Eigenvalues of smooth kernels". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 95.1, pp. 135–140.

— (1992). "Eigenvalues of smooth positive definite kernels". In: *Proceedings of the Edinburgh Mathematical Society* 35.1, pp. 41–45.

Reed, Michael and Barry Simon (1972). *Methods of modern mathematical physics [by] Michael Reed [and] Barry Simon.* New York: Academic Press.

Remes, Sami, Markus Heinonen, and Samuel Kaski (2017). "Non-Stationary Spectral Kernels". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al., pp. 4642–4651.

Rudelson, Mark, Alex Samorodnitsky, and Ofer Zeitouni (2016). "Hafnians, perfect matchings and Gaussian matrices". In: *The Annals of Probability* 44.4, pp. 2858–2888.

Rudin, Walter (1976). *Principles of mathematical analysis / Walter Rudin.* 3rd ed. International series in pure and applied mathematics. Auckland ; McGraw-Hill.

Schölkopf, Bernhard, Ralf Herbrich, and Alexander J. Smola (2001). "A Generalized Representer Theorem". In: *Computational Learning Theory, 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001, Amsterdam, The Netherlands, July 16-19, 2001, Proceedings*. Ed. by David P. Helmbold and Robert C. Williamson. Vol. 2111. Lecture Notes in Computer Science. Springer, pp. 416–426.

Shi, Yongtang et al. (2016). "Graph polynomials". In: *Graph Polynomials*. Ed. by Y. Shi et al. 1st Editio. Taylor and Francis Ltd. Chap. 5, pp. 1–249.

Shoubridge, Peter et al. (2002). "Detection of abnormal change in a time series of graphs". In: *Journal of Interconnection Networks* 03.01n02, pp. 85–101.

Solin, Arno and Simo Särkkä (2020). "Hilbert space methods for reduced-rank Gaussian process regression". In: *Statistics and Computing* 30.2, pp. 419–446. arXiv: 1401.5508.

Stein, William and David Joyner (2005). "SAGE: System for Algebra and Geometry Experimentation". In: *SIGSAM Bull.* 39.2, pp. 61–64.

Steinwart, Ingo, Don Hush, and Clint Scovel (2005). "A Classification Framework for Anomaly Detection". In: *Journal of Machine Learning Research* 6.8, pp. 211–232.

Stone, Marshall H (Marshall Harvey) (1932). *Linear transformations in Hilbert space and their applications to analysis / by Marshall*

Harvey Stone. Colloquium publications / American Mathematical Society ; v. 15. Providence: American Mathematical Society.

Strens, Malcolm J. A. (2000). "A Bayesian Framework for Reinforcement Learning". In: *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*. Ed. by Pat Langley. Morgan Kaufmann, pp. 943–950.

Sun, Jimeng et al. (2007). "GraphScope: parameter-free mining of large time-evolving graphs". In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '07. New York, NY, USA: Association for Computing Machinery, pp. 687–696.

Tao, Terence (2012). *Topics in random matrix theory / Terence Tao.* Graduate studies in mathematics ; v. 132. Providence, R.I: American Mathematical Society.

Taylor, Benjamin M and Peter J Diggle (2014). "INLA or MCMC? A tutorial and comparative evaluation for spatial prediction in log-Gaussian Cox processes". In: *Journal of Statistical Computation and Simulation* 84.10, pp. 2266–2284.

Titsias, Michalis K. (2009). "Variational learning of inducing variables in sparse Gaussian processes". In: *Journal of Machine Learning Research*. Ed. by David van Dyk and Max Welling. Vol. 5. Proceedings of Machine Learning Research. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR, pp. 567–574.

Trecate, Giancarlo Ferrari, Christopher K.I. Williams, and Manfred Opper (1999). "Finite-dimensional approximation of Gaussian processes". In: *Advances in Neural Information Processing Systems*. MIT Press, pp. 218–224.

Tronarp, Filip and Toni Karvonen (2022). "Orthonormal Expansions for Translation-Invariant Kernels". In: *ArXiv preprint* abs/2206.08648.

UCI (2021). *Wikipedia Math Essentials Dataset*. UCI Machine Learning Repository.

Valiant, Leslie G (1979). "The complexity of computing the permanent". In: *Theoretical Computer Science* 8.2, pp. 189–201.

Walder, Christian J. and Adrian N. Bishop (2017). "Fast Bayesian Intensity Estimation for the Permanental Process". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 3579–3588.

Williams, Christopher K. I. and Matthias W. Seeger (2000). "The Effect of the Input Density Distribution on Kernel-based Classifiers". In: *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*. Ed. by Pat Langley. Morgan Kaufmann, pp. 1159–1166.

Wilson, James T. et al. (2020). "Efficiently sampling functions from Gaussian process posteriors". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 10292–10302.

Withers, C S (1985). "The moments of the multivariate normal". In: *Bulletin of the Australian Mathematical Society* 32.1, pp. 103–107.

Xu, Yuan (1994a). “Multivariate orthogonal polynomials and operator theory”. In: *Transactions of the American Mathematical Society* 343.1, pp. 193–202.

— (1994b). *Recurrence formulas for multivariate orthogonal polynomials*. Tech. rep. 206, pp. 687–702.

Yaglom, Akiva. M. (1987). *Correlation Theory of Stationary and Related Random Functions. Vol. I: Basic Results*. Vol. II. Springer series in Statistics Vol. 1. Springer, p. 538.

Yang, J., K. Miescke, and P. Mccullagh (2012). “Classification based on a permanental process with cyclic approximation”. In: *Biometrika* 99.4, pp. 775–786.

Zhu, Huaiyu et al. (1998). “Gaussian Regression and Optimal Finite Dimensional Linear Models”. In: *Neural Networks and Machine Learning*. Ed. by C.M. Bishop, pp. 167–184.