Carnegie Mellon University

A SPECTRAL SERIES APPROACH TO HIGH-DIMENSIONAL NONPARAMETRIC INFERENCE

A Dissertation Submitted to the Graduate School in Partial Fulfillment of the Requirements

For the Degree

Doctor of Philosophy
in
Statistics

by

Rafael Izbicki

Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213

April 2014

Dedicated to my grandmother Rebecca.

ABSTRACT

A key question in modern statistics is how to make efficient inferences for complex, high-dimensional data, such as images, spectra, and trajectories. While a large body of work has revolved on adapting nonparametric regression methods to high dimensions, statisticians have devoted less effort to redesigning estimators of other quantities to such settings. Some of these tasks are of key importance for the sciences; an example is the conditional density estimation problem, which plays an important role in modern cosmology. In this thesis, we propose a nonparametric framework for estimating unknown functions in high dimensions. The basic idea is to expand these functions in terms of a *spectral basis* – the eigenfunctions of a kernel-based operator. If the kernel is appropriately chosen, then the eigenfunctions adapt to the intrinsic geometry of the data, forming an efficient Fourier-like orthogonal basis for smooth functions on the data. We show how this framework can be used for estimating a regression curve, a conditional density, a likelihood function, or a density ratio. We provide theoretical guarantees on the developed estimators, we discuss their computational aspects, and we illustrate their use for several applications in astronomy, including estimation of photometric redshift distributions under selection bias.

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

- P.E. Freeman, R. Izbicki, A.B. Lee, J.A. Newman, C.J. Conselice, A.M. Koekemoer, J.M. Lotz, and M. Mozena. “New image statistics for detecting disturbed galaxy morphologies at high redshift." *Monthly Notices of the Royal Astronomical Society*, 434:282–295, 2013.

- R. Izbicki, A. B. Lee, and C. M. Schafer. “High-dimensional density ratio estimation with extensions to approximate likelihood computation." *Journal of Machine Learning Research (AISTATS track)*, 33:420–429, 2014.

- A. B. Lee, and R. Izbicki. “A spectral series approach to high-dimensional nonparametric regression", submitted for publication.

- R. Izbicki, and A. B. Lee. “Nonparametric conditional density estimation in a high-dimensional regression setting," in preparation.

- R. Izbicki, A. B. Lee, and P. E. Freeman. “Nonparametric density estimation under selection bias with applications to galaxy redshift prediction," in preparation.

Software implementation of the methods and estimators presented in this thesis are publicly available as an R package (R Development Core Team, 2010) at www.stat.cmu.edu/~rizbicki/specSeries.html.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my advisors Ann Lee and Peter Freeman for all the guidance and support in these years. Thanks not only for all the interesting and motivating questions, but also for being extremely patient and enthusiastic on discussing ideas, sharing knowledge, and giving me good advice. I am also very thankful for Chad Schafer for all insightful thoughts and conversations. Thanks to the rest of my committee members, Jessi Cisewski, Jeff Newman, Cosma Shalizi, and Larry Wasserman for all the valuable suggestions that contributed to this work. I am also grateful to Jing Lei and Ryan Tibshirani for the useful comments on some of the chapters of this thesis.

I would like to thank my parents Deborah and Meyer, my sister Sarah, and my grandparents Ita and Leon for being so supportive with the idea of me pursing a Ph.D. degree outside Brazil. I also owe a lot to Lea Veras, without whom it would have been almost impossible to complete this thesis. Thanks for all the encouragement, confidence, support, and patience. I wish to thank Tiago Mendonça for all the suggestions, in especial regarding the figures, and for all the years of friendship, even though being in another country. I am also very thankful to Mauricio Sadinle and Rafael Stern for their friendship and hours spent in front of a whiteboard, discussing ideas and proving bounds.

My sincere thanks also goes to Luís Gustavo Esteves, who not only taught me all the statistics I know, but has also been a great friend since my first year of undergraduate studies. Carlos Alberto de Bragança Pereira (Carlinhos) also deserves my acknowledgements for inumerous reasons, in special for teaching me to question everything I learn. Thanks to Julio Stern and Sergio Wechsler for all the guidance in these years.

Finally, I would like to acknowledge *Conselho Nacional de Desenvolvimento Científico e Tecnológico*, whose financial support (grant 200959/2010-7) has made it possible for me to pursue my Ph.D. degree outside Brazil.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOTATION

— $\mathfrak{L}^2(\mathcal{X}, P)$: the Hilbert space of square integrable functions with domain $\mathcal{X}$, image $\mathbb{R}$, and norm $\|g\|_P^2 = \langle g, g\rangle_P = \int_{\mathcal{X}} |g(\mathbf{x})|^2 dP(\mathbf{x})$.

— $\mathfrak{L}^2(\mathcal{X})$: $\mathfrak{L}^2(\mathcal{X}, P)$, where P is the Lebesgue measure.

— $\delta_{i,j}$: $\mathbb{I}(i \neq j) = \begin{cases} 1 & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$

ACRONYMS

ABC Approximate Bayesian Computation

CANDELS Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey

CDE Conditional Density Estimation

i.i.d. Independent and identically distributed

p.d.f. Probability Density Function

RKHS Reproducing Kernel Hilbert Space

SDSS Sloan Digital Sky Survey

Part I

MOTIVATION AND THESIS OVERVIEW

1

INTRODUCTION

A challenging problem in modern statistics is how to handle complex, high-dimensional data, such as images, spectra, and trajectories. While a large body of work has revolved around adapting nonparametric regression methods to high-dimensions, statisticians have devoted less effort to redesigning estimators of other quantities to such settings. However, some of these tasks are of extreme importance for the sciences; an example is the conditional density estimation problem, which plays an important role in modern cosmology (e.g., Sheldon et al. 2012). It is therefore essential to develop efficient tools for performing such tasks in high-dimensions.

In this thesis we provide a new framework for nonparametric high-dimensional inference; that is, we present a tool that can be used for estimation tasks other than regression analysis. The idea is to approximate functions of interest by using series expansions. More precisely, we estimate functions by expanding them into a well chosen Hilbert basis that adapts to the data structure. Unfortunately, standard Fourier-like bases used in traditional nonparametric curve estimation are not well suited for high-dimensional problems. Instead, we propose to use *spectral bases* - the eigenfunctions of kernel-based operators. We will see that such bases allows one to build flexible methods that have good computational and statistical properties for high-dimensional data.

It is widely known that, without extra assumptions, it is not possible to learn functions in high-dimensions with a reasonable sample size, a phenomenon known

as the "curse of dimensionality" (Bellman 1961). The assumption we make in this thesis is that although the data $\mathbf{x}$ live in a high-dimensional space, they have a *sparse structure*. "Sparse" here refers to a situation where the underlying data distribution, $P(\mathbf{x})$, places most of its mass on a subset $\mathcal{X}$ of $\mathbb{R}^d$ of small Lebesgue measure. This scenario includes, but is not limited to, hyperplanes, Riemannian submanifolds of $\mathbb{R}^d$, and high-density clusters separated by low-density regions. In practice, $\mathbf{x}$ often lives in spaces with these structures; e.g., this is typically the case of images, spectra, trajectories, movies, etc; see Figure 1.1. See also Tenenbaum et al. (2000), Belkin and Niyogi (2001), Kpotufe (2010), and Cheng and Wu (2013) for many other examples. As an illustration, consider the ZIP Code database from USPS (Hastie et al. 2001). These data are composed of images stored as 16×16 matrices, i.e., the sample space has dimension 256. However, not every 16×16 matrix represents an image of a digit. In fact, only a very small number of them represent the image of *any* object a human being would be able to recognize. Hence, the real dimensionality of ZIP Code data is much smaller than 256, although it not always obvious how to take this into account in the analyses of interest.

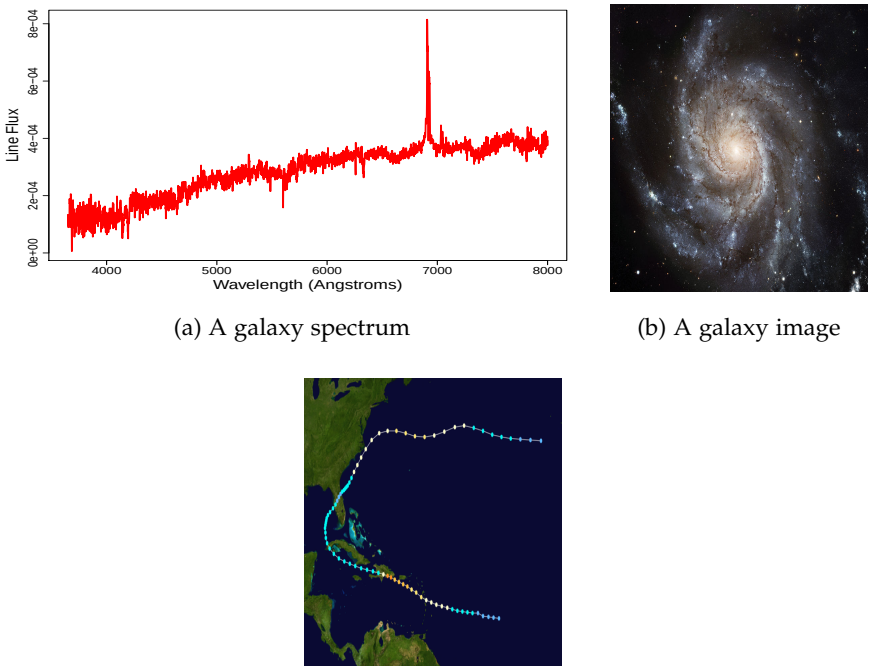
1.1 Previous Work

Although we leave a thorough discussion of previous literature to Chapters 3-6 – in which we discuss how to apply the spectral series method to different tasks – here we briefly overview it.

1.1.1 *High-Dimensional Inference*

Several works attempt to avoid the curse of dimensionality for the specific task of estimating a *regression curve*. While many approaches are based on variable selection and lasso-type regularization (e.g., Tibshirani 1996; Lafferty and Wasserman 2008; Ravikumar et al. 2009; Fan et al. 2011), there has been a growing interest in methods that take advantage of the fact that data often live close to a submanifold

of the sample space (e.g., Pelletier 2006; Bickel and Li 2007; Aswani et al. 2011; Cheng and Wu 2013). Another important class of algorithms consists of performing a regularization in a Reproducing Kernel Hilbert Space (RKHS), the so-called kernel machine learning (Schölkopf and Smola 2001; Cucker and Zhou 2007). As we will see, this technique has some similarities to our method. See Chapter 3 for a more detailed review of these and other regression methods.



(a) A galaxy spectrum

(b) A galaxy image

(c) A hurricane track

Figure 1.1: Examples of data with large *ambient* dimensionality, but small *intrinsic* dimensionality. Although such data has hundreds, or even thousands of dimensions, these are highly redundant. Source: Galaxy image provided by ESA/Hubble.

On the other hand, there are very few attempts of estimating other unknown quantities than the regression curve in high dimensions. Most of them rely on a dimension reduction of the covariates prior to implementation (e.g., Pelletier 2005; Hall and Yao 2005; Fan et al. 2009; Sugiyama et al. 2011; Buchman et al. 2011). As is the case with any data reduction, such a step can result in significant

loss of information, which is in general hard to quantify both theoretically and empirically. Only a few other works propose different methodologies (e.g., Hall et al. 2004; Liu et al. 2007; Efromovich 2007; Buchman 2011). In the subsequent chapters we summarize relevant work for the specific tasks we deal with.

1.1.2 *Spectral Methods*

Spectral methods have emerged as an efficient tool for finding low-dimensional structure in high-dimensional data via the eigendecomposition of certain data-dependent matrices. They have been successfully used in the literature of clustering (Shi et al. 2009), classification (Sinha and Belkin 2009), dimension reduction and manifold learning (Schölkopf et al. 1997; Belkin and Niyogi 2003; Coifman and Lafon 2006). On the other hand, only a limited amount of work use them as way of creating a basis for approximating functions of interest (Nadler et al. 2009; Zhou and Srebro 2011; Ji et al. 2012). As a matter of fact, these are typically concerned with the regression function only, and deal mainly with a semi-supervised learning setting. Moreover, they do not explore the computational benefits one can obtain via orthogonality of the basis functions. An exception to this is Hendriks (1990), who uses spectral series to approximate density functions on manifolds, however he assumes the manifold is known a priori, and does not deal with high-dimensional problems.

In this work we will show how spectral methods can be efficiently used for estimating functions in high-dimensional domains without going through dimension reduction.

1.2 THESIS STRUCTURE

In Chapter 2, we first briefly review traditional series methods, and then discuss *spectral bases*. We also explain why they are good candidates for approximating functions defined on high-dimensional sample spaces. In each of the subsequent

chapters, we describe the details of how such bases can be used for the following tasks:

Chapter 3 : Estimation of a regression function

Chapter 4 : Estimation of a conditional density in a regression setting

Chapter 5 : Estimation of the ratio between two probability densities functions

Chapter 6 : Estimation of a likelihood function

We present comparisons with standard methods, as well as rates of convergence. We then show our main application in Chapter 7, where we use our conditional density estimator for photometric redshift prediction in the Sloan Digital Sky Survey data (Aihara et al. 2011), an important problem in cosmology. In particular, we discuss how selection bias can be taken into account in this task. Final remarks and some ideas for future work are in Chapter 8. Proofs of selected theorems are shown in Appendices A and B.

2

OVERVIEW OF THE SPECTRAL SERIES METHOD

We begin this chapter by reviewing *traditional* orthogonal series methods. Such methods make use of Fourier-like basis, which are well suited for low-dimensional problems. Then, we introduce *spectral bases* and describe some of their advantages over standard bases in high-dimensional problems.

2.1 Traditional Orthogonal Series Methods

Let $\mathbf{X}$ be a random vector defined over a domain $\mathcal{X} \subseteq \mathbb{R}^d$, and suppose we are interested in approximating an unknown function

$$g : \mathcal{X} \longrightarrow \mathbb{R}$$
$$\mathbf{x} \longrightarrow g(\mathbf{x})$$

based on an i.i.d. sample $\mathbf{X}_1, \ldots, \mathbf{X}_n$ of $\mathbf{X}$. An example of a function of interest is the probability density function (p.d.f.) of the random vector $\mathbf{X}$, $g(\mathbf{x}) := f(\mathbf{x})$. The function g can also be the regression of a random variable $Y \in \mathbb{R}$ on the covariates $\mathbf{x} \in \mathcal{X}$, $g(\mathbf{x}) := \mathbb{E}[Y|\mathbf{x}]$. Other examples of such functions are given in the subsequent chapters.

The idea of *traditional* orthogonal series methods is to expand g in terms of an orthonormal basis function $\{\psi_j(\mathbf{x})\}_{j\in\mathbb{N}}$ of $\mathfrak{L}^2(\mathcal{X})$.[1] That is, the method consists in first writing

$$g(\mathbf{x}) = \sum_{j\in\mathbb{N}} \beta_j \psi_j(\mathbf{x}) \tag{2.1}$$

and then estimating the coefficients $\{\beta_j\}_j$ based on the data. The details change according the function one is approximating and the data that are available. As an example, if g is the p.d.f. of $\mathbf{X}$, $f(\mathbf{x})$, we have

$$\beta_j = \langle g, \psi_j \rangle = \int g(\mathbf{x})\psi_j(\mathbf{x})d\mathbf{x} = \int \psi_j(\mathbf{x})f(\mathbf{x})d\mathbf{x} = \mathbb{E}\left[\psi_j(\mathbf{X})\right],$$

which motivates the estimator

$$\widehat{\beta}_j := \frac{1}{n}\sum_{k=1}^{n} \psi_j(\mathbf{x}_k)$$

Hence, a simple estimator of the p.d.f. is given by

$$\widehat{f}(\mathbf{x}) = \sum_{j=1}^{J} \widehat{\beta}_j \psi_j(\mathbf{x}),$$

where the truncation point J is typically chosen so as to control the bias/variance tradeoff. Because $\{\psi_j\}_j$ is typically chosen so as to have lower-order terms smoother than higher-order terms (see Figure 2.1), smooth functions generally require a small value of J.

If $d = 1$, a popular choice for the basis $\{\psi_j\}_j$ is the Fourier basis:

$$\begin{cases} \psi_1(x) := 1 \\ \psi_{2j}(x) := \sqrt{2}\sin(2\pi j x), \ j = 1, 2, \ldots \\ \psi_{2j+1}(x) := \sqrt{2}\cos(2\pi j x), \ j = 1, 2, \ldots \end{cases}$$

1 We assume $g \in \mathfrak{L}^2(\mathcal{X})$. Also, notice that orthonormality in traditional orthogonal series methods is with respect to the Lebesgue measure:

$$\int_{\mathcal{X}} \psi_i(\mathbf{x})\psi_j(\mathbf{x})d\mathbf{x} = \delta_{i,j}.$$

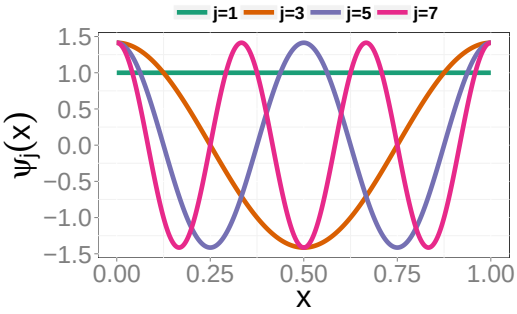


Figure 2.1: Some elements of a Fourier basis. Lower order terms are smoother than higher order terms. They are good candidates for approximating smooth functions in one dimension.

If $d > 1$, $\{\psi_j(\mathbf{x})\}_j$ is typically built using d *tensor products* for each of the coordinates of $\mathbf{x}$. For instance, if $d = 2$, the tensor products basis is

$$\{\psi_{i,j}(\mathbf{x}) = \psi_i(x_1)\psi_j(x_2) : i, j \in \mathbb{N}\},$$

where $\mathbf{x} = (x_1, x_2)$, and $\{\psi_i(\mathbf{x}_1)\}_i$ and $\{\psi_j(\mathbf{x}_2)\}_j$ are bases for functions in $\mathfrak{L}^2(\mathbb{R})$. This is a basis for $\mathfrak{L}^2(\mathbb{R}^2)$ (Wasserman 2006). We refer the reader to Efromovich (1999) for a comprehensive account of orthogonal series methods.

Orthogonal series methods present several advantages over other nonparametric estimators. For instance, they are usually easy and fast to implement, and additionally offer a compression of the data in terms of a few Fourier coefficients. However, there is currently no way of extending them to higher dimensions. This is because tensor products quickly become computationally intractable even for as few as 10 covariates. Moreover, traditional bases do not always capture the complexity of such data which, despite their apparent high dimensionality, are often highly redundant with a low intrinsic dimensionality. In the next section we show a family of bases that overcomes these issues, the *spectral bases*[2].

2 These are often called *eigenbases* in the literature.

2.2 The Spectral Series Method

We start by describing how to build a spectral basis. Our starting point is $K(\mathbf{x}, \mathbf{y})$, a *Mercer kernel* that measures the similarity between samples $\mathbf{x}$ and $\mathbf{y}$. That is, $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ is bounded, symmetric, and positive definite[3] function. An example is the Gaussian kernel,

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(\frac{-d^2(\mathbf{x}, \mathbf{y})}{4\epsilon}\right),$$

where $d(\cdot, \cdot)$ is the Euclidean distance in $\mathbb{R}^d$ and $\epsilon > 0$ is a user-chosen bandwidth; see Schölkopf and Smola (2001) for other examples. For now we assume K is fixed; in the subsequent chapter we describe some advantages of specific kernels, as well as methods for choosing tuning parameters such as the bandwidth ϵ.

Let $P(\mathbf{x})$ be the distribution of the random variable $\mathbf{X}$. As is standard in the spectral methods literature (e.g, Shi et al. 2009), we define the following integral operator

$$\begin{aligned} \mathbf{K} : \mathfrak{L}^2(\mathcal{X}, P) &\longrightarrow \mathfrak{L}^2(\mathcal{X}, P) \\ \mathbf{K}(g)(\mathbf{x}) &= \int_{\mathcal{X}} K(\mathbf{x}, \mathbf{y}) g(\mathbf{y}) dP(\mathbf{y}) \end{aligned} \tag{2.2}$$

The operator $\mathbf{K}$ has a countable number of eigenfunctions $\{\psi_j\}_{j \in \mathbb{N}}$ with respective eigenvalues $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant 0$ (Minh et al. 2006).

Definition 2.1. *We refer to the eigenfunctions $\{\psi_j\}_{j \in \mathbb{N}}$ of the kernel operator in Equation 2.2 as a* spectral basis.

These eigenfunctions form an orthonormal basis of $\mathfrak{L}^2(\mathcal{X}, P)$ (Minh 2010). Hence, we can expand any function in $\mathfrak{L}^2(\mathcal{X}, P)$ into them. We propose to use $\{\psi_j\}_j$ as a basis to approximate functions of $\mathbf{x}$.

There are two main reasons why spectral bases are ideal candidates for approximating smooth functions of $\mathbf{x}$ in high dimensions:

3 i.e., Matrix (2.3) is positive definite $\forall n \in \mathbb{N}$ and $\forall \mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$.

1. The eigenfunctions are *adapted to the intrinsic geometry of the data*. More precisely, they are concentrated around high-density regions, and, in the case the domain $\mathcal{X}$ is a submanifold of $\mathbb{R}^d$ (where d may be large), the bases behave like a Fourier basis adapted to the intrinsic geometry of the data, where lower-order terms are smoother than higher-order terms.[4] As an example, in Figure 2.2 we show the eigenfunctions of the kernel operator when the domain of the data is a submanifold (a spiral) of $\mathbb{R}^2$. Compare it to Figure 2.1; the basis behaves like a Fourier basis along the direction of the spiral. It follows that if $g(\mathbf{x})$ is smooth relative to P, then we only need a few eigenfunctions to approximate it. As we will show, the adaptation of the basis yields convergence rates that depend only on the *intrinsic* dimensionality of the data, instead of the potentially larger *ambient* dimensionality.

2. Unlike bases used in traditional series methods, the eigenfunctions are *orthogonal with respect to* $P(\mathbf{x})$, the underlying data distribution, as opposed to the Lebesgue measure of the ambient space (Bengio et al. 2004). That is,

$$\int_{\mathcal{X}} \psi_i(\mathbf{x})\psi_j(\mathbf{x})dP(\mathbf{x}) = \delta_{i,j}.$$

   We will see that for many problems this leads to faster estimators of the expansion coefficients of Equation 2.1. Moreover, there is no need for using tensor products in high dimensions, and thus our basis is computationally more efficient than traditional bases from Section 2.1.

We will also see that in many problems this framework has the advantage of allowing a natural extension to *semi-supervised learning* settings where, in addition to the labeled sample, we also observe an unlabeled sample. Moreover, the eigenfunctions can additionally be used for data visualization.

4 This is because Euclidean distance is locally the same as geodesic distance; see, e.g., Shi et al. 2009 for a derivation.

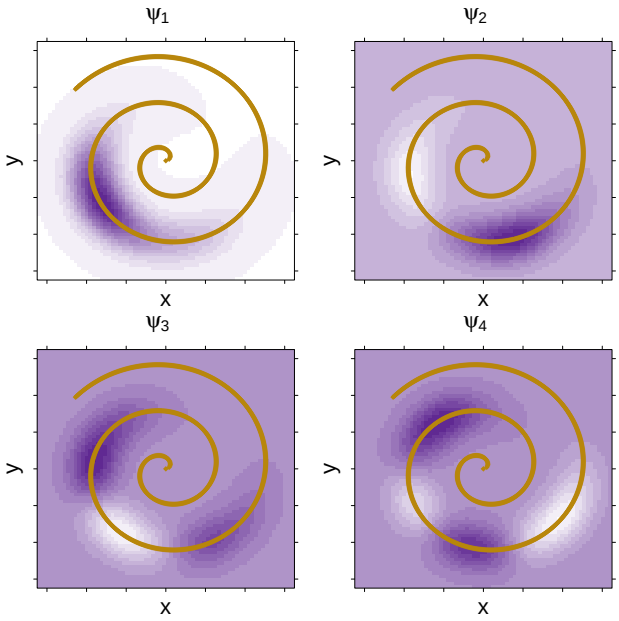


Figure 2.2: Level sets of the top eigenfunctions of the Gaussian kernel operator when the domain of the data $\mathbf{x} = (x, y)$ is on a spiral. The eigenfunctions form a Fourier-like basis adapted to the geometry of the data, and are well-suited for approximating smooth functions of $\mathbf{x}$ on this domain. Compare this figure with Figure 2.1.

Remark: The operator from Equation 2.2 is implicitly used in Kernel Principal Component Analysis (Schölkopf et al.). Hence, from here on, we denote it as the Kernel PCA operator. Notice, however, that we do not use it with the goal of performing dimension reduction.

2.2.1 *Estimating the Basis*

As $P(\mathbf{x})$ is unknown, we need to estimate $\{\psi_j\}_j$. This can be done by first computing the Gram Matrix

$$\mathbf{G} = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & K(\mathbf{x}_1, \mathbf{x}_2) & \cdots & K(\mathbf{x}_1, \mathbf{x}_n) \\ K(\mathbf{x}_2, \mathbf{x}_1) & K(\mathbf{x}_2, \mathbf{x}_2) & \cdots & K(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ K(\mathbf{x}_n, \mathbf{x}_1) & K(\mathbf{x}_n, \mathbf{x}_2) & \cdots & K(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix} \tag{2.3}$$

Let

$$\widetilde{\psi}_j := \left(\widetilde{\psi}_j(\mathbf{x}_1), \ldots, \widetilde{\psi}_j(\mathbf{x}_n)\right)$$

be the j-th eigenvector of Matrix 2.3, and let $\widehat{l}_j$ be its associated eigenvalue, where we sort the eigenvectors by decreasing order of eigenvalues, and normalize them such that $\sum_{k=1}^{n} \widetilde{\psi}_j^2(\mathbf{x}_k) = 1$. A consistent estimator of ψ_j is

$$\widehat{\psi}_j(\mathbf{x}) = \frac{\sqrt{n}}{\widehat{l}_j} \sum_{k=1}^{n} \widetilde{\psi}_j(\mathbf{x}_k) K(\mathbf{x}, \mathbf{x}_k). \tag{2.4}$$

This estimator is the Nyström extension of the eigenvector $\widetilde{\psi}_j$ to out-of-sample values $\mathbf{x}$ (Bengio et al. 2004; Drineas and Mahoney 2005).

2.2.2 *Scalability*

A naive implementation of the eigendecomposition of the Gram matrix, $\mathbf{G}$, has computational time $\mathcal{O}(n^3)$. However, it is possible to speed it up, and also make it more memory-efficient. These make it possible to scale spectral series estimators to larger datasets.

A simple approach for fast approximate eigendecomposition— named Randomized SVD — was proposed in Halko et al. (2011). We summarize it in Algorithm 1 for the case of a Gram matrix. This method leads to considerably faster solutions when J, the number of desired eigenvalues, is much smaller than n (in this case, it is roughly $\mathcal{O}(n^2)$); moreover, as we will see in Chapter 4, in general there is no substantial decrease in statistical performance.

It is also possible to make the algorithm memory-efficient by inducing sparsity on the Gram matrix. This can be achieved via thresholding, i.e., by assigning a value 0 to all entries with $K(\mathbf{x}_i, \mathbf{x}_j)$ smaller than a small user-chosen $\xi > 0$. If one uses local kernels (e.g., the Gaussian kernel), samples that are far from each other will have a small $K(\mathbf{x}_i, \mathbf{x}_j)$, and hence Matrix (2.3) will be very sparse after thresholding. This allows $\mathbf{G}$ to be stored using less memory. Sparsity can also be used to make the eigendecomposition step even faster. Although we do not explore

Algorithm 1 Randomized Singular Value Decomposition (SVD)

Input: $n \times n$ Gram Matrix $\mathbf{G}$, scalars J, p and q.

▷ The default input values are typically $p = 10$ and $q = 1$

Output: Eigenvectors $\mathbf{U}$ and eigenvalues $\mathbf{\Lambda}$

1: generate $\mathbf{\Omega}$, a $n \times (J + p)$ Gaussian matrix with components of mean 0 and variance 1

2: let $\mathbf{Z} = \mathbf{G\Omega}$, $\mathbf{Y} = \mathbf{G}^{q-1}\mathbf{Z}$

3: compute an orthonormal matrix $\mathbf{Q}$ via $\mathbf{QR}$ decomposition on $\mathbf{Y}$

4: compute the SVD of $\mathbf{Q}^t\mathbf{Z}(\mathbf{Q}^t\mathbf{\Omega})^{-1} = \mathbf{U\Sigma U}^t$

5: **return** $\mathbf{U}$ and $\mathbf{\Lambda}$ =diag($\mathbf{\Sigma}$)

this here, we refer the reader to Halko et al. (2011) for further improvements, such as exploitation of multi-processor architectures.

2.2.3 *Connection to Dimension Reduction Methods*

The eigenfunctions $\{\psi_j\}_j$ have a dual interpretation:

1. They define new coordinates of the data which are primarily useful for manifold learning, data visualization, and nonlinear dimensionality reduction. That is, it transforms the data according to a so-called "eigenmap"

$$\mathbf{x} \mapsto (\psi_1(\mathbf{x}), \psi_2(\mathbf{x}), \ldots, \psi_J(\mathbf{x})).$$

   The eigenmap can be used for data visualization and manifold learning. More generally , if $J < d$, then we are effectively reducing the dimensionality of the problem by mapping the data from $\mathbb{R}^d$ to $\mathbb{R}^J$.

2. They form a Hilbert basis for functions on the data and hence are a means to nonparametric curve estimation via the classical series method.

There is a large body of work addressing the first perspective; see, e.g., Laplacian maps (Belkin and Niyogi 2003), Hessian maps (Donoho and Grimes 2003), Diffusion Maps (Coifman et al. 2005), Euclidean Commute Time maps (Saerens

et al. 2004), and Kernel Principal Component Analysis (Schölkopf et al. 1997). In this work, we are mainly concerned with the second view. We will see that in some specific problems there is an equivalence between the first and second perspective, although this is not the case in general.

2.2.4 *Variations of the Kernel Operator*

Several variants of the operator from Equation (2.2) can be defined. For example, it is usual to work with non-symmetric kernels in spectral methods literature (e.g., Lee and Wasserman 2010). Such operators also yield interesting bases functions. In Chapters 3 and 4 we will explore one of these, namely the *diffusion operator*. We will see that, although in our experiments bases derived from the diffusion operator lead to similar performance to those based on Equation (2.2), they have better understood theoretical properties, which in turn yield more interesting bounds. More precisely, the limit for the bandwidth $\epsilon \to 0$ is well-defined. In particular, there is a series of works on the convergence of the graph Laplacian to the Laplace-Beltrami operator on Riemannian manifolds (Coifman and Lafon 2006; Belkin and Niyogi 2005; Hein et al. 2005; Singer 2006; Giné and Koltchinskii 2006). As Fourier functions originate from solving a Laplace eigenvalue problem on a bounded domain, the eigenfunctions of the diffusion operator can be seen as a generalization of Fourier series to manifolds. This makes the diffusion kernel with decreasing bandwidth especially appealing. As we shall see in Section 3.4.1, the connection to the Laplace operator also implies a direct link between Sobolev differentiability and sparsity.

For the diffusion operator, we assume the kernel K is a local, radially symmetric function

$$K_\epsilon(\mathbf{x}, \mathbf{y}) = g\left(d(\mathbf{x}, \mathbf{y})/\sqrt{\epsilon}\right),$$

such that the elements $K_\epsilon(\mathbf{x}, \mathbf{y})$ are positive and bounded for all $x, y \in \mathcal{X}$. We use the notation K_ϵ to emphasize the dependence of K on the kernel bandwidth. The first step is to renormalize the kernel according to

$$a_\epsilon(\mathbf{x}, \mathbf{y}) = \frac{K_\epsilon(\mathbf{x}, \mathbf{y})}{p_\epsilon(x)},$$

where $p_\epsilon(\mathbf{x}) = \int K_\epsilon(\mathbf{x}, \mathbf{y}) dP(\mathbf{y})$. We refer to $a_\epsilon(\mathbf{x}, \mathbf{y})$ as the *diffusion kernel* (Meila and Shi 2001). As Lee and Wasserman (2010), we define the "diffusion operator" $\mathbf{A}_\epsilon$ according to

$$\mathbf{A}_\epsilon(h)(\mathbf{x}) = \int_{\mathcal{X}} a_\epsilon(\mathbf{x}, \mathbf{y}) h(\mathbf{y}) dP(\mathbf{y}). \tag{2.5}$$

The operator $\mathbf{A}_\epsilon$ has a discrete set of non-negative eigenvalues $\lambda_{\epsilon,0} = 1 \geqslant \lambda_{\epsilon,1} \geqslant \ldots \geqslant 0$ with associated eigenfunctions $(\psi_{\epsilon,j})_j$. The eigenfunctions are orthogonal with respect to the weighted L^2 inner product

$$\langle f, g \rangle_\epsilon = \int_{\mathcal{X}} f(\mathbf{x}) g(\mathbf{x}) dS_\epsilon(\mathbf{x}),$$

where

$$S_\epsilon(A) = \frac{\int_A p_\epsilon(\mathbf{x}) dP(\mathbf{x})}{\int p_\epsilon(\mathbf{x}) dP(\mathbf{x})}$$

can be interpreted as a *smoothed* version of P. The density of S_ϵ with respect to P is $s_\epsilon(\mathbf{x}) = \frac{p_\epsilon(\mathbf{x})}{\int p_\epsilon(\mathbf{y}) P(\mathbf{y})}$.

Remark: It can be shown that the first eigenfunction of the diffusion operator is constant. Because of this, we call it "trivial eigenfunction", and denote it by $\psi_0(\mathbf{x}) \equiv 1$.

2.2.4.1 *Estimating the Diffusion Basis*

The ideas used to estimate the diffusion basis are similar to those used for estimating quantities associated to the Kernel PCA operator from Equation (2.2). More precisely, given $\mathbf{x}_1, \ldots, \mathbf{x}_n$, use the kernel K_ϵ to construct a row-stochastic matrix $\mathbb{A}_\epsilon$, where

$$\mathbb{A}_\epsilon(i, j) = \frac{K_\epsilon(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{l=1}^n K_\epsilon(\mathbf{x}_i, \mathbf{x}_l)} \tag{2.6}$$

for $i, j = 1, \ldots, n$. Let $\widehat{p}_\epsilon(\mathbf{x}) = \frac{1}{n}\sum_{j=1}^{n} K_\epsilon(\mathbf{x}, \mathbf{x}_j)$. The stationary measure S_ϵ can be estimated by

$$\widehat{s}_\epsilon(\mathbf{x}_i) = \frac{\widehat{p}_\epsilon(\mathbf{x}_i)}{\sum_{j=1}^{n} \widehat{p}_\epsilon(\mathbf{x}_j)}. \tag{2.7}$$

To estimate the eigenfunctions $\psi_{\epsilon,1}, \ldots, \psi_{\epsilon,J}$ of the continuous diffusion operator A_ϵ in Eq. 2.5, we first calculate the eigenvalues $\lambda^{\mathbb{A}}_{\epsilon,1}, \ldots, \lambda^{\mathbb{A}}_{\epsilon,J}$ and the associated (orthogonal) eigenvectors $\tilde{\psi}^{\mathbb{A}}_{\epsilon,1}, \ldots, \tilde{\psi}^{\mathbb{A}}_{\epsilon,J}$ of the symmetrized kernel matrix $\tilde{\mathbb{A}}_\epsilon$, where

$$\tilde{\mathbb{A}}_\epsilon(i, j) = \frac{K_\epsilon(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{\sum_l K_\epsilon(\mathbf{x}_i, \mathbf{x}_l)}\sqrt{\sum_l K_\epsilon(\mathbf{x}_l, \mathbf{x}_j)}}. \tag{2.8}$$

We normalize the eigenvectors so that $\frac{1}{n}\sum_{i=1}^{n} \tilde{\psi}^{\mathbb{A}}_{\epsilon,j}(i)\tilde{\psi}^{\mathbb{A}}_{\epsilon,k}(i) = \delta_{j,k}$, and define the new vectors $\psi^{\mathbb{A}}_{\epsilon,j}(i) = \tilde{\psi}^{\mathbb{A}}_{\epsilon,j}(i)/\sqrt{\widehat{s}_\epsilon(\mathbf{x}_i)}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, J$.

The n-dimensional vector $\psi^{\mathbb{A}}_{\epsilon,j}$ can be regarded as estimates of $\psi_{\epsilon,j}(\mathbf{x})$ at the observed values $\mathbf{x}_1, \ldots, \mathbf{x}_n$. As in the case of the Kernel PCA operator, we estimate the function $\psi_{\epsilon,j}(\mathbf{x})$ at values of $\mathbf{x}$ not corresponding to one of the $\mathbf{x}_i$'s using Nyström method. The idea is to first rearrange the eigenfunction-eigenvalue equation $\lambda_{\epsilon,j}\psi_{\epsilon,j} = \mathbf{A}_\epsilon\psi_{\epsilon,j}$ as

$$\psi_{\epsilon,j}(\mathbf{x}) = \frac{\mathbf{A}_\epsilon\psi_{\epsilon,j}}{\lambda_{\epsilon,j}} = \frac{1}{\lambda_{\epsilon,j}}\int \frac{K_\epsilon(\mathbf{x}, \mathbf{y})}{\int K_\epsilon(\mathbf{x}, \mathbf{y})\mathrm{d}P(\mathbf{y})}\psi_{\epsilon,j}(\mathbf{y})\mathrm{d}P(\mathbf{y}),$$

and use the kernel-smoothed estimate

$$\widehat{\psi}_{\epsilon,j}(\mathbf{x}) = \frac{1}{\widehat{\lambda}_{\epsilon,j}}\sum_{i=1}^{n} \frac{K_\epsilon(\mathbf{x}, \mathbf{x}_i)}{\sum_{l=1}^{n} K_\epsilon(\mathbf{x}, \mathbf{x}_l)}\widehat{\psi}_{\epsilon,j}(\mathbf{x}_i). \tag{2.9}$$

for $\widehat{\lambda}_{\epsilon,j} > 0$.

In the remaining of this thesis we describe in details how to use spectral series to estimate certain functions of interest.

Part II

THE USE OF SPECTRAL SERIES

3

REGRESSION ESTIMATION

Estimating a regression curve $\mathbb{E}[Z|\mathbf{x}]$ is a key problem in statistics. Therefore, this is our starting point for illustrating the spectral series method.

3.1 Introduction

In high-dimensional regression estimation, much research has revolved around variable selection and the problem of recovering a sparse coefficient vector in the *original* coordinate system. That is, low-dimensional structure is apparent in the *original* space. Such approaches include, for example, lasso-type regularization (Tibshirani 1996), the Dantzig selector (Candès and Tao 2005) and RODEO (Lafferty and Wasserman 2008). These methods are widely applicable to a range of different situations, but are known to suffer from collinearity or near-collinearity of predictors. This has prompted work on extensions, e.g., the grouped lasso (Yuan and Lin 2006) and elastic net (Zou and Hastie 2005), that take groupings of covariates into account. Similarly, sparse additive models (Ravikumar et al. 2009) can incorporate lower-order interactions between covariates but, like lasso-type estimators, they are not directly applicable to the type of collinearities observed in, e.g., images and spectra.

At the same time, there has been a growing interest in statistical methods that explicitly consider (sparse) geometric structure *in the data themselves*. Most traditional dimension-reducing regression techniques, e.g., principal component regression

(PCR; Jolliffe 2002) and partial least squares (PLS; Wold et al. 2001), are based on *linear* data transformations and enforce sparsity of the regression in a rotated space. More recently, Bickel and Li (2007), Aswani et al. (2011) and Cheng and Wu (2013) have studied local regression methods on *non-linear* manifolds. In Aswani et al. (2011), the authors propose a geometry-based regularization scheme designed for a setting where the predictors lie on a lower-dimensional nonlinear manifold. Under the manifold assumption, their approach is to first use a local covariance matrix to estimate the manifold at a point, and then penalize regression coefficients perpendicular to the manifold direction. Similarly, Cheng and Wu (2013) propose to first estimate the dimensionality of the manifold, and then perform a local linear regression on an estimated tangent plane.

In this chapter we will show how spectral series can be effectively used to estimate a regression function in high dimensions. Figure 3.1, for example, shows a 2D visualization of the Isomap face data using the eigenvectors of the Gaussian kernel as coordinates. Assume we want to estimate the pose of the faces. How does one solve a regression problem where the predictors are *entire* images? Traditional methods do not cope well with this task while our approach can use high-dimensional complex data objects as predictors without a prior dimension reduction step; we will return to the face pose estimation problem in Section 3.5.1.

This chapter is organized as follows. In Section 3.2, we describe the construction of the spectral series method for regression estimation. Section 3.3 discusses the connection to related work in machine learning and statistics. In Section 3.4, we provide some theoretical guarantees of the basis method. Finally, in Section 3.5, we compare the performance of the series estimator with other nonparametric estimators for three high-dimensional data sets with, respectively, images of faces, galaxy spectra, and images of galaxies.

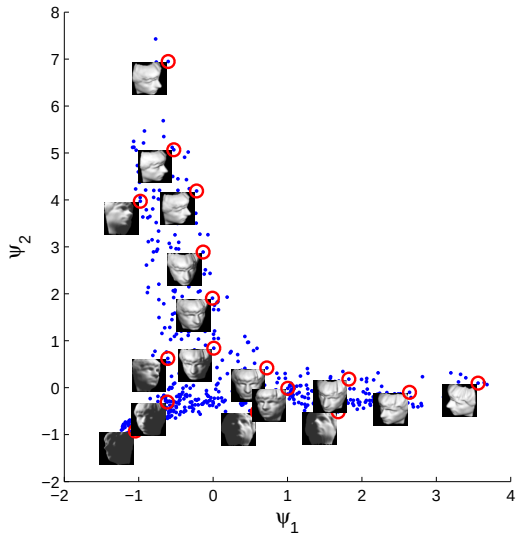


Figure 3.1: Embedding of the Isomap face data using the first two non-trivial eigenvectors of the Gaussian diffusion kernel.

3.2 Methodology

Let $(Z_1, \mathbf{X}_1), \ldots, (Z_n, \mathbf{X}_n)$ denote an i.i.d. sample, where $\mathbf{X}_i \in \mathcal{X} \subseteq \mathbb{R}^d$, and $Z_i \in \mathbb{R}$. Our goal is to estimate the regression function

$$r(\mathbf{x}) = \mathbb{E}(Z|\mathbf{X} = \mathbf{x}).$$

We now show how the eigenfunctions from both kernel PCA and the diffusion operators (Eqs. 2.2 and 2.5, respectively) can be used to implement the spectral series estimator.

3.2.1 *Kernel PCA Operator*

Assume a kernel K is fixed, and let $\{\psi_j\}_{j\in\mathbb{N}}$ be the orthonormal basis of the operator of Equation (2.2). The expansion of the regression function r onto this basis is given by

$$r(\mathbf{x}) = \sum_{j \geqslant 1} \beta_j \psi_j(\mathbf{x}),$$

where

$$\beta_j = \int_{\mathcal{X}} \psi_j(\mathbf{x}) r(\mathbf{x}) dP(\mathbf{x}) = \int_{\mathcal{X}} \psi_j(\mathbf{x}) \mathbb{E}[Z|\mathbf{x}] dP(\mathbf{x}) = \mathbb{E}[Z\psi_j(\mathbf{X})]$$

Notice that the orthogonality of the spectral basis with respect to $P(\mathbf{x})$ is the key for β_j to be simply $\mathbb{E}[Z\psi_j(\mathbf{X})]$.

The spectral series estimator of the regression function is therefore given by

$$\widehat{r}(\mathbf{x}) = \sum_{j=1}^{J} \widehat{\beta}_j \widehat{\psi}_j(\mathbf{x}), \tag{3.1}$$

where $\widehat{\psi}_j$'s are estimated as in Section 2.2.1, and

$$\widehat{\beta}_j = \frac{1}{n} \sum_{k=1}^{n} z_k \widehat{\psi}_j(\mathbf{x}_k).$$

We choose J according to Section 3.2.3.

3.2.2 *Diffusion Operator*

If one uses the basis $(\psi_{\epsilon,i})_i$ given by the diffusion operator (Eq. 2.5)[1], the expansion is instead given by

$$r(\mathbf{x}) = \sum_{j \geqslant 0} \beta_{\epsilon,j} \psi_{\epsilon,j}(\mathbf{x}),$$

where

$$\beta_{\epsilon,j} = \int_{\mathcal{X}} \psi_{\epsilon,j}(\mathbf{x}) r(\mathbf{x}) dS_\epsilon(\mathbf{x}) = \int_{\mathcal{X}} \psi_{\epsilon,j}(\mathbf{x}) \mathbb{E}[Z|\mathbf{x}] s_\epsilon(\mathbf{x}) dP(\mathbf{x}) = \mathbb{E}[Z\psi_{\epsilon,j}(\mathbf{X}) s(\mathbf{X})].$$

Hence, our estimator is

$$\widehat{r}(\mathbf{x}) = \sum_{j=0}^{J} \widehat{\beta}_j \widehat{\psi}_j(\mathbf{x}), \tag{3.2}$$

where

$$\widehat{\beta}_{\epsilon,j} = \frac{1}{n} \sum_{i=1}^{n} Z_i \widehat{\psi}_{\epsilon,j}(\mathbf{x}_i) \widehat{s}_\epsilon(\mathbf{x}_i), \tag{3.3}$$

where $\widehat{\psi}_{\epsilon,j}$ and $\widehat{s}_\epsilon$ were discussed in Section 2.2.4.1.

1 Recall that for the diffusion basis we are especially interested in the case $\epsilon \longrightarrow 0$, and hence emphasize the dependence of the kernel on the bandwidth ϵ.

Semi-Supervised Learning. In a semi-supervised learning (SSL) setting, where we have additional unlabeled data

$$\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m} \sim P,$$

we can improve our estimates of $\lambda_{\epsilon,j}$, $\psi_{\epsilon,j}$ and S_ϵ by incorporating both labeled and unlabeled examples into the kernel matrix $\mathbb{A}_\epsilon$. See e.g. Zhou and Srebro (2011) for SSL with Laplacian eigenmaps in the limit of infinite unlabeled data, i.e., when $m \to \infty$. This can also be done for the estimator based on the Kernel PCA operator.

Although in this chapter we focus on the estimator of Equation 3.2, i.e., the estimator based on the diffusion basis, much of the discussion also applies to the estimator based on the kernel PCA basis, Eq. 3.1. Moreover, in the experiments we implement both methods.

3.2.3 *Loss Function and Tuning of Parameters*

To measure the performance of an estimator $\widehat{r}(\mathbf{x})$, we consider the L^2 loss function

$$L(r, \widehat{r}) = \int_{\mathcal{X}} (r(\mathbf{x}) - \widehat{r}(\mathbf{x}))^2 \, dP(\mathbf{x}). \tag{3.4}$$

We split the data into training, validation and test sets. For each choice of ϵ and a sufficiently large constant $J_{\max}$, we use the training set to estimate the eigenvectors $\psi_{\epsilon,1}, \ldots, \psi_{\epsilon,J_{\max}}$ and the expansion coefficients $\beta_{\epsilon,0}, \ldots, \beta_{\epsilon,J_{\max}}$. We then use the validation set $(\mathbf{x}'_1, z'_1), \ldots, (\mathbf{x}'_{n'}, z'_{n'})$ to minimize the estimated loss

$$\widehat{L}(r, \widehat{r}) = \frac{1}{n'} \sum_{i=1}^{n'} \left(z'_i - \widehat{r}(\mathbf{x}'_i)\right)^2 = \frac{1}{n'} \sum_{i=1}^{n'} \left(z'_i - \sum_{j=0}^{J} \widehat{\beta}_{\epsilon,j} \widehat{\psi}_{\epsilon,j}(\mathbf{x}'_i) \right)^2$$

for different values of J. The last computation is very fast. Due to orthogonality, we need not recompute $\widehat{\beta}_{\epsilon,j}$ and $\widehat{\psi}_{\epsilon,j}$ for $J \leqslant J_{\max}$. We choose the model with the lowest estimated loss on the validation set.

3.3 Connection to Other Work

3.3.1 *Linear Regression and Weighted Least Squares*

Consider the data transformation $\mathbf{y} = \Psi(\mathbf{x})$, where $\Psi = (\psi_1, \ldots, \psi_J)$ are the first J eigenvectors of the diffusion operator A_ϵ. Our series model can be viewed as a (weighted) linear regression in the *transformed* data $(Z_1, \mathbf{y}_1), \ldots, (Z_n, \mathbf{y}_n)$. By increasing J, the dimension of the feature space, we achieve more flexible (non-parametric) representations. Decreasing J adds more structure to the regression as dictated by the eigenstructure of the data.

Equation 3.3 can be viewed as a weighted least squares (WLS) solution to the linear regression of Z in $\mathbf{y}$. Define the $n \times (J+1)$ matrix of predictors,

$$\mathbb{Y} = \begin{pmatrix} 1 & \psi_1(\mathbf{x}_1) & \cdots & \psi_J(\mathbf{x}_1) \\ 1 & \psi_1(\mathbf{x}_2) & \cdots & \psi_J(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \psi_1(\mathbf{x}_n) & \cdots & \psi_J(\mathbf{x}_n) \end{pmatrix}, \tag{3.5}$$

and the weight matrix

$$\mathbb{W} = \begin{pmatrix} s(\mathbf{x}_1) & 0 & \cdots & 0 \\ 0 & s(\mathbf{x}_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s(\mathbf{x}_n) \end{pmatrix}, \tag{3.6}$$

where Ψ_j and s are estimated from data (Equations 2.7 and 2.9). Let $Z = (Z_1, \ldots, Z_n)^T$, $e = (\epsilon_1, \ldots, \epsilon_n)^T$ and $\beta = (\beta_1, \ldots, \beta_J)^T$. The basis model assumes that $Z = \mathbb{Y}\beta + e$. By minimizing the weighted residual sum of squares

$$\mathrm{RSS}(\beta) = (Z - \mathbb{Y}\beta)^T \mathbb{W} (Y - \mathbb{Y}\beta), \tag{3.7}$$

which puts more weight on observations in high-density regions, we arrive at the WLS estimator

$$\widehat{\beta} = (\mathbb{Y}^T \mathbb{W} \mathbb{Y})^{-1} (\mathbb{Y}^T \mathbb{W} Z) = \frac{1}{n} \mathbb{Y}^T \mathbb{W} Z. \tag{3.8}$$

This expression is equivalent to Equation 3.3. Because of the orthogonality property $\mathbb{Y}^T\mathbb{W}\mathbb{Y} = n\mathbb{I}$, model search and model selection are feasible even for complex models with very large J. This is in clear contrast with standard multiple regression where the $\widehat{\beta}_j$ estimates need to be recomputed for each model, and the inputs (columns of the design matrix $\mathbb{Y}$) may be linearly dependent.

Remarks:

1. It is straightforward to incorporate heteroscedastic errors into the above framework. For a regression model $Z_i = r(\mathbf{x}_i) + \sigma(\mathbf{x}_i)\epsilon_i$, where ϵ_i are iid realizations of a random variable ϵ with zero mean and unit variance, and $\sigma(\mathbf{x})$ is a non-negative function, simply perform an orthogonal series expansion of the rescaled regression function $g(\mathbf{x}) = r(\mathbf{x})/\sigma(\mathbf{x})$.

2. If one uses the Kernel PCA operator (Eq. 2.2) to derive the basis, then performing an *unweighted* linear regression of Z on the new coordinates $\mathbf{y}$ is equivalent to computing the spectral series estimator of Eq. 3.1.

3.3.2 *Kernel Machine Learning and Regularization in RKHS*

In kernel machine learning (Schölkopf et al. 1997; Cucker and Zhou 2007), it is common to consider the variational problem

$$\min_{r \in \mathcal{H}_K} \left[\frac{1}{n} \sum_{i=1}^{n} L(z_i, r(\mathbf{x}_i)) + \gamma \|r\|^2_{\mathcal{H}_K} \right], \tag{3.9}$$

where $L(z_i, r(\mathbf{x}_i))$ is a convex loss function, $\gamma > 0$ is a penalty parameter, and $\mathcal{H}_K$ is the Reproducing Kernel Hilbert Space (RKHS) associated with a symmetric positive semi-definite kernel K. [2] Penalizing the RKHS norm $\|\cdot\|_{\mathcal{H}_K}$ imposes smoothness conditions on possible solutions. Now suppose that

$$K(\mathbf{x}, \mathbf{y}) = \sum_{j=0}^{\infty} \lambda_j \phi_j(\mathbf{x}) \phi_j(\mathbf{y}),$$

2 To every continuous, symmetric, and positive semi-definite kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is associated a unique RKHS $\mathcal{H}_K$ (Aronszajn 1950). This RKHS is defined to be the closure of the linear span of the set of functions $\{K(\mathbf{x}, \cdot) : \mathbf{x} \in \mathcal{X}\}$ with the inner product satisfying the reproducing property $\langle K(\mathbf{x}, \cdot), r \rangle_{\mathcal{H}_K} = r(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}, r \in \mathcal{H}_K$.

where the RKHS inner product is related to the L^2-inner product according to $\langle \phi_i, \phi_j \rangle_{\mathcal{H}_k} = \frac{1}{\lambda_i} \langle \phi_i, \phi_j \rangle_{L^2(\mathcal{X},P)} = \frac{1}{\lambda_i} \delta_{i,j}$. Eq. 3.9 is then equivalent to considering eigen-expansions

$$r(\mathbf{x}) = \sum_{j=0}^{\infty} \beta_j \phi_j(\mathbf{x}),$$

and seeking solutions to $\min_{r \in \mathcal{B}_r} \frac{1}{n} \sum_{i=1}^{n} L(z_i, r(\mathbf{x}_i))$, where the hypothesis space

$$B_r = \{r \in \mathcal{H}_K : \|r\|_{\mathcal{H}_k} \leqslant r\} \tag{3.10}$$

is a ball of the RKHS $\mathcal{H}_k$ with radius r, and the RKHS norm is given by $\|r\|_{\mathcal{H}_k} = \left(\sum_{j=0}^{\infty} \frac{\beta_j^2}{\lambda_j} \right)^{1/2}$.

The above setting is similar to ours, but there are algorithmic differences, as well as differences in the interpretation of the regression estimator. For Support Vector Machines (Steinwart and Christmann 2008) and other kernel-based regularization methods (such as splines, ridge regression, and radial basis functions), eigen-expansions are never explicitly computed. Instead, these methods rely on the classical Representer Theorem (Wahba 1990) which states that the solution to Eq. 3.9 is a finite expansion of the form $r(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x)$. The functions $k(x_i, \cdot)$ themselves are often referred to as basis functions. For example, for a Gaussian kernel, these are the so-called Gaussian *radial basis functions*. Because of the finite expansion, the original infinite-dimensional variational problem is reduced to a finite-dimensional optimization of the coefficients α_i. These coefficients have to be recomputed for each choice of the penalty parameter γ. This can make cross-validation approaches cumbersome. In our spectral series approach, we take advantage of the orthogonality of the basis for fast model selection and computation of the β_j parameters. As in spectral clustering methods, we explicitly compute the eigenvectors of the kernel and use them to analyze the data.

Finally, another difference is that while we use a projection (i.e., a basis subset selection) method to choose which eigenfunctions to use, the regularization in Eq. 3.9 differentially shrinks contributions from higher-order terms with small λ_j values.

3.4 Theory

Here we investigate how the performance of a spectral series estimator derived from the diffusion operator depends on the choice of the smoothing parameters J and ϵ. We also address questions such as when the basis representation $\widehat{r}(\mathbf{x})$ in Eq. 3.2 is sparse, and whether our method adapts to the intrinsic dimensionality of the data.

Consider the loss function in Equation 3.4. Using the same notation as before, let

$$r(\mathbf{x}) = \sum_{j=0}^{\infty} \beta_{\epsilon,j}\psi_{\epsilon,j}(\mathbf{x}), \qquad r_{\epsilon,J}(\mathbf{x}) = \textstyle\sum_{j=0}^{J} \beta_{\epsilon,j}\psi_{\epsilon,j}(\mathbf{x}),$$
$$\widehat{r}_{\epsilon,J}(\mathbf{x}) = \textstyle\sum_{j=0}^{J} \widehat{\beta}_{\epsilon,j}\widehat{\psi}_{\epsilon,j}(\mathbf{x}),$$

We write

$$|r(\mathbf{x}) - \widehat{r}_{\epsilon,J}(\mathbf{x})|^2 \leqslant 2|r(\mathbf{x}) - r_{\epsilon,J}(\mathbf{x})|^2 + 2|r_{\epsilon,J}(\mathbf{x}) - \widehat{r}_{\epsilon,J}(\mathbf{x})|^2,$$

and refer to the two terms as "bias" and "variance". Hence, define

$$L_{\text{bias}} = \int_{\mathcal{X}} |r(\mathbf{x}) - r_{\epsilon,J}(\mathbf{x})|^2 dP(\mathbf{x}),$$

and

$$L_{\text{var}} = \int_{\mathcal{X}} |r_{\epsilon,J}(\mathbf{x}) - \widehat{r}_{\epsilon,J}(\mathbf{x})|^2 dP(\mathbf{x}).$$

In what follows, we bound the two components under the following assumptions:

(A1) P has compact support $\mathcal{X}$ and bounded density $0 < a \leqslant p(\mathbf{x}) \leqslant b < \infty$, $\forall x \in \mathcal{X}$.

(A2) The weights are positive and bounded; that is, $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$,

$$0 < m \leqslant K_\epsilon(\mathbf{x}, \mathbf{y}) \leqslant M,$$

where m and M are constants that do not depend on ϵ.

(A3) The psd operator A_ϵ has nondegenerate eigenvalues; i.e.,

$$1 \equiv \lambda_{\epsilon,0} > \lambda_{\epsilon,1} > \lambda_{\epsilon,2} > \ldots \lambda_{\epsilon,J} > 0.$$

(A4) $\forall 0 \leqslant j \leqslant J$ and $X \sim P$, there $\exists$ some constant $C < \infty$ (not depending on n) such that

$$\mathbb{E}\left[|\widehat{\varphi}_{\epsilon,j}(\mathbf{X}) - \varphi_{\epsilon,j}(\mathbf{X})|^2\right] < C,$$

where $\varphi_{\epsilon,j}(\mathbf{x}) = \psi_{\epsilon,j}(\mathbf{x})s_\epsilon(\mathbf{x})$ and $\widehat{\varphi}_{\epsilon,j}(\mathbf{x}) = \widehat{\psi}_{\epsilon,j}(\mathbf{x})\widehat{s}_\epsilon(\mathbf{x})$.

Without loss of generality, we assume that the eigenfunctions $\psi_{\epsilon,j}$ are estimated using an unlabeled sample $\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_n$ that is drawn independently from the data used to estimate the coefficients $\beta_{\epsilon,j}$. This is to simplify the proofs and can always be achieved by splitting the data in two sets.

3.4.1 *Bias*

The approximation error of the regression depends on the smoothness of r relative to P. To measure the smoothness of a given function $g(\mathbf{x})$ (not necessarily the regression function), we first define the operator

$$G_\epsilon = \frac{A_\epsilon - I}{\epsilon}, \tag{3.11}$$

where I is the identity. The operator G_ϵ has the same eigenvectors $\psi_{\epsilon,j}$ as the differential operator A_ϵ. Its eigenvalues are given by $-\nu_{\epsilon,j}^2 = \frac{\lambda_{\epsilon,j}-1}{\epsilon}$, where $\lambda_{\epsilon,j}$ are the eigenvalues of A_ϵ. Define the functional

$$\mathcal{J}_\epsilon(g) = -\langle G_\epsilon g, g\rangle_\epsilon \tag{3.12}$$

which maps a function $g \in L^2(\mathcal{X}, P)$ into a non-negative real number. For small ϵ, $\mathcal{J}_\epsilon(g)$ measures the variability of the function g with respect to the distribution P. The expression is a variation of the graph Laplacian regularizers popular in semi-supervised learning (Zhu et al. 2003). In fact, a Taylor expansion yields $G_\epsilon g = -\triangle g + \frac{\nabla p}{p} \cdot \nabla g + O(\epsilon)$ where ∇ is the gradient operator and $\triangle = -\sum_{j=1}^{d} \frac{\partial^2}{\partial x_j^2}$ is the psd Laplace operator in $\mathbb{R}^d$. In kernel regression smoothing, the extra term $\frac{\nabla p}{p} \cdot \nabla g$ is considered an undesirable extra bias, called design bias. In classical regression, it is removed by using local linear smoothing (Fan 1993), which is asymptotically equivalent to replacing the original kernel $K_\epsilon(\mathbf{x}, \mathbf{y})$ by the bias-corrected kernel $K_\epsilon^*(\mathbf{x}, \mathbf{y}) = \frac{K_\epsilon(\mathbf{x},\mathbf{y})}{p_\epsilon(\mathbf{x})p_\epsilon(\mathbf{y})}$ (Coifman and Lafon 2006).

The following result bounds the approximation error for an orthogonal series expansion of g. The bound is consistent with Theorem 2 in Zhou and Srebro (2011), which applies to the more restrictive setting of SSL with *infinite* unlabeled data and $\epsilon \to 0$. Our result holds for all ϵ and J and does not assume unlimited data.

Proposition 3.1. *For* $g \in L^2(\mathcal{X}, P)$,

$$\int_{\mathcal{X}} |g(x) - g_{\epsilon,J}(x)|^2 dS_\epsilon(x) \leqslant \frac{\mathcal{J}_\epsilon(g)}{\nu^2_{\epsilon,J+1}} \tag{3.13}$$
$$L_{\text{bias}} = O\left(\frac{\mathcal{J}_\epsilon(g)}{\nu^2_{\epsilon,J+1}}\right),$$

where $-\nu^2_{\epsilon,J+1}$ *is the* $(J+1)^{\text{th}}$ *eigenvalue of* G_ϵ*, and* $g_{\epsilon,J}$ *is the projection of* g *onto the* J *first elements of the eigenbasis.*

Smoothness and Sparsity. In the limit $\epsilon \to 0$, we have several interesting results, including a generalization of the classical connection between Sobolev differentiability and the error decay of Fourier approximations (Mallat 2009, Section 9.1.2) to a setting with adaptive bases and high-dimensional data. We denote the quantities derived from the bias-corrected kernel K^*_ϵ by A^*_ϵ, G^*_ϵ, $\mathcal{J}^*_\epsilon$ and so forth.

Definition 3.1. (Smoothness relative to P) *A function* g *is smooth relative to* P *if*

$$\int_{\mathcal{X}} \|\nabla g(x)\|^2 dS(x) \leqslant c^2 < \infty,$$

where $S(A) = \frac{\int_A p(x) dP(x)}{\int p(x) dP(x)}$ *is the stationary distribution of the diffusion operator as* $\epsilon \to 0$*. The smaller the value of* c*, the smoother the function.*

The following Lemma shows how smoothness relative to P relates to Proposition 3.1.

Lemma 3.1. *For functions* $g \in C^3(\mathcal{X})$ *whose gradients vanish at the boundary,*

$$\lim_{\epsilon \to 0} \mathcal{J}^*_\epsilon(g) = \int_{\mathcal{X}} \|\nabla g(x)\|^2 dS(x).$$

This result is similar to the convergence of the (un-normalized) graph Laplacian regularizer to the density-dependent smoothness functional $\int_{\mathcal{X}} \|\nabla g(x)\|^2 p^2(x) dx$ (Bousquet et al. 2003).

In what follows, we show that *smoothness* relative to P (Definition 3.1) and *sparsity* in the eigenbasis of the diffusion operator (Definition 3.2 below) are really the same thing. Furthermore, we link smoothness and sparsity to the rate of the error decay of the eigenbasis approximation.

Definition 3.2. (Sparsity in $\mathcal{B}$) *A set of real numbers $\theta_1, \theta_2, \ldots$ lies in a Sobolev ellipsoid $\Theta(s, c)$ if $\sum_{j=1}^{\infty} j^{2s}\theta_{(j)}^2 \leqslant c^2$ for some number $s > 1/2$. For a given basis $\mathcal{B} = \{\psi_1, \psi_2, \ldots\}$, let*

$$W_{\mathcal{B}}(s, c) = \left\{ g = \sum_j \beta_j \psi_j : \beta_1, \beta_2, \ldots \in \Theta(s, c) \right\}$$

where $s > 1/2$. Functions in $W_{\mathcal{B}}(s, c)$ are sparse in $\mathcal{B}$. The larger the value of s, the sparser the representation.

Theorem 3.1. *Assume that $\mathcal{B} = \{\psi_1, \psi_2, \ldots\}$ are the eigenfunctions of $\triangle$ with eigenvalues $\nu_j^2 = O(j^{2s})$ for some $s > 1/2$[3]. Let $g_J(x) = \sum_{j \leqslant J} \beta_j \psi_j(x)$. Then, the following two statements are equivalent:*

1. $\int_{\mathcal{X}} \|\nabla g(x)\|^2 dS(x) \leqslant c^2$ *(smoothness relative to P)*
2. $g \in W_{\mathcal{B}}(s, c)$ *(sparsity in $\mathcal{B}$).*

Furthermore, sparsity in $\mathcal{B}$ (or smoothness relative to P) implies

$$\int_{\mathcal{X}} |g(x) - g_J(x)|^2 dS(x) = o\left(\frac{1}{J^{2s}}\right).$$

The rate s of the error decay depends on the dimension of the data. We will address this issue in Section 3.4.3.

3.4.2 *Variance*

In Appendix A we show the following bound on the variance term.

3 Notice that by Proposition 3 from Coifman and Lafon (2006), $\psi_{\epsilon,j} \overset{\epsilon}{\longrightarrow} \psi_j$

Proposition 3.2. *For* $r \in L^2(\mathcal{X}, P)$, $\epsilon_n \to 0$, *and* $n\epsilon_n^{d/2}/\log(1/\epsilon_n) \to \infty$, *it holds under Assumptions (A1)-(A4) and regularity conditions (see Appendix A) that*

$$L_{\text{var}} = J\left(O_P\left(\frac{1}{n}\right) + O_P\left(\frac{\gamma_n^2}{\Delta_{\epsilon,J}^2}\right)\right).$$

where $\Delta_{\epsilon,J} = \min_{0\leqslant j\leqslant J}(\lambda_{\epsilon,j} - \lambda_{\epsilon,j+1})$, *and* $\gamma_n = \sqrt{\frac{\log(1/\epsilon_n)}{n\epsilon_n^{d/2}}}$.

3.4.3 *Total Loss*

From Propositions 3.1 and 3.2, we have the following result:

Theorem 3.2. *Let* $\epsilon_n \to 0$ *and* $n\epsilon_n^{d/2}/\log(1/\epsilon_n) \to \infty$. *Then, for* $r \in L^2(\mathcal{X}, P)$,

$$L(r, \widehat{r}) = O\left(\frac{\mathcal{J}_\epsilon(r)}{\nu_{\epsilon,J+1}^2}\right) + JO_P\left(\frac{1}{n}\right) + JO_P\left(\frac{\gamma_n^2}{\Delta_{\epsilon,J}^2}\right), \tag{3.14}$$

where $\mathcal{J}_\epsilon(r) = -\langle G_\epsilon r, r\rangle_\epsilon$, $\nu_{\epsilon,J+1}^2$ *is the* $(J+1)^{\text{th}}$ *eigenvalue of* $-G_\epsilon$, $\gamma_n = \sqrt{\frac{\log(1/\epsilon_n)}{n\epsilon_n^{d/2}}}$, *and* $\Delta_{\epsilon,J} = \min_{0\leqslant j\leqslant J}(\lambda_{\epsilon,j} - \lambda_{\epsilon,j+1})$.

Corollary 3.1. *Assume that* $r \in C_b^3(\mathcal{X})$ *and that the kernel* K_ϵ^* *is corrected for bias. Then, for* $\epsilon_n \to 0$ *and* $n\epsilon_n^{d/2}/\log(1/\epsilon_n) \to \infty$,

$$L(r, \widehat{r}) = \frac{\mathcal{J}(r)O(1) + O(\epsilon_n)}{\nu_{J+1}^2} + JO_P\left(\frac{1}{n}\right) + JO_P\left(\frac{\gamma_n^2}{\epsilon_n\Delta_J^2}\right), \tag{3.15}$$

where ν_{J+1}^2 *is the* $(J+1)^{\text{th}}$ *eigenvalue of* $\triangle$, $\mathcal{J}(r) = \int_{\mathcal{X}} \|\nabla r(\mathbf{x})\|^2 dS(\mathbf{x})$, *and* $\Delta_J = \min_{0\leqslant j\leqslant J}(\nu_{j+1}^2 - \nu_j^2)$.

Some remarks on the interpretation of these results: The first term in Equation 3.14 corresponds to the approximation error of the estimator and decays with J. The second and third terms correspond to the variance. Note that the variance term $JO_P\left(\frac{1}{n}\right)$ is the same as the variance of a *traditional* orthogonal series estimator in *one* dimension; in d dimensions, the variance term for a traditional tensor product basis is $O_P\left(\frac{1}{n}\right)\prod_{i=1}^d J_i$ where J_i is the number of components in the ith direction (Efromovich 1999). By estimating the basis in our spectral series method, we incur an additional variance term $JO_P\left(\frac{\gamma_n^2}{\epsilon_n\Delta_J^2}\right)$. [4]

4 In a SSL setting, this extra estimation error vanishes in the limit of infinite unlabeled data.

Assume r is smooth with respect to P. If we balance the two ϵ-terms in Equation 3.14, we get a bandwidth of $\epsilon_n \asymp (1/n)^{2/(d+4)}$. With this choice of ϵ_n and by ignoring terms of lower order, the rate becomes

$$L(r, \widehat{r}) = O\left(\frac{\mathcal{J}(r)}{\nu_{J+1}^2}\right) + \frac{J}{\Delta_J^2} O_P\left(\frac{\log n}{n}\right)^{\frac{2}{d+4}}. \tag{3.16}$$

Corollary 3.2. *Suppose the support of the data is on a compact C^∞ submanifold of $\mathbb{R}^d$ with intrinsic dimension p. Under the assumptions of Theorem 3.2 and Corollary 3.1, and assuming r is smooth with respect to P (recall Definition 3.1), we obtain the rate*

$$L(r, \widehat{r}) = O\left(\frac{1}{J^{2/p}}\right) + J^{2(1-\frac{1}{p})} O_P\left(\frac{\log n}{n}\right)^{\frac{2}{p+4}}.$$

It is then optimal to take $J \asymp (n/\log n)^{\frac{1}{p+4}}$, in which case the upper bound becomes

$$\left(\frac{\log n}{n}\right)^{\frac{2}{(p+4)p}}.$$

We make the following observations:

1. **Adaptiveness to Low-Dimensional Structure.** If the data in $\mathbb{R}^d$ has intrinsic dimension $p \ll d$, then the rate $n^{-1/O(p^2)}$ above is a significant improvement of the minimax rate $n^{-1/O(d)}$ for a nonparametric regressor in $\mathbb{R}^d$.

2. **Minimax Optimality.** In a semi-supervised learning setting, the estimation error of the basis vanishes in the limit of infinite unlabeled data. The loss then reduces to

$$L(r, \widehat{r}) = O\left(\frac{1}{J^{2/p}}\right) + J O_P\left(\frac{1}{n}\right), \tag{3.17}$$

which is minimized by taking $J \asymp n^{p/(p+2)}$. At the minimum, we achieve the rate

$$n^{-\frac{2}{2+p}},$$

the minimax rate for a nonparametric estimator of Sobolev smoothness $\beta = 1$ in $\mathbb{R}^D$, where $D = p$. The latter result is also, up to a logarithmic term, in agreement with Zhou and Srebro (2011).

Finally, we note that although for simplicity we did not present bounds for the regression estimator based on the kernel PCA operator, the rates we compute in the next chapters for the problem of estimating other unknown functions can be easily adapted to the regression case.

3.5 Applications

We illustrate the spectral series method on three prediction problems that involve high-dimensional data and predictors with complex dependencies. We investigate how the performance depends on the choice of the kernel. We also provide a comparison with classical kernel smoothing, regularization in RKHS, and some state-of-the-art geometric methods for regression on manifolds. In all examples, we tune parameters using the methodology from Section 3.2.3. After having chosen a final model, we use the test set to estimate its prediction error on new data. We also use this set to estimate the variability of the loss. We estimate the standard error of $\widehat{L}$ to be $s/\sqrt{n}$, where s^2 is the empirical variance of $(z_i - \widehat{r}(\mathbf{x}_i))^2$ for the test data.

3.5.1 *Estimating Pose Using Images of Faces*

The first data set contains images of artificial faces from the Isomap database[5] used in Tenenbaum et al. (2000). There are a total of 698 64×64 gray-scale images rendered with different orientation and lighting directions. We are interested in estimating the horizontal left-right pose of each face given an image. Figure 3.1 shows an embedding of the data using the first two non-trivial eigenvectors of the Gaussian diffusion kernel. We have included some sample images to illustrate that the eigenvectors capture the variations in pose fairly well although this information was not taken into account in the construction of the basis.

5 www.isomap.stanford.edu/datasets.html

We implement several regression estimators: As a baseline, we include the classical Nadaraya-Watson estimator (*NW*) with a Gaussian smoothing kernel, as well as a nearest neighbors regression estimator (*NN*), known to automatically adapt to the intrinsic dimension of the data if it lives on a submanifold of the original space (Kpotufe 2011). For the spectral series method (*series*), we implement the Gaussian kernel (*radial*) (via diffusion and kernel PCA basis) and polynomial kernels of degrees 1, 2 and 3 (*poly, poly2, poly3*). Note that the series approach with a first-order polynomial is equivalent to a linear regression on eigenvectors computed with PCA. We also implement the RKHS method in Section 3.3.2 with a squared-error loss and different kernels (*radial, poly1, poly2, poly3*). For a squared-error loss, Equation 3.9 reduces to a infinite-dimensional, generalized ridge regression problem (Hastie et al. 2001, Section 5.8.2); hence, we use the term kernel ridge regression (*KRR*).

Furthermore, using the implementation[6] by Aswani et al., we provide additional comparison to several local regression and manifold regression methods: *locOLS* is a local ordinary least squares, *locRR* is a local ridge regression, *locEN* is a local elastic net, *locPLS* is a local partial least squares, *locPCR* is a local principal components regression, *NEDE* is the nonparametric exterior derivative estimator, *NALEDE* is the nonparametric adaptive lasso exterior derivative estimator, *NEDEP* is the nonparametric exterior derivative estimator for the "large p, small n" case, and *NALEDEP* is the nonparametric adaptive lasso exterior derivative estimator for the "large p, small n" case. The last 4 regression estimators (*NEDE, NALEDE, NEDEP, NALEDP*) pose the regression as a least-squares problem with a term that penalizes for the regression vector lying in directions perpendicular to an estimated manifold; see Aswani et al. (2011) for details. We also compute *MALLER* (Cheng and Wu 2013), a local polynomial regression estimator designed to work under a manifold assumption. We use the code provided by the authors[7]. Finally,

6 www.eecs.berkeley.edu/~aaswani/EDE_Code.zip

7 www.math.princeton.edu/~hauwu/regression.zip

we compute *LPR*, a standard local polynomial regression, which is known to automatically adapt to the intrinsic dimension of the manifold (Bickel and Li 2007).

The locally linear and manifold regression methods are computationally intensive in high dimensions. Hence, in Aswani et al. (2011), the authors first rescale the images from 64×64 to 7×7 pixels in size, which reduces the number of predictors from $d = 4096$ to $d = 49$. The covariates are normalized to have mean 0 and standard deviation 1. We use 50% of the data for training, 25% for validation and 25% for testing. Results of the regression are shown in Table 3.1. The approaches that have best performance are *series* and *KRR* when using the Gaussian kernel. Notice both the kernel PCA operator and the diffusion operator yields very similar risks. The first-order polynomial kernel, i.e., a global principal component regression, leads to worse performance than *NW*. Higher-order polynomial kernels, the locally linear estimators and the manifold regression estimators (in particular, *NEDE*) improve the *NW* result but *series-radial* and *KRR-radial* are still the best choices in terms of computational as well as statistical performance.

3.5.2 *Estimating Redshift Using SDSS Galaxy Spectra*

Next we apply the formalism developed here to the problem of predicting redshift using spectra from the Sloan Digital Sky Survey (SDSS). Redshift is a quantity related to how fast an object moves away from the observer. It plays a key role in astronomy in determining the distances and ages of objects in the Universe, see more details in Chapter 7. When given spectroscopic data, astronomers can typically estimate redshift with great precision using template fitting and cross-correlation techniques. As we lack knowledge of the true redshift, we use SDSS estimates of spectroscopic redshift (z_{SDSS}) to train and test our estimators.

Our initial data sample consists of spectra that are classified as galaxies from ten arbitrarily chosen spectroscopic plates of SDSS DR6[8]. We preprocess and remove spectra according to the three cuts described in Richards et al. (2009). The final

8 http://www.sdss.org/dr6/algorithms/redshift_type.html

sample consists of 2812 high-resolution galaxy spectra. The predictors are flux measurements at 3501 different wavelengths, and the response is the SDSS redshift.

Table 3.1: Estimated loss for Isomap face data.

Method	Loss (SE)
NW	1.71 (0.23)
NN	1.74 (0.21)
series-poly1	2.96 (0.40)
series-poly2	0.22 (0.04)
series-poly3	0.80 (0.22)
seriesDiff-radial	**0.16 (0.04)**
series-radial	**0.16 (0.03)**
KRR-poly1	2.95 (0.41)
KRR-poly2	0.19 (0.03)
KRR-poly3	0.59 (0.14)
KRR-radial	**0.15 (0.04)**

Method	Loss (SE)
OLS	0.65 (0.17)
RR	0.46 (0.16)
EN	0.47 (0.16)
PLS	0.65 (0.21)
PCR	0.95 (0.20)
NEDE	0.44 (0.14)
NALEDE	0.46 (0.14)
NEDEP	0.81 (0.31)
NALEDEP	0.85 (0.33)
MALLER	0.24 (0.06)
LPR	0.37 (0.06)

For the regression task, we implement our spectral series method and Kernel Ridge Regression with the Gaussian kernel, as well as first, second- and third-order polynomial kernels. Because of the large number of variables ($d = 3501$), we were not able to implement the computationally more intensive locally linear and manifold regression estimators from Aswani et al. (2011), nor the local polynomial regression. We use 50% of the data for training, 25% for validation and 25% for testing. The results are summarized in Table 3.2. Figure 3.2 shows an embedding of the SDSS galaxy spectra using the first three non-trivial eigenvectors of the Gaussian diffusion kernel. The color codes for redshift.

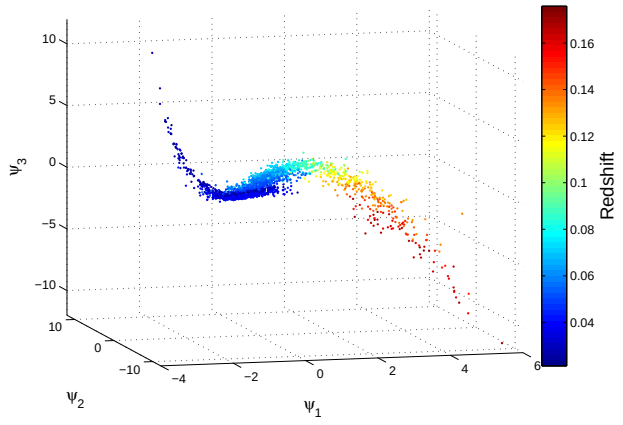


Figure 3.2: Embedding of a sample of SDSS galaxy spectra using the first three non-trivial eigenvectors of the Gaussian diffusion kernel. The color codes for redshift.

As before, the radial kernel yields the best performance and, as expected, the series and the kernel ridge regression estimators are essentially equivalent in terms of performance. Unlike the Isomap face application, a dimensionality reduction with PCA (*series-poly1*) improves upon the *NW* regression results. Using higher-order polynomials (e.g., *series-poly2*), however, does not improve the results further. The reason is the high dimension of the problem. In $\mathbb{R}^p$, the kernel $k(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + 1)^q$ has a total of $M = \binom{p+q}{d}$ eigenfunctions that span the space of polynomials of degree q. For $p = 3501$ and $q = 1$, we have 3506 eigenfunctions already. Adding more eigenfunctions seems to only increase the variance of the series estimator, even when chosen nonlinearly according to decreasing values of $|\widehat{\beta}_j|^2$, or when penalized by $|\widehat{\beta}_j|^2/\lambda_j$ as in *KRR*. Finally, we observe that *MALLER* achieves similar statistical performance as series, however its computation time is much larger: *Series* takes less than one minute on a 2.70GHz Intel Core i7-4800MQ CPU, but *MALLER* takes ~1 hour, one of the reasons being that the estimator has to be recomputed for each testing point.

Table 3.2: Estimated loss for redshift prediction using SDSS galaxy spectra.

Method	Loss (SE) $\times 10^{-5}$
NW	11.73 (3.23)
NN	10.18 (2.01)
series-poly1	6.99 (0.65)
series-poly2	6.99 (0.66)
series-poly3	6.99 (0.66)
seriesDiff-radial	**3.97 (0.51)**
series-radial	**4.26 (0.50)**
KRR-poly 1	6.61 (0.65)
KRR-poly 2	6.62 (0.64)
KRR-poly 3	6.61 (0.64)
KRR-radial	**4.29 (0.51)**
MALLER	**4.65** (0.8)

3.5.3 *Galaxy Morphology Classification*

Astronomers usually divide galaxies into different morphological groups that capture their structure (Hubble 1926). Some of these include *disks*, *spheroids*, *irregulars* and *mergers*, see Figure 3.3 for some examples. An important question in modern astronomy is how to automatically classify new galaxies into the several different morphological groups. This is typically done using a labeled sample of human-classified galaxies; see, e.g., Peng et al. (2002), Conselice (2003), Lotz et al. (2004), and Gauci et al. (2010). More specifically, it is traditional to perform automatic classification by extracting a small number of meaningful features from the images. Several features have been proposed, see, e.g., Peng et al. (2002), Conselice (2003), and Lotz et al. (2004). In particular, in Freeman et al. (2013), we develop

statistics designed to detect *non-regular* galaxies, i.e., galaxies that are either irregulars or mergers.

(a) Spheroid (b) Disk (c) Irregular (d) Merger

Figure 3.3: Examples of galaxy morphologies. Source: ESA/Hubble.

Here we investigate how the series method performs in classifying galaxies from CANDELS - Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey - data (Windhorst et al. 2011). The goal is to evaluate whether it is possible to create good classifiers without extracting task-specific features from the images. Hence, the response here is the indicator function of the class of the galaxy, and the covariates are the whole 84 by 84 images, which we normalized to be centered, and to have the same major axis length[9]. Moreover, we rotate them so that the major axis is perpendicular to the y-axis.

We compare the series approach (*Series* and *SeriesDiff*) with *PCA*, a least squares regression on the first components of a principal components regression; *NN*, a nearest neighbors regression; and *Features*, a logistic regression based on the features we developed in Freeman et al. (2013), along with traditional features from the literature, such as Gini and M20; see Freeman et al. (2013) for more details. We split data into training, validation and testing (50%, 25% and 25% of the data, respectively). All tuning parameters are chosen so as to minimize the estimated risk based on the validation set. Because most of the classes unbalanced (e.g., only $\approx$ 5% of the galaxies are mergers), rather than using a 0-1 loss, we use 2-

9 This is done by fitting an ellipse to the image via least squares.

(Sensitivity+Specificity), i.e., $1 - \mathbb{P}(\widehat{Z} = 1|Z = 0) + 1 - \mathbb{P}(\widehat{Z} = 0|Z = 1)$, where $\widehat{Z}$ is the estimated response; see Freeman et al. (2013) for additional details.

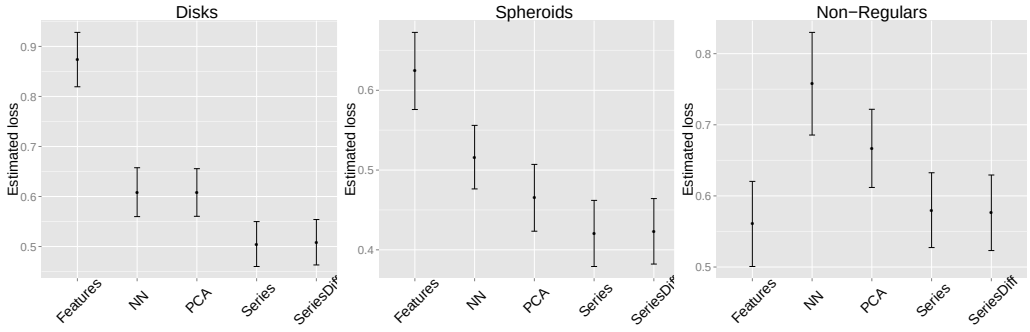


Figure 3.4: Estimated losses of the various methods for predicting galaxy morphology, with standard errors. While *Features* is based on task-specific summary statistics of the images, the other methods work directly with the images.

Figure 3.4 shows the estimated losses for the various methods, along with standard errors. The spectral series methods have better performance than the others when detecting disks and spheroids, and are as good as the task-specific summary statistics for detecting non-regular galaxies. Hence, spectral series allows one to get reasonable classification rates in these problems without going through the process of building task-specific statistics.

3.5.4 *Summary of the Experimental Results*

Our main findings in the experiments were:

- Both normalizations of the kernel operator, *kernel PCA* and *diffusion*, yield spectral series estimators with similar performance.
- Using a Gaussian kernel leads to better estimates than polynomial kernels.
- The spectral series estimator with a Gaussian kernel performed better than most estimators, including those designed to give good results under a mani-

fold assumption, and those known to adapt to manifold structure. Moreover, the series estimator has better computational performance.

- Although the spectral series regression estimators have similar performance to kernel ridge regression, they present additional advantages in terms of interpretation and visualization.

4

CONDITIONAL DENSITY ESTIMATION IN A REGRESSION SETTING

4.1 Introduction

In the last chapter we saw how spectral series can be used to estimate the regression of a random variable $Z \in \mathbb{R}$ given a high-dimensional random vector $\mathbf{X} \in \mathbb{R}^d$, i.e., the conditional mean $\mathbb{E}[Z|\mathbf{x}]$. However, in many modern applications, one benefits more from estimating $f(z|\mathbf{x})$ – the *full* conditional density – rather than only the regression curve. For example, the conditional density function is useful in constructing more accurate predictive intervals for new observations (Fernández-Soto et al. 2001). Estimating $f(z|\mathbf{x})$ is also a simple way of performing nonparametric quantile regression (Takeuchi et al. 2006) of many quantiles simultaneously. Moreover, in, for example, forecasting and prediction in economics (Filipović et al. 2012; Kalda and Siddiqui 2013; Gneiting and Katzfuss 2014), the conditional density itself is often the quantity of interest.

Finally, there are situations where the regression $\mathbb{E}[Z|\mathbf{x}]$ is simply not informative enough to create good predictions of Z, because of *multi-modality*, *asymmetry*, or *heteroscedastic noise* in $f(z|\mathbf{x})$. It is then more useful to construct a nonparametric estimate of $f(z|\mathbf{x})$ to describe the association between a predictor and a response. As $Z \in \mathbb{R}$, Efromovich (2007) refers to this problem as a "conditional density estimation in a regression setting." As a case in point, several recent works in

cosmology (Cunha et al. 2009; Wittman 2009; Sheldon et al. 2012) have shown that using the full probability distribution of photometric redshifts given galaxy colors $\mathbf{x}$, instead of a single-point estimate per galaxy, can significantly reduce systematic errors in cosmological analyses; thereby, improving estimates of the parameters that dictate the structure and evolution of our Universe. In fact, in a review of the current state of data mining and machine learning in astronomy, Ball and Brunner (2010) listed working with full probability density functions as one of the future trends of the field. See Chapter 7 for more details.

In statistics and machine learning, several nonparametric estimators have been proposed to estimate conditional densities when $\mathbf{x}$ lies in a *low-dimensional* space. Many of them are based on first estimating $f(z, \mathbf{x})$ and $f(\mathbf{x})$ with for example kernel density estimators (Rosenblatt 1969), and then combining the estimates according to $f(z|\mathbf{x}) = f(z, \mathbf{x})/f(\mathbf{x})$. Several works further improve upon such an approach by using different criteria and shortcuts to tune parameters (e.g., Hyndman et al. 1996; Bashtannyk and Hyndman 2000; Ichimura and Fukuda 2010). Other approaches to conditional density estimation in low dimensions include using locally polynomial regression (Fan et al. 1996), copulas (Faugeras 2009), nearest neighbors (Zhao and Liu 1985), direct estimation of $f(z|\mathbf{x})$ using a least squares approach (Sugiyama et al. 2010b) and density estimation through quantile estimation (Takeuchi et al. 2009).

Most attempts to estimate $f(z|\mathbf{x})$ when $\mathbf{x}$ has more than about $d = 3$ dimensions rely on a dimension reduction of $\mathbf{x}$ prior to implementation (e.g., Fan et al. 2009). In a different attempt to overcome the curse of dimensionality, Hall et al. (2004) propose a method for tuning parameters in kernel density estimators which automatically determines which components of $\mathbf{x}$ are relevant to $f(z|\mathbf{x})$. The method produces very good results but is not practical for high-dimensional data sets: Because it is based on using a different bandwidth for each covariate, it has a high computational cost that increases with both the sample size n and the dimension d, with prohibitive costs even for moderate n's and d's. A second framework is developed by Efromovich (2010), who proposes an orthogonal series estimator

that automatically performs dimension reduction on $\mathbf{x}$ when several components of this vector are conditionally independent of the response. He shows that this method produces results comparable to those from Hall et al. (2004). The estimator is based on expanding the conditional density into a sum of projections on all possible subspaces of reduced dimension, and uses shrinkage procedures to estimate each projection. Unfortunately, the method requires one to compute $d + 1$ tensor products, which quickly becomes computationally intractable even for as few as 10 covariates.

Here, we demonstrate that spectral series bases allow one to use orthogonal series methods more effectively in high dimensions. Besides the advantages discussed in Chapters 2 and 3, spectral series also allow $f(z|\mathbf{x})$ to be estimated *directly*, avoiding estimators based on ratios of random variables, such as for example $\widehat{f}(z, \mathbf{x})/\widehat{f}(\mathbf{x})$. The latter two-step approach is common in other approaches and is not reliable in practical situations (Kanamori et al. 2012; Chagny 2013), especially because estimating $f(z, \mathbf{x})$ and $f(\mathbf{x})$ are non-trivial tasks in high dimensions. Estimating $f(\mathbf{x})$ can in fact be harder than estimating $f(z|\mathbf{x})$ when $f(\mathbf{x})$ is less smooth than $f(z|\mathbf{x})$; see the discussion in Efromovich (2010). Division by an estimated density also tends to magnify estimation errors, particularly in regions where $f(\mathbf{x})$ is small (Sugiyama et al. 2010b).

In Figure 4.1 we display the two first basis functions, $\psi_1(\mathbf{x})$ and $\psi_2(\mathbf{x})$, for the data set we use in our application in Section 4.4. Notice they capture the structure of the data; they vary smoothly with the response z (redshift). Similar data points are grouped together in the eigenmap. That is, samples with similar covariates are mapped to similar eigenfunction values, see plots in the bottom of that figure. Hence, when $f(z|\mathbf{x})$ is smooth as a function of $\mathbf{x}$, our estimator will yield good results.

This chapter is organized as follows. In Section 4.2 we introduce our estimator. Theoretical guarantees are provided in Section 4.3. In Section 4.4, we illustrate the effectiveness of our proposed method with numerical examples, including an

application to redshift estimation for galaxy data. Although for simplicity we work mainly with the case where Z and $\mathbf{X}$ are continuous variables, we briefly discuss how the methodology can be adapted to other situations.

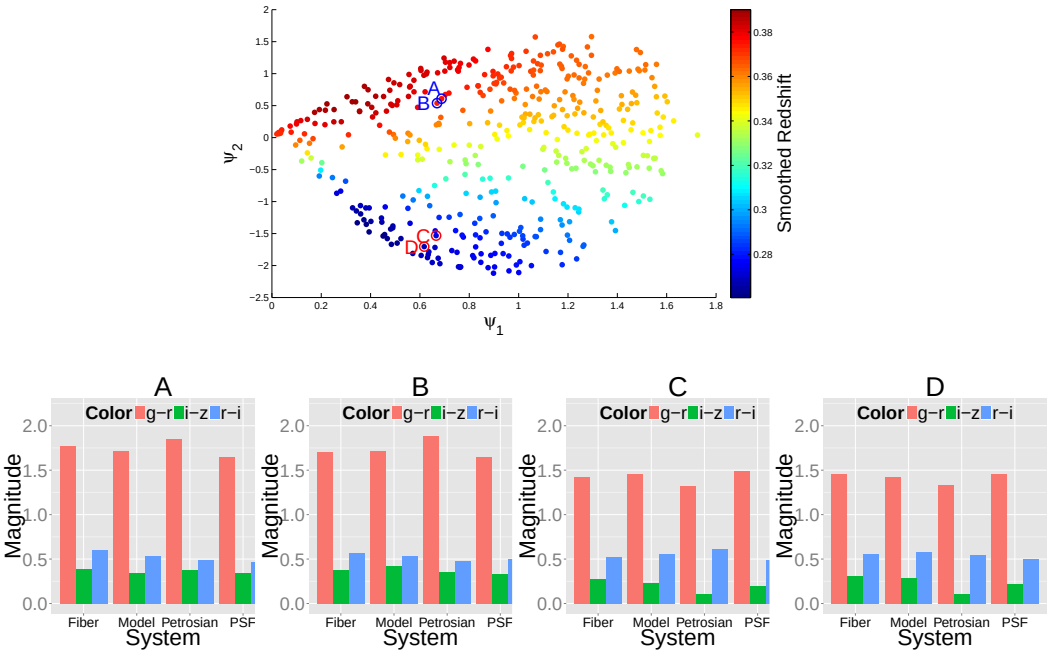


Figure 4.1: Top: Embedding of the red luminous galaxies of SDSS data using the first two eigenvectors of the Gaussian kernel operator. Bottom: Covariates of 4 selected galaxies with their covariates. The two eigenfunctions capture the structure of the data and vary smoothly with the response (redshift).

4.2 METHODOLOGY

Let $(Z_1, \mathbf{X}_1), \ldots, (Z_n, \mathbf{X}_n)$ denote an i.i.d. sample, where $\mathbf{X}_i \in \mathcal{X} \subseteq \mathbb{R}^d$, and the domain of z is bounded; for simplicity, we assume $Z_i \in [0, 1]$. In what follows, we describe the construction of the spectral series estimator of the conditional density $f(z|\mathbf{x})$.

First, using the samples $\mathbf{X}_1, \ldots, \mathbf{X}_n$, we estimate the basis $\{\psi_j(\mathbf{x})\}_j$ just as described in Chapter 2. Recall that these eigenfunctions form an orthonormal basis of $\mathfrak{L}^2(\mathcal{X}, P)$ — the Hilbert space of square integrable functions with domain $\mathcal{X}$ and

norm $\|g\|_P^2 = \langle g, g\rangle_P = \int_{\mathcal{X}} |g(\mathbf{x})|^2 dP(\mathbf{x})$. The conditional density $f(z|\mathbf{x})$, however, is a function of both $\mathbf{x}$ and z. In order to approximate $f(z|\mathbf{x})$, we need an additional basis $(\phi_i)_{i\in\mathbb{N}}$ on $[0,1]$, the domain of z. Because z is a scalar, we can choose $(\phi_i)_{i\in\mathbb{N}}$ to be any orthonormal basis with respect to the Lebesgue measure; that is, any basis such that

$$\int_{[0,1]} \phi_i(z)\phi_j(z)dz = \delta_{i,j}.$$

Whereas $(\psi_j)_j$ is adapted to the distribution of the data, the basis $(\phi_i)_i$ is fixed a priori. It can, for example, be the standard Fourier basis.

The *tensor product* $\left(\Psi(z,\mathbf{x})_{i,j}\right)_{i,j\in\mathbb{N}}$, where

$$\Psi_{i,j}(z,\mathbf{x}) = \phi_i(z)\psi_j(\mathbf{x}),\ i,j \in \mathbb{N}.$$

is then a basis for functions of $\mathbf{x}$ and z. More precisely, it is an orthonormal basis for the space $\mathfrak{L}^2([0,1]\times\mathcal{X}, \lambda\times P)$, where λ is the Lebesgue measure. The central idea of the spectral series estimator is to project $f(z|\mathbf{x})$ onto $\left(\Psi_{i,j}\right)_{i,j}$. The projection is given by

$$f(z|\mathbf{x}) = \sum_{i,j} \beta_{i,j}\Psi_{i,j}(z,\mathbf{x}), \tag{4.1}$$

where

$$\beta_{i,j} = \iint f(z|\mathbf{x})\Psi_{i,j}(z,x)dP(\mathbf{x})dz = \iint \Psi_{i,j}(z,\mathbf{x})dP(\mathbf{x},z) = \mathbb{E}[\Psi_{i,j}(Z,\mathbf{X})]. \tag{4.2}$$

Because ψ is orthogonal with respect to the data distribution, and ϕ is orthogonal with respect to the Lebesgue measure, the expansion coefficients $\beta_{i,j}$ have a simple form in Eq. 4.2: they are simply expectations over the joint distribution of $\mathbf{X}$ and Z.

After estimating the ψ_j's, we estimate the coefficients in Eq. 4.2 by using empirical averages:

$$\widehat{\beta}_{i,j} = \frac{1}{n}\sum_{k=1}^{n} \widehat{\Psi}_{i,j}(z_k,\mathbf{x}_k), \tag{4.3}$$

where

$$\widehat{\Psi}_{i,j}(z_k,\mathbf{x}_k) = \phi_i(z_k)\widehat{\psi}_j(\mathbf{x}_k) \tag{4.4}$$

is the estimate of $\Psi_{i,j}(z_k, \mathbf{x}_k)$.

Our final estimator is given by inserting the estimated basis and coefficients into Equation 4.1:

$$\widehat{f}(z|\mathbf{x}) = \sum_{i=1}^{I} \sum_{j=1}^{J} \widehat{\beta}_{i,j} \widehat{\Psi}_{i,j}(z, \mathbf{x}). \tag{4.5}$$

The parameters I and J control the bias/variance tradeoff: by decreasing their values, we decrease the variance, but increase the bias of the estimator. In Section 4.2.1, we explain how to choose these tuning parameters in a principled way.

Remarks:

1. One can use other bases $(\phi_i(z))_i$ than the Fourier basis to model $f(z|\mathbf{x})$ as a function of z. For example, for spatially inhomogeneous densities (in z), wavelets and related bases may be a good choice. This gives series methods more flexibility as compared to estimators based on kernel smoothers (Efromovich 1999). Moreover, if Z assumes values in a discrete space $\{1, \ldots, p\}$, the spectral series technique can be used to estimate a conditional probability mass function by using as basis the functions $\phi_i(z) = \mathbb{I}(z = i),\ i = 1, \ldots, p$ and defining the inner products with respect to the counting measure, i.e., $\langle f, g \rangle = \sum_{i=1}^{p} f(i)g(i)$.

2. By choosing an appropriate kernel, it is also possible to handle different data types on the covariates. For example, in Lee et al. (2010), the authors suggest a distance kernel that takes into account the discrete nature of genetic SNP data. Similarly, $\mathbf{x}$ can represent functional data, circular data and others; see, e.g., Schölkopf and Smola (2001) for a list of kernels that can be used in different contexts.

3. As in the regression case, it is straightforward to extend this estimator to a semi-supervised learning setting: if additional unlabeled data (that is, data where the covariates $\mathbf{x}$ are known but z is not) are available, they can be used

to estimate the eigenfunctions ψ_j's — as long as the marginal distribution $P(\mathbf{x})$ is the same for both the labeled and unlabeled data.

4.2.1 *Loss Function and Tuning of Parameters*

To measure the accuracy of a given estimator $\widehat{f}(z|\mathbf{x})$, we consider the loss function

$$\begin{aligned} L(\widehat{f}, f) &= \iint \left(\widehat{f}(z|\mathbf{x}) - f(z|\mathbf{x})\right)^2 dP(\mathbf{x})\,dz \\ &= \iint \widehat{f}^2(z|\mathbf{x})\,dP(\mathbf{x})\,dz - 2\iint \widehat{f}(z|\mathbf{x}) f(z,\mathbf{x})\,d\mathbf{x}\,dz + C, \end{aligned} \tag{4.6}$$

where C does not depend on the estimator. This loss is appropriate for many applications: the weighting by P reflects the fact that we are primarily interested in accurately estimating the density at $\mathbf{x}$'s that occur frequently.

To tune the parameters of interest, we split the data into training and validation sets. For each choice of the tuning parameters, we use Equation 4.3 to estimate the coefficients in the training set. We then estimate the loss (4.6) (up to the constant C) using the validation set $(z'_1, \mathbf{x}'_1), \ldots, (z'_{n'}, \mathbf{x}'_{n'})$:

$$\widehat{L}(\widehat{f}, f) = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{m=1}^{J} \widehat{\beta}_{i,j}\widehat{\beta}_{i,m}\widehat{W}_{j,m} - 2\frac{1}{n'}\sum_{k=1}^{n'} \widehat{f}(z'_k|\mathbf{x}'_k), \tag{4.7}$$

where

$$\widehat{W}_{j,m} = \frac{1}{n'}\sum_{k=1}^{n'} \widehat{\psi}_j(\mathbf{x}'_k)\widehat{\psi}_m(\mathbf{x}'_k).$$

It is also possible to use a cross-validation version of this estimator. Orthogonality makes choosing I and J fast because coefficients do not have to be recomputed for different I's and J's.

As in traditional low-dimensional series methods (Efromovich 1999), the final estimate $\widehat{f}$, however, may not be a bona fide density: it can be negative and does not have to integrate to one. We deal with this issue in the next section.

4.2.2 *Normalization and Spurious Bumps*

Here we describe a procedure for transforming the estimate $\widehat{f}$ into a bona fide density $\widetilde{f}$. Several different approaches can be found in the literature (e.g., Glad et al. 2003). Let

$$\widehat{f}_{\max}(z|\mathbf{x}) = \max\left\{0, \widehat{f}(z|\mathbf{x})\right\}.$$

In our experiments, the following procedure gave the best results:

- If $\int \widehat{f}_{\max}(z|\mathbf{x})dz \geqslant 1$, then for each $\mathbf{x}$ and z, define $\widetilde{f}(z|\mathbf{x}) = \max\{0, \widehat{f}(z|\mathbf{x}) - \xi\}$, where ξ is such that $\int \widetilde{f}(z|\mathbf{x})dz = 1$. This approach was proposed by Glad et al. (2003) in the context of unconditional density estimation.

- If $\int \widehat{f}_{\max}(z|\mathbf{x})dz < 1$, then define

$$\widetilde{f}(z|\mathbf{x}) = \frac{\widehat{f}_{\max}(z|\mathbf{x})}{\int \widehat{f}_{\max}(z|\mathbf{x})dz}.$$

  This is a common procedure to create bona fide densities; see (Wasserman 2006).

See Glad et al. (2003) from some theoretical guarantees on these procedures.

Many times, densities estimated through orthogonal series expansions contain small spurious bumps that arise from the approximation of the flat parts of the underlying density (see, e.g., Efromovich 1999 for more detailed explanations). Following Efromovich (1999), we remove these artifacts by choosing a threshold δ and removing bumps with mass smaller than δ, i.e., we remove a bump in the interval $[a, b]$ when $\int_a^b \widetilde{f}(z|\mathbf{x})dz < \delta$. Here we choose the threshold value δ by minimizing the estimated loss (Equation 4.7) based on the validation set. We then renormalize the density so that it integrates to one. In order to decrease computational complexity, we take a greedy approach and choose δ after the other tuning parameters have been chosen. Figure 4.2 shows the estimated conditional density for a fixed $\mathbf{x}$ before and after removing spurious bumps in the example of Section 4.4.1.

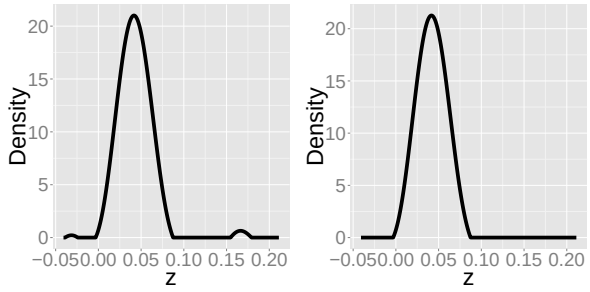


Figure 4.2: Example of estimated conditional density of Section 4.4.3 before (left) and after (right) removing spurious bumps.

Algorithm 2 summarizes our spectral series approach.

Algorithm 2 Spectral Series Conditional Density Estimator

Input: Training data $(z_1, \mathbf{x}_1), \ldots, (z_n, \mathbf{x}_n)$; validation data $(z'_1, \mathbf{x}'_1), \ldots, (z'_{n'}, \mathbf{x}'_{n'})$; grid of ϵ's, I's and J's ▷ ϵ's are the tuning parameters associated to the kernel

Output: Estimator $\widetilde{f}(z|\mathbf{x})$

1: **for all** ϵ **do**
2: calculate the eigenvectors $\widetilde{\psi} = \widetilde{\psi}_\epsilon$ of the Gram matrix ▷ Equation 2.3
3: estimate the eigenbasis $\widehat{\Psi}_{i,j}$ ▷ Equations 2.4 and 4.4
4: estimate the coefficients $\widehat{\beta}_{i,j}$ ▷ Equation 4.3
5: **for all** I, J **do**
6: Calculate the estimated loss $\widehat{L}(\widehat{f}_{\epsilon,I,J}, f)$ ▷ Equation 4.7
7: **end for**
8: **end for**
9: Define $\widehat{f} = \arg\min_{\widehat{f}_{\epsilon,I,J}(z|\mathbf{x})} \widehat{L}(\widehat{f}_{\epsilon,I,J}, f)$
10: Calculate $\widetilde{f}$, the renormalization of $\widehat{f}$ ▷ Section 4.2.2
11: **return** $\widetilde{f}(z|\mathbf{x})$

4.2.3 *Diffusion Kernel*

In order to get bounds similar from those of Chapter 3, we also implement the spectral series estimator based on the *diffusion operator* (see Section 2.2.4).

The construction of both the tensor basis $\{\Psi_{i,j}\}_{i,j}$ and the conditional density estimator proceeds just as described in Section 4.2 when using the operator from Equation (2.2). The only difference is that as $\{\Psi_{i,j}\}_{i,j}$ are now orthonormal with respect to $\lambda \times S_\epsilon$ instead of $\lambda \times P$, the coefficients of the projection are given by

$$\begin{aligned}\beta_{i,j} &= \iint f(z|\mathbf{x})\Psi_{i,j}(z,x)\mathrm{d}S_\epsilon(\mathbf{x})\mathrm{d}z = \iint f(z|\mathbf{x})\Psi_{i,j}(z,x)s_\epsilon(\mathbf{x})\mathrm{d}P(\mathbf{x})\mathrm{d}z \\ &= \iint \Psi_{i,j}(z,\mathbf{x})s_\epsilon(\mathbf{x})\mathrm{d}P(\mathbf{x},z) = \mathbb{E}[\Psi_{i,j}(Z,\mathbf{X})s_\epsilon(\mathbf{X})].\end{aligned}$$

In the next section, we bound how far the estimator $\widehat{f}(z|\mathbf{x})$ is from the true conditional density $f(z|\mathbf{x})$ for both the Kernel PCA and the Diffusion operators.

4.3 THEORY

We now present bounds on the loss (4.6) for the spectral series estimator of Equation 4.5. We make the following assumptions:

Assumption 4.1. $\int f^2(z|\mathbf{x})\mathrm{d}P(\mathbf{x})\mathrm{d}z < \infty$.

Assumption 4.2. $M_\Phi \stackrel{def}{=} \sup_z \sup_i \phi_i(z) < \infty$.

Assumption 4.3. $\lambda_1 > \lambda_2 > \ldots > \lambda_J > 0$.

Assumption 4.1 implies that it is possible to expand f into the basis Ψ. Assumption 4.2 depends on the basis that are used for z and holds, e.g., for cosine or Fourier basis. Assumption 4.3 allows one to uniquely define each of the eigenfunctions (see, e.g., Ji et al. 2012 for similar assumptions, and Zwald and Blanchard 2005 on how to proceed if it does not hold). Denoting by $\mathcal{H}_K$ the Reproducing Kernel Hilbert Space (RKHS) associated to a universal kernel K[1] (as for example the Gaussian radial kernel), the assumption that the eigenvalues are strictly positive hold as long as $P(\mathbf{x})$ is nondegenerate (Cucker and Zhou 2007, pages 51 and 61).

Now, for every $s > \frac{1}{2}$ and $0 < c < \infty$, let $W(s,c)$ denote a Sobolev space (Wasserman 2006). We also assume the following:

1 K is universal if $\mathcal{H}_K$ is dense in the space of continuous functions on $\mathcal{X}$.

Assumption 4.4. (Smoothness in z direction) $\forall \mathbf{x} \in \mathcal{X}$ *fixed*, $f(z|\mathbf{x}) \in W(s_{\mathbf{x}}, c_{\mathbf{x}})$, *where* $f(z|\mathbf{x})$ *is seen as a function of* z, *and* $s_{\mathbf{x}}$ *and* $c_{\mathbf{x}}$ *are such that* $\inf_{\mathbf{x}} s_{\mathbf{x}} \stackrel{def}{=} \beta > \frac{1}{2}$ *and* $\int_{\mathcal{X}} c_{\mathbf{x}}^2 dP(\mathbf{x}) < \infty$.

Assumption 4.4 require $f(z|\mathbf{x})$ to be smooth in the z direction. This is enforced by requiring $f(z|\mathbf{x})$ to be in a Sobolev space for all $\mathbf{x}$'s. The quantities β and $\int_{\mathcal{X}} c_{\mathbf{x}}^2 dP(\mathbf{x})$ are used to link the parameters that control the degrees of smoothness of the different functions of $\mathbf{x}$. Larger β's indicate smoother functions. We also assume $f(z|\mathbf{x})$ is smooth in the $\mathbf{x}$ direction. More precisely, for the bounds on the estimator derived from the operator of Equation (2.2), we assume

Assumption 4.5. (Smoothness in $\mathbf{x}$ direction) $\forall z \in [0,1]$ *fixed*, $f(z|\mathbf{x}) \in \left\{ g \in \mathcal{H}_K : \|g\|_{\mathcal{H}_K}^2 \leqslant c_z^2 \right\}$, *where* $f(z|\mathbf{x})$ *is seen as a function of* $\mathbf{x}$, *and* c_z*'s are such that* $c_K \stackrel{def}{=} \int_{[0,1]} c_z^2 dz < \infty$.

Hence, smoothness on the $\mathbf{x}$ direction it is enforced by requiring $f(z|\mathbf{x})$ to be in a RKHS for all z's. Notice smaller c_K's indicate smoother functions. The reader is referred to, e.g., Minh et al. (2006) for an account of measuring smoothness through norms in RKHSs.

Let n be the sample size of data used to estimate the coefficients $\beta_{i,j}$, and m be the sample size of the data used to estimate the basis functions (as discussed in the remarks of Section 4.2, they don't have to be the same). In Appendix B we prove the following:

Theorem 4.1. *Let* $\widehat{f}_{I,J}(z|\mathbf{x})$ *be the spectral series estimator from Section 4.2 with cutoffs* I *and* J, *based on the eigenfunctions of operator of Equation (2.2). Under Assumptions 4.1-4.5 we have*

$$L(\widehat{f}_{I,J}, f) = IJ \times \left[O_P\left(\frac{1}{n}\right) + O_P\left(\frac{1}{\lambda_J \Delta_J^2 m}\right) \right] + c_K O\left(\lambda_J\right) + O\left(\frac{1}{I^{2\beta}}\right),$$

where $\Delta_J = \min_{1 \leqslant j \leqslant J} \left|\lambda_j - \lambda_{j+1}\right|$.

While the first term of the bound of Theorem 4.1 corresponds to the sampling error of the estimator, the second and third terms correspond to the approximation error, see Appendix B for details.

The bound depends on how smooth the functions are (it depends on c_K and β), as well as on the decay of the eigenvalues and the eigengaps. Notice that for a fixed kernel, c_K is fixed; however, by changing the kernel (e.g., by changing the bandwidth of the Gaussian kernel), we change the notion of smoothness and therefore its value. Because both c_K and the eigenvalues depend on the kernel, tuning parameters associated to K severely affect these rates. Therefore, it is of key importance to choose the tuning parameters properly — a problem we address in Section 4.2.1.

Next, to provide some insight regarding the effect of estimating the basis on the final estimator, we work out the details of two examples where the eigenvalues follow a polynomial decay. That is, we assume $\lambda_j \asymp j^{-2\alpha}$ for some $\alpha > \frac{1}{2}$. See Ji et al. (2012) for some empirical motivations of why this typically holds in practice, and Steinwart et al. (2009) for some theory and examples.

Example 4.1. In the limit of infinite unlabeled data (i.e., $m \longrightarrow \infty$) and for a power law decay of the eigenvalues and a fixed kernel K, the bound from Theorem 4.1 becomes

$$Ij \times O_P\left(\frac{1}{n}\right) + O\left(\frac{1}{j^{2\alpha}}\right) + O\left(\frac{1}{i^{2\beta}}\right).$$

In this case, the optimal cutoffs are $i \asymp n^{\frac{\alpha}{2\alpha\beta+\alpha+\beta}}$ and $j \asymp n^{\frac{\beta}{2\alpha\beta+\alpha+\beta}}$, yielding the rate $O_P\left(n^{-\frac{2\alpha\beta}{2\alpha\beta+\alpha+\beta}}\right)$. □

Example 4.2. Assume the same eigenvalue decay as in Example 4.1. Then $\lambda_j - \lambda_{j+1} = O\left(j^{-2\alpha-1}\right)$. If the number of observations used to estimate the basis is the same as that used to estimate the coefficients of the expansion, i.e., if $n = m$, then the optimal cutoffs for the bound from Theorem 4.1 are $i \asymp n^{\frac{\alpha}{8\alpha\beta+\alpha+3\beta}}$ and $j \asymp n^{\frac{\beta}{8\alpha\beta+\alpha+3\beta}}$, yielding the rate $O_P\left(n^{-\frac{2\alpha\beta}{8\alpha\beta+\alpha+3\beta}}\right)$. □

By comparing the rates from Examples 4.1 and 4.2, we see that estimating the basis $(\psi_j)_j$ decreases the rate of convergence. On the other hand, the possibility of using unlabeled data as described in Section 4.2 attenuates this problem.

As an illustration, consider that the RKHS $\mathcal{H}$ in Assumption 4.5 is the isotropic Sobolev space with smoothness $s > d$ and $\beta = s$ (i.e., $f(z|\mathbf{x})$ belongs to a Sobolev space with the same smoothness on both directions). It is known that under some conditions on the domain $\mathcal{X}$, $\lambda_J \asymp J^{-2s/d}$ (see more details on, e.g., Steinwart et al. 2009; Koltchinskii and Yuan 2010). In this case, the rate achieved in the limit of infinite unlabeled data is $O_P\left(n^{-\frac{2s}{2s+(1+d)}}\right)$, the standard minimax rate for estimating functions in $d+1$ dimensions (recall the conditional density is defined on $d+1$ dimensions) (Stone 1982; Hoffmann and Lepski 2002). If $m = n$, the rate is $O_P\left(n^{-\frac{2s}{8s+(1+3d)}}\right)$. Notice similar rates are obtained when learning *regression* functions via RKHS's (see discussion in, e.g., Ye and Zhou 2008 and Steinwart et al. 2009). Although the technique of assuming the kernel is fixed is common in the literature, in practice one often chooses a kernel that adapts to the data. This may substantially increase the performance of the algorithms (Steinwart et al. 2009). In what follows we present rates for variable Gaussian kernels, where we use the eigenfunctions of the operator of Equation (2.5), and moreover measure smoothness via a *density weighted* operator rather than RKHSs: this is a more natural measure under the manifold assumption (recall Chapter 3). With this, it is possible to obtain better bounds.

Assumption 4.6. (Smoothness in $\mathbf{x}$ direction) $\forall z \in [0,1]$ *fixed,* $\int_{\mathcal{X}} \|\nabla f(z|\mathbf{x})\|^2 dS(\mathbf{x}) < c_z$, *where* c_z*'s are such that* $\int_{[0,1]} c_z dz < \infty$.

Theorem 4.2. *Let* $\widehat{f}_{I,J}(z|\mathbf{x})$ *be the spectral series estimator from Section 4.2 with cutoffs* I *and* J, *based on the eigenfunctions of operator of Equation (2.5). Assume 4.1-4.4 and 4.6. Moreover, assume* $n = m$ *and that the kernel* $k = k_\epsilon^*$ *is corrected for bias*[2]. *Then, if*

2 That is, the kernel is used is $K_\epsilon^*(\mathbf{x},\mathbf{y}) = \frac{K_\epsilon(\mathbf{x},\mathbf{y})}{\sqrt{p_\epsilon(\mathbf{x})}}$, where K_ϵ is the original kernel (Coifman and Lafon 2006).

the support of the data is on a manifold with intrinsic dimension p*, and under regularity conditions (see the Appendix A), we have that, for* $\epsilon \asymp n^{-2/(p+4)}$*,*

$$L(\widehat{f}_{I,J}, f) = O\left(\frac{1}{J^{2/p}}\right) + O\left(\frac{1}{I^{2\beta}}\right) + IJ^{2\left(1-\frac{1}{p}\right)} O_P\left(\frac{\log n}{n}\right)^{\frac{2}{p+4}}.$$

It is then optimal to take $J \asymp n^{\frac{4\beta}{(p+4)(2/p+4\beta)}}$ *and* $I \asymp n^{\frac{4}{p(p+4)(2/p+4\beta)}}$*, in which case the upper bound becomes*

$$O_P\left(n^{\frac{-4\beta}{(p+4)(1+2\beta p)}}\right) = O_P\left(n^{-\frac{1}{O(p^2)}}\right).$$

If, on the other hand, there is infinite unlabeled data, we have

$$L(\widehat{f}_{I,J}, f) = O\left(\frac{1}{J^{2/p}}\right) + O\left(\frac{1}{I^{2\beta}}\right) + IJO_P\left(\frac{1}{n}\right),$$

in which case it is optimal to take $I \asymp n^{\frac{1}{2\beta+1+p\beta}}$ *and* $J \asymp n^{\frac{p\beta}{2\beta+1+p\beta}}$*, yielding the rate*

$$O_P\left(n^{-\frac{2\beta}{2\beta+1+\beta p}}\right) = O_P\left(n^{-\frac{1}{O(p)}}\right).$$

Theorem 4.2 shows that the rates of convergence of the spectral series estimator depend only on the intrinsic dimensionality p. More precisely, in the limit of infinite unlabeled data, the rates are of the form $O_P\left(n^{-1/O(p)}\right)$. Hence, the estimator adapts to the intrinsic dimensionality p, which can be much smaller than the *ambient* dimensionality d. Recall that standard rates are of the form $O_P\left(n^{-1/O(d)}\right)$ (Hall et al. 2004), which are much slower if $p \ll d$. In fact, in the isotropic scenario (i.e., $\beta = 1$ due to Assumption 4.6), the estimator achieves the rate $O_P\left(n^{-2/(2+(1+p))}\right)$, which is the minimax rate for estimators defined on $p+1$ dimensions. If there is no unlabeled data, the rate is of the form $O_P\left(n^{-1/O(p^2)}\right)$, which is still much better than $O_P\left(n^{-1/O(d)}\right)$ if $p \ll d$.

4.4 Applications

We illustrate our method using both simulated and real-world data: In Section 4.4.1, we provide a numerical example with simulated data on a low-dimensional

submanifold. Sections 4.4.2 and 4.4.3 show examples where the covariates $\mathbf{x}$ are true observations of high-dimensional data, namely images of digits and spectra of galaxies, respectively. In both examples we have added a continuous predictor Z with the goal of illustrating the method. Our main application is in Section 4.4.4, where we deal with the problem photometric redshift prediction.

We compare the results of five different methods: The first method, *Series*, is our proposed spectral series estimator using a radial Gaussian kernel and a Fourier basis for the ϕ_i's. The parameters I, J and the bandwidth of the kernel are chosen according to Section 4.2.1. *SeriesDiff* is also the spectral series estimator, but using the basis from the diffusion operator of Section 4.2.3. *KDE* is the kernel density estimator with a Gaussian kernel. We use the package NP in R (Hayfield and Racine 2008) to implement it. In the example of Section 4.4.1, we are able to use the least-squares cross-validation from Hall et al. (2004) to choose the bandwidth. On the other examples, this would be too time consuming due to the large sample sizes and number of covariates, therefore we use the rule-of-thumb implemented in the package. *KNN* is the kernel nearest neighbors approach (Zhao and Liu 1985), with the bandwidth and the number of nearest neighbors chosen by cross-validation. The last estimator is *LS*, the direct least squares conditional density estimator (Sugiyama et al. 2010b). We use the MATLAB implementation provided by the authors.[3] Their approach has some similarities to ours: their estimator also consists of a direct expansion of $f(z|\mathbf{x})$ onto functions ψ; however, the functions do not have to be orthogonal — which makes tuning parameters more time-consuming — nor do they have to form a basis for functions in $\mathbb{R}^{d+1}$. Moreover, such functions do not behave like a Fourier series, where lower-order terms are smoother than higher-order terms.

To evaluate the final estimates, we use three diagnostic tests. Similar tests have been introduced in the time series literature (see, e.g., Corradi and Swanson 2006).

3 http://sugiyama-www.cs.titech.ac.jp/~sugi/software/LSCDE/index.html

Denoting by $\widehat{F}_{z|x_i}$ the estimated conditional cumulative distribution function for z given the covariates $\mathbf{x}_i$, we use

1. (**QQPlot**) For each c in a grid of values in $[0,1]$ and each data point i in the test sample, compute $Q_i^c = \widehat{F}^{-1}_{z|x_i}(c)$. Define $\widehat{c} = \frac{1}{n}\sum_{i=1}^n \mathbb{I}(Z_i \leqslant Q_i^c)$. If the estimates are reasonable, $\widehat{c} \approx c$. We plot a graph of $\widehat{c}$'s versus c's and see how close they are to the diagonal.

2. (**P-value**) For each test data point i, let $U_i = \widehat{F}_{z|x_i}(Z_i)$. If the estimates are correct, $U_1, \ldots, U_n \overset{\text{iid}}{\sim} \mathtt{Unif}(0,1)$. We compute a p-value based on a Kolmogorov-Smirnoff test that compares the distributions of these statistics to the uniform distribution.

3. (**Coverage Plot**) For each α in a grid of values in $[0,1]$ and each data point i in the test sample, let A_i be a set such that $\int_{A_i} \widehat{f}(z|\mathbf{x}_i)dz = \alpha$. Here we choose the set A_i with the smallest area (the highest density region). Define $\widehat{\alpha}_i = \frac{1}{n}\sum_{i=1}^n \mathbb{I}(Z_i \in A_i)$. If the estimates are reasonable, $\widehat{\alpha} \approx \alpha$. We plot a graph of $\widehat{\alpha}$'s versus α's and see how close they are to the diagonal. For each α, we also include a 95% confidence interval based on a normal approximation to the binomial distribution.

We also estimate the loss (4.6) (up to the constant C) for the different procedures, and calculate approximate 95% confidence intervals based on 500 bootstrap samples.

4.4.1 *Klein Bottle*

In the first example, we simulate data with support close to a submanifold: data are i.i.d. observations of $(X^{(1)}, X^{(2)}, X^{(3)}, X^{(4)}, Z)$, where

$$\begin{cases} X^{(1)} = 2(\cos V + 1)\cos U + N(0,1) \\ X^{(2)} = 2(\cos V + 1)\sin U + N(0,1) \\ X^{(3)} = 2\sin V \cos U/2 + N(0,1) \\ X^{(4)} = 2\sin V \sin U/2 + N(0,1) \\ Z = \frac{1}{2}N(U, 0.5) + \frac{1}{2}N(V, 0.5) \end{cases}$$

where $U, V \overset{\text{iid}}{\sim} U(0, 2\pi)$. Hence, data $\mathbf{x} = (x^{(i)})_{i=1}^{4}$ lie *close to* a two-dimensional Klein Bottle embedded in $\mathbb{R}^4$. The goal is to estimate the conditional density $f(z|\mathbf{x})$. We use 3,000 samples for training, 1,000 for validation, and 1,000 for testing.

Table 4.1 shows the estimated losses of the various estimators. The best performance is achieved by the two spectral series estimators and the kernel density estimator (KDE). The computational time of KDE is however 450 times slower than Series; Series takes less than two minutes on a 2.70GHz Intel Core i7-4800MQ CPU, but KDE takes around 14 hours due to the slow model selection procedure. Figure 4.3 shows the diagnostic measures of the series estimator; it indicates that the final estimates are reasonable. This is in agreement with the p-value given by the Kolmogorov-Smirnoff test, which is 0.794.

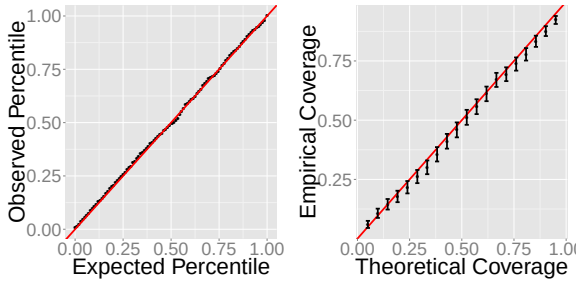


Figure 4.3: Diagnostic tests for the spectral series estimator for klein bottle data of Example 4.4.1.

4.4.2 *ZIP Code Data*

We use the ZIP Code database from USPS (Hastie et al. 2001). The image of each digit is represented by a vector of covariates $\mathbf{x} \in \mathbb{R}^{16\times 16}$. Hence, data lives in a 256-dimensional space. To illustrate our method, we use the data for conditional density estimation of a *simulated* continuous response variable; this is not a classification problem. We generate a response Z (not part of the original data) by

$$Z|\mathbf{X} = \mathbf{x} \sim d(\mathbf{x}) + \mathrm{Beta}(d(\mathbf{x}) + 2, d(\mathbf{x}) + 2) - \frac{1}{2},$$

where $d(\mathbf{x})$ is the digit corresponding to image $\mathbf{x}$. That is, Z has a shifted beta distribution, with mode on $d(\mathbf{x})$ and support $[d(\mathbf{x}) - 1/2, d(\mathbf{x}) + 1/2]$. This particular distribution was chosen for illustration purposes; we tried other distributions with similar results. We use 4,739 samples for training, 2,552 for validation and 2,007 for testing.

Table 4.1 shows the results of fitting the estimators. It indicates that best performance is obtained by the spectral series estimators. The only estimator that is able to get close to is KNN, however even it does not perform as well as Series. Figure 4.4 shows the diagnostic plots, which indicate that the spectral series method yields reasonable estimates (although they can still be improved; the p-value is 0.002). It also presents examples of the estimated densities for 4 samples; the estimates are indeed close to the real densities.

4.4.3 *Galaxy Spectra Data*

Next we apply the series method to the problem of predicting galaxy redshift (z) using its spectra ($\mathbf{x}$) (in our case, flux measurements at 3501 different wavelengths) from the Sloan Digital Sky Survey (SDSS). This is the same data as that of Section 3.5.2.

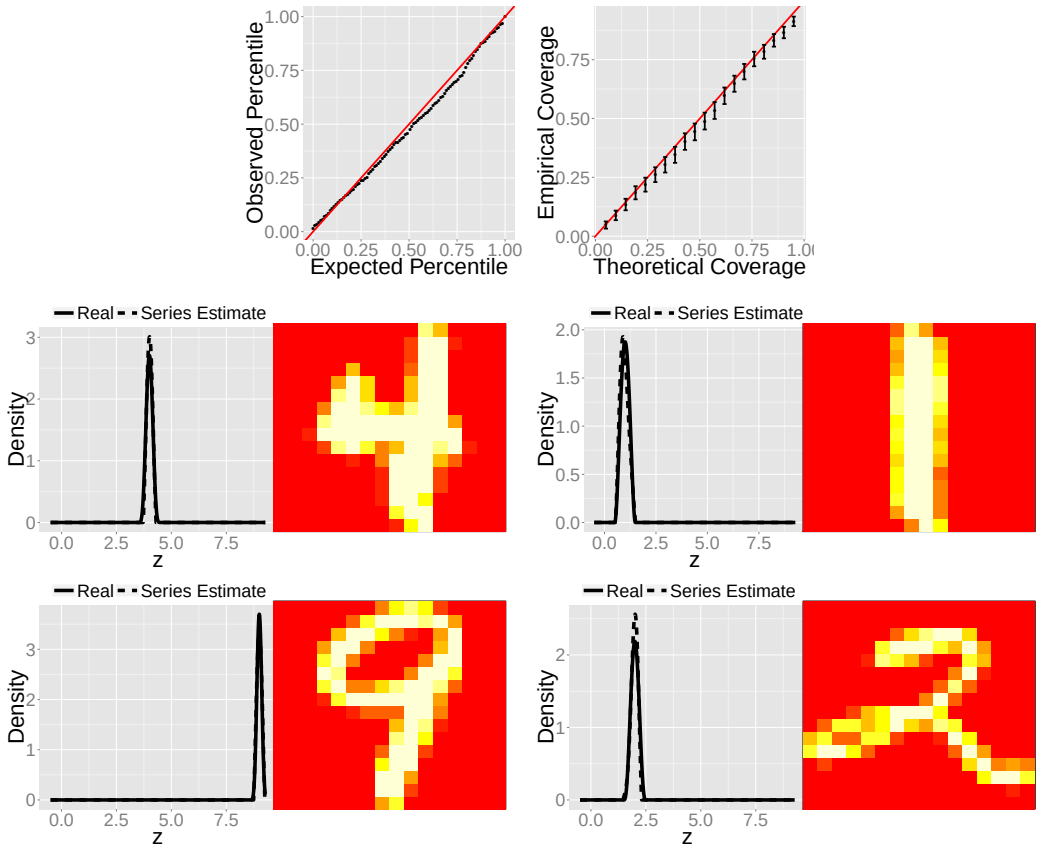


Figure 4.4: ZIP code data from Example 4.4.2: diagnostic tests for the spectral series method (top row); estimated conditional densities of the simulated response Z for 4 test samples (bottom rows). Although the covariate space has $d = 256$ covariates, the spectral series estimator yields reasonable estimates of $f(z|\mathbf{x})$.

Because the spectra determines redshift with great precision (this is not the case for photometric data, see Section 4.4.4 and Chapter 7), the conditional density $f(z|\mathbf{x})$ is typically degenerate. For the sake of illustrating the method, we proceed as in the digits example, and add noise to the real redshift in order to have a random variable with a genuine probability density function. More precisely, we create the variable

$$z_i = z_i^{\text{SDSS}} + \epsilon_i,$$

where ϵ_i are i.i.d. $N(0, 0.02)$ and z_i^{SDSS} is the true redshift galaxy i. Hence, $f(z_i|\mathbf{x}_i)$ is Gaussian with mean z_i^{SDSS} and variance 0.02, because $\mathbf{x}_i$ uniquely determines z_i^{SDSS}.

Table 4.1 shows the results of fitting different estimators. It indicates that best performance is obtained by the spectral series estimators. Because of the high-dimensionality, KDE does not yield reasonable estimates: both the numerator and denominator of the ratio are typically very close to zero, yielding numerical instabilities to the estimates. Figure 4.5 shows the diagnostic plots, which indicate that the spectral series method yields reasonable estimates. The goodness-of-fit is also indicated by the Kolmogorov-Smirnoff test p-value, 0.874. Figure 4.5 also presents examples of the estimated densities for 4 samples; the estimates are indeed close to the real densities.

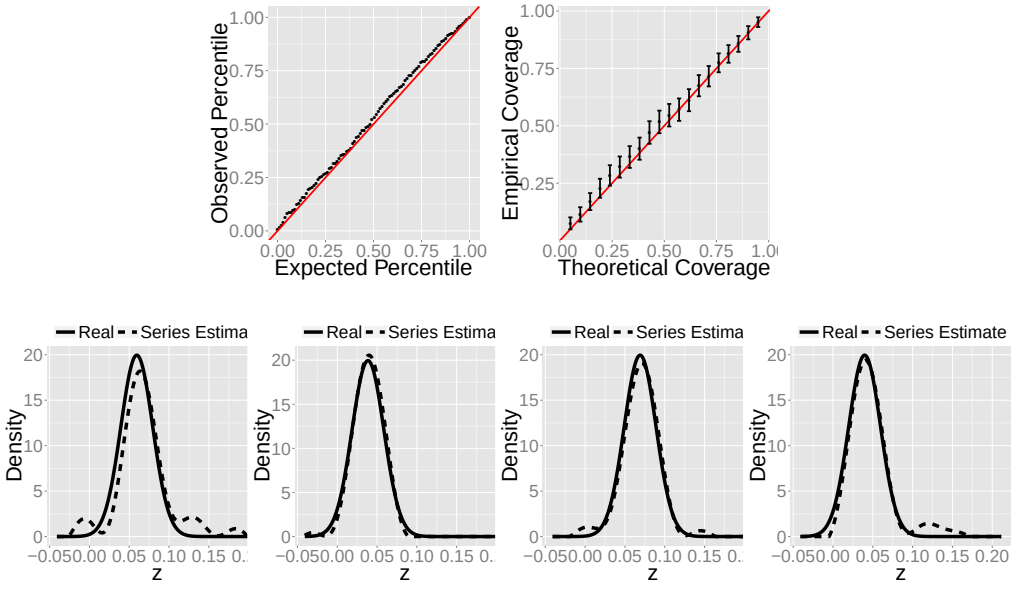


Figure 4.5: Spectra data from Example 4.4.3: diagnostic tests for the spectral series method (two row); estimated and real conditional densities of the simulated response Z for 4 test samples (bottom row). Although the covariate space has dimension $d = 3501$, the spectral series estimator yields reasonable estimates of $f(z|\mathbf{x})$.

4.4.4 *Main Application: Photometric Redshift Prediction*

We now present the results of our main application. The goal is again to estimate galaxy redshifts. However, rather then using spectra as in Section 4.4.3, we use photometric data. Photometry is low-resolution spectroscopy, where all the photons spanning the range of human vision are collected in several wavelength bins. While spectroscopy allows one to estimate z with extremely high accuracy, it is not as time-efficient as photometry, hence the need for methods based on photometric data. Here, we are interested in five magnitudes (logarithmic measures of photon flux in each wavelength bin), denoted by u, g, r, i, and z. The differences between contiguous magnitudes (e.g., $g - r$) are named colors and are used as covariates for estimating z. Several systems exist for defining the magnitudes of a galaxy, here we work with five of them: `psf`, `fiber`, `petrosian`, `model` and `cmodel`. See additional details in Chapter 7. We are interested in estimating the conditional density $f(z|\mathbf{x})$, where $\mathbf{x}$'s are the observed colors. We train our model using redshifts obtained via spectroscopy.

The first application is to a subset of Sloan Digital Sky Survey Data (SDSS)[4] that contains information about 3 colors in 4 magnitude systems[5] on 3,000 red luminous galaxies. Hence there are 3*4=12 covariates. More details about this data can be found in Freeman et al. (2009). Although 12 is relatively small, most conditional density estimators already fail for such d. Moreover, because photometric covariates are derived from the (nonobserved) spectrum of a galaxy, we expect them to live in a smaller dimensional space. Hence, it is expected the spectral series estimator will have better performance than traditional methods. We use 70% of the data for training, 15% for validation and 15% for testing.

The next-to-last row of Table 4.1 shows the estimated loss of the conditional density estimators. Again, spectral series estimators have the best performance, followed by KNN (the estimator used in astronomy literature). The top row of

4 http://www.sdss.org/

5 The colors are $g - r$, $r - i$, and $i - z$, and are observed in `psf`, `fiber`, `petrosian` and `model` magnitudes.

Figure 4.6 presents the diagnostic plots, which indicate that the series model indeed has good performance. This is in agreement with the Kolmogorov-Smirnoff p-value, 0.393.

In Figure 4.6 we also display the estimated densities for the four selected samples from Figure 4.1. As expected, the estimates vary smoothly as a function of the eigencoordinates, but can be quite different in different regions of the map. Notice some of the estimates have multimodalities and asymmetries. These distributions are informative to astronomers: they are typically galaxies where a regression estimate is not accurate, and hence using it may induce large errors. In such cases, it is recommended to work with the entire distribution; see Ball and Brunner (2010) for a review.

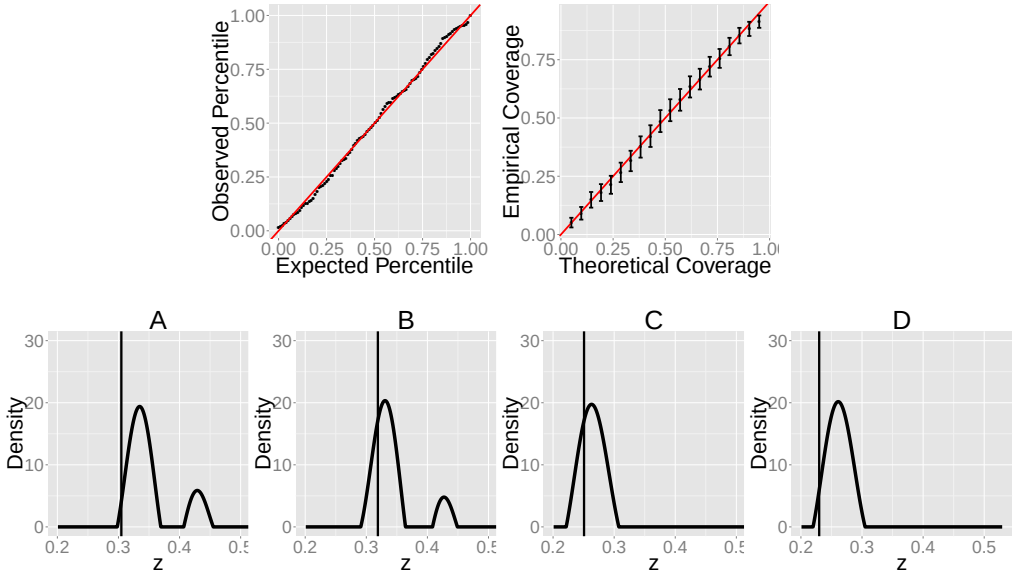


Figure 4.6: Photometry on red luminous galaxies of SDSS: diagnostic tests for the spectral series method (top row), and estimated densities for 4 galaxies (A,B,C, and D) from Figure 4.1 (bottom row). Vertical lines in the bottom plots indicate spectroscopically observed redshift.

Next, we apply that methods to data from Sheldon et al. (2012)[6], which contains information on `model` and `cmodel` magnitudes of u, g, r, i, and z bands for galaxies

6 See more details about such data in Chapter 7.

of several surveys. Besides using the colors from this bands, we also use the raw value of the r-band, as Sheldon et al. (2012) do. Hence, there are 10 covariates. We use 5,000 samples for training, 2,500 for validation and 2,500 for testing. Results are shown in the last row of Table 4.1. Again series estimators have better performance. The top row of Figure 4.7 shows the diagnostic plots. The goodness-of-fit is not as good as that of the model for red luminous galaxies. This is possibly because this dataset contains fainter galaxies, and is also contaminated by stars. Hence, it is harder to estimate $f(z|\mathbf{x})$, which is expected to be more multimodal and asymmetric; nevertheless, the spectral series still has better performance than state-of-the-art methods. The plots in the bottom row show that indeed some galaxies have very wiggly estimates.

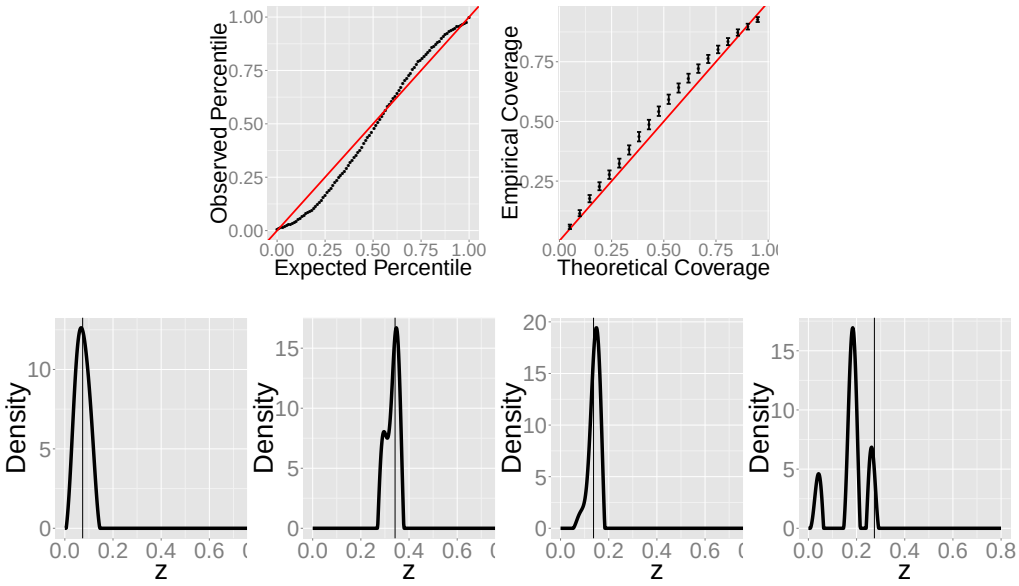


Figure 4.7: Photometry on data from Sheldon et al. (2012): diagnostic tests for the spectral series method (two row); and estimated densities for 4 random test galaxies (bottom row). Vertical lines in the bottom plots indicate spectroscopically observed redshift.

Table 4.1: Estimated L^2 loss (with standard errors) of the conditional density estimators. Best-performing models with smallest loss are in bold fonts.

Dataset	**Dim**	**Loss**				
		Series	*SeriesDiff*	*KDE*	*KNN*	*LS*
Klein Bottle	4	**-0.81 (0.02)**	**-0.80 (0.02)**	**-0.79 (0.02)**	-0.76 (0.02)	-0.47 (-0.02)
ZIP Code	256	**-8.10 (0.25)**	**-7.91 (0.27)**	-0.78 (0.02)	-7.25 (0.23)	-0.22 (0.003)
Spectra	3501	**-1.75 (0.06)**	**-1.77 (0.07)**	—*	-1.61 (0.07)	-0.256 (0.02)
Photo-z (Red)	12	**-1.88 (0.07)**	**-1.84 (0.06)**	-1.53 (0.03)	-1.72 (0.07)	-1.53 (0.05)
Photo-z (All)	10	**-11.81 (0.20)**	**-11.49 (0.21)**	-7.22 (0.06)	–11.06 (0.21)	-8.49 (0.25)

*Due to the high-dimensionality, numerical instabilities such as divisions by 0 in KDE make such estimator not usable, see text for details.

We now use the data from Sheldon et al. (2012) to investigate the computational shortcuts described in Section 2.2.2. First, we compare the statistical and computational performance of standard SVD when compared to Randomized SVD. We use a fixed number of 3,000 validation samples and 10,000 testing samples, and vary the training sample size. We always use $J = 600$. Results are on the top row of Figure 4.8. Not only does Randomized SVD yield essentially the same loss as standard SVD for all sample sizes, but it also results in much smaller computational costs, especially for large sample sizes. Hence, Randomized SVD brings computational efficiency with no loss in statistical performance. The bottom row of Figure 4.8 shows the results of the second experiment, where we evaluate the benefits of using sparse Gram matrices. We use 5,000 training, 2,000 validation and 2,000 testing samples, and change the cutoff ξ that defines sparseness. For this sample size, it is possible to save 30% of the memory with almost no loss in statistical performance; hence it is possible to use a 42% larger training set with the same memory cost. Typically, as the training sample size increases, ξ can be made even smaller due to the larger number of neighbors each sample has.

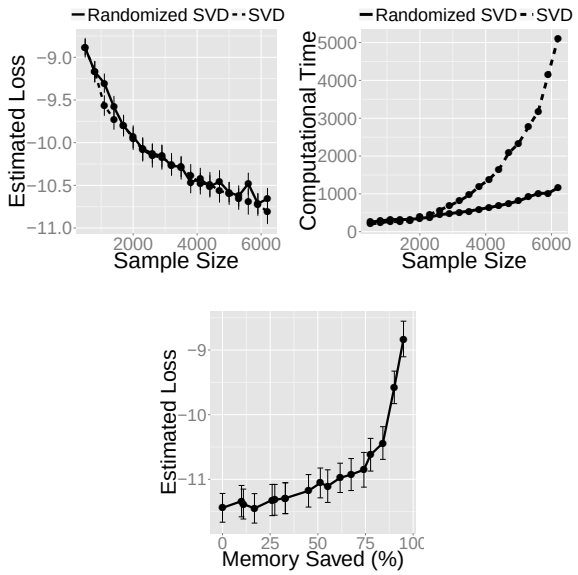


Figure 4.8: Top row: Benefits of using the randomized SVD from Halko et al. (2011). There is a substantial gain in time for large sample sizes, with almost no loss in statistical performance. Bottom row: Benefits of using sparse gram matrices; for this sample size (5,000), it is possible to reduce the memory use in about 30% with almost no loss in statistical performance; see text for details.

4.4.5 *Summary of the Experimental Results*

Our main findings in the experiments were:

- Both normalizations of the kernel operator, *kernel PCA* and *diffusion*, yield spectral series estimators with similar performances, which dominate the other estimators for datasets with dimension at least 10.
- Spectral series also have the advantage of being fast, specially after the computational improvements described in Section 2.2.2. Such improvements allow the estimator to scale to larger datasets with almost no decrease of statistical performance.
- In high-dimensions, KNN yields better estimates than KDE and LS, although our proposed estimators typically have better performance.

- KDE performs well if the sample space dimension is small. However, in higher dimensions, using the bandwidth selection method from Hall et al. (2004) is computationally very intensive. Moreover, because KDE is based on ratios, it suffers from high numerical instabilities due to divisions by 0.

In Chapter 7 we present a more thorough analysis of the problem of predicting redshift based on photometric data, where we also take into account the selection bias that often occurs in such application.

5

DENSITY RATIO ESTIMATION

5.1 Introduction

There has been growing interest in the problem of estimating the ratio of two probability densities, $\beta(\mathbf{x}) \equiv f(\mathbf{x})/g(\mathbf{x})$, given i.i.d. samples from unknown distributions F and G. For example, these ratios play a key role in matching training and test data in so-called transfer learning or domain adaptation (Sugiyama et al. 2010a), where the goal is to predict an outcome y given test data $\mathbf{x}$ from a distribution (G) that is different from that of the training data (F). Estimated density ratios also appear in novelty detection (Hido et al. 2011), conditional density estimation (Sugiyama et al. 2010b), selection bias correction (Gretton et al. 2010), and classification (Nam et al. 2012).

Experiments have shown that it is suboptimal to estimate $\beta(\mathbf{x})$ by first estimating the two component densities and then taking their ratio (Sugiyama et al. 2008). Hence, several alternative approaches have been proposed that directly estimate $\beta(\mathbf{x})$; e.g., *uLSIF*, an estimator obtained via least-squares minimization (Kanamori et al. 2009); *KLIEP*, which is obtained via Kullback-Leibler divergence minimization (Sugiyama et al. 2008); *KuLSIF*, a kernelized version of *uLSIF* (Kanamori et al. 2012); and *kernel mean matching*, which is based on minimizing the mean discrepancy between transformations of the two samples in a Reproducing Kernel Hilbert Space (RKHS) (Gretton et al. 2010). For a review of techniques see Margolis (2011).

Existing methods are not effective when $\mathbf{x}$ is of high dimension, and hence authors recommend a dimension reduction prior to implementation (Sugiyama et al. 2011), which can result in significant loss of information. Here we show to use spectral series to approximate this quantity.

This chapter is organized as follows. In Section 5.2 we explain how to use spectral series to estimate a density ratio. In Section 5.3, we present some theoretical guarantees on the estimator. Finally, in Section 5.4, we compare our estimators with other approaches.

5.2 Methodology

In this section we will present the mathematical details behind the spectral series estimator of a density ratio. To begin, let $\mathbf{x}$ denote a d-dimensional random vector, assumed to lie in the subspace $\mathcal{X}$. We observe an i.i.d. sample $\mathbf{x}_1^F, \ldots, \mathbf{x}_{n_F}^F$ from an unknown distribution F, as well as an i.i.d. sample $\mathbf{x}_1^G, \ldots, \mathbf{x}_{n_G}^G$ from an unknown distribution G. The goal is to estimate

$$\beta(\mathbf{x}) \equiv f(\mathbf{x})/g(\mathbf{x}).$$

We assume that $F \ll G$ so that this ratio is well-defined.

For this task, we define the kernel operator of Equation (2.2) with respect to population G, that is, we use the eigenfunctions $\{\psi_j\}_{j\in\mathbb{N}}$ of the operator $\mathbf{K}_\mathbf{x} : L^2(\mathcal{X}, G) \longrightarrow L^2(\mathcal{X}, G)$:

$$\mathbf{K}_\mathbf{x}(h)(\mathbf{z}) = \int_{\mathcal{X}} K_\mathbf{x}(\mathbf{z}, \mathbf{y}) h(\mathbf{y}) dG(\mathbf{y}). \tag{5.1}$$

Hence, $\{\psi_j\}_{j\in\mathbb{N}}$ is an orthonormal basis of $L^2(\mathcal{X}, G)$, i.e., the eigenfunctions are orthonormal with respect to the data distribution G:

$$\int_{\mathcal{X}} \psi_i(\mathbf{x})\psi_j(\mathbf{x}) dG(\mathbf{x}) = \mathbb{I}(i = j).$$

Therefore, for $\beta(\mathbf{x}) \in L^2(\mathcal{X}, G)$, we can write

$$\beta(\mathbf{x}) = \sum_{j\in\mathbb{N}} \beta_j \psi_j(\mathbf{x}), \tag{5.2}$$

where $\beta_j = \int \beta(x)\psi_j(x)dG(x) = \mathbb{E}_F[\psi_j(X)]$.

To estimate ψ_j's, we proceed just as in Section 2.2.1, but using the Gram matrix based on the sample from G,

$$\begin{pmatrix} K_x(x_1^G, x_1^G) & K_x(x_1^G, x_2^G) & \cdots & K_x(x_1^G, x_{n_G}^G) \\ K_x(x_2^G, x_1^G) & K_x(x_2^G, x_2^G) & \cdots & K_x(x_2^G, x_{n_G}^G) \\ \vdots & \vdots & \ddots & \vdots \\ K_x(x_{n_G}^G, x_1^G) & K_x(x_{n_G}^G, x_2^G) & \cdots & K_x(x_{n_G}^G, x_{n_G}^G) \end{pmatrix}$$

Next we estimate the β_j's in Eq. (5.2) using the sample from F:

$$\widehat{\beta}_j = \frac{1}{n_F} \sum_{k=1}^{n_F} \widehat{\psi}_j \left(x_k^F\right).$$

Our spectral series estimator is finally given by

$$\widehat{\beta}(x) = \left(\sum_{j=1}^{J} \widehat{\beta}_j \widehat{\psi}_j(x) \right)_+ . \tag{5.3}$$

The tuning parameter J controls the bias/variance tradeoff: Decreasing J decreases the variance, but increases the bias of the estimator. We choose J (and the other tuning parameters) in a principled way described below.

5.2.1 *Loss Function and Tuning of Parameters*

To evaluate the performance of an estimator $\widehat{\beta}(x)$, we use the loss function

$$\begin{aligned} L(\widehat{\beta}, \beta) &\equiv \int \left(\widehat{\beta}(x) - \beta(x)\right)^2 dG(x) \\ &= \int \widehat{\beta}(x)^2 dG(x) - 2\int \widehat{\beta}(x) dF(x) + K \end{aligned}$$

where K does not depend on $\widehat{\beta}$. We estimate this quantity (up to K) using

$$\widehat{L}(\widehat{\beta}, \beta) = \frac{1}{\widetilde{n}_G} \sum_{k=1}^{\widetilde{n}_G} \widehat{\beta}^2(\widetilde{x}_k^G) - \frac{2}{\widetilde{n}_F} \sum_{k=1}^{\widetilde{n}_F} \widehat{\beta}(\widetilde{x}_k^F), \tag{5.4}$$

where $\widetilde{x}_1^G, \ldots, \widetilde{x}_{\widetilde{n}_G}^G$ is a validation sample from G, and $\widetilde{x}_1^F, \ldots, \widetilde{x}_{\widetilde{n}_F}^F$ is a validation sample from F. Tuning parameters are chosen to minimize $\widehat{L}(\widehat{\beta}, \beta)$. Note that because of the orthogonality of the $\widehat{\psi}_j$, it is not necessary to recompute the estimated coefficients $\widehat{\beta}_j$'s for each value of J in Eq. (5.3), unlike most estimation procedures, where estimated coefficients have to be recomputed for each configuration of the tuning parameters. In other words, only the tuning parameters associated with the kernel (in our case, the kernel bandwidth ϵ) affect the computation time.

Remark: The density ratio can more generally be defined using Radon-Nikodym derivatives: $\beta(x) = \frac{dF}{dG}(x)$, which can handle cases such as, e.g., text data, functional data and other data types common in modern applications, in which the distributions F and G are not dominated by Lebesgue measure. The methodology we develop still applies as long as on is able to create a meaningful similarity function between two samples (more specifically, a kernel).

5.3 THEORY

Next we provide theoretical guarantees of the performance of the estimator $\widehat{\beta}$. The bounds we derive have the same nature as the bounds with a fixed kernel we derived in Chapters 3 and 4. In particular, the assumptions we make have a similar nature:

Assumption 5.1. $\int \beta^2(x) dG(x) < \infty$.

Assumption 5.2. $\lambda_1 > \lambda_2 > \ldots > \lambda_J > 0$.

Assumption 5.3. $c_{K_x} \equiv \|\beta(x)\|^2_{\mathcal{H}_{K_x}} < \infty$.

See the theory section of Chapters 3 and 4 for an interpretation of these. Under these assumptions, we have the following Theorem:

Theorem 5.1. *Under Assumptions 5.1 – 5.3, the loss* $\int \left(\widehat{\beta}_J(x) - \beta(x)\right)^2 dG(x)$ *is bounded by*

$$J \times \left[O_P\left(\frac{1}{n_F}\right) + O_P\left(\frac{1}{\lambda_J \Delta_J^2 n_G}\right) \right] + c_{K_x} O(\lambda_J),$$

where $\Delta_J = \min_{1 \leqslant j \leqslant J} |\lambda_j - \lambda_{j+1}|$ *and* $\widehat{\beta}_J(\mathbf{x})$ *is the spectral series ratio estimator truncated at* J.

See Izbicki et al. (2014) for additional details and proofs, which follow along the same lines of those presented in the Appendix of this thesis.

As an illustration, assume a *fixed* kernel $K_{\mathbf{x}}$. Then, if $n \equiv n_F = n_G$, $\lambda_J \asymp J^{-2\alpha}$ for some $\alpha > \frac{1}{2}$, and $\lambda_J - \lambda_{J+1} \asymp J^{-2\alpha-1}$ (see the reasons for such assumption in Chapter 4), then the optimal smoothing is given by $J \asymp n^{1/(8\alpha+3)}$. With this choice of J, the rate of convergence is

$$O_P\left(n^{-\frac{2\alpha}{8\alpha+3}}\right).$$

5.4 Application: Correction to Covariate Shift

Assume we observe a sample of unlabeled data, as well as a sample of labeled data, where the Z's represent the labels and $\mathbf{x}$'s are the covariates. One is often interested in estimating the regression function $\mathbb{E}[Z|\mathbf{x}]$ under selection bias, i.e., in situations where the distributions of labeled and unlabeled samples ($f_L(\mathbf{x})$ and $f_U(\mathbf{x})$, respectively) are different. If the estimate $\widehat{\mathbb{E}}[Z|\mathbf{x}]$ is constructed using the labeled data with the goal of predicting Z from $\mathbf{x}$ on the *unlabeled* data, corrections have to be made. A key quantity for making this correction under the *covariate shift assumption* (Shimodaira 2000) is the density ratio $f_U(\mathbf{x})/f_L(\mathbf{x})$, the so-called *importance weights* (Gretton et al. 2010). See Chapter 7 for a more detailed explanation. We now compare various estimators of importance weights for the problem of photometric redshift estimation.

We use a subset of the data described in Chapter 7. The ultimate goal is to build a predictor of galaxy redshift Z based on photometric data $\mathbf{x}$; see Chapter 7 for details. We are given a training set with covariates $\mathbf{x}$ of galaxies and their redshifts, as well as unlabeled target data. Because it is difficult to acquire the true redshift of faint galaxies, these data suffer from selection bias. We compare our method of estimating the importance weights (*Series*) to *uLSIF*, *KLIEP*, and *KuLSIF*, described

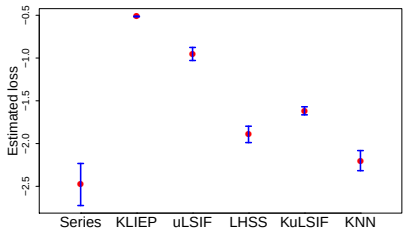


Figure 5.1: Estimated losses of $\widehat{\beta}(\mathbf{x})$ with standard errors for SDSS data. The spectral series estimator has best performance.

in the introduction of this chapter. We also compute *LHSS*, which uses *uLSIF* after applying a dimension reduction technique specifically designed for estimating a density ratio, see Sugiyama et al. (2011). Moreover, we include a comparison with a k-nearest neighbors estimator (*KNN*) proposed in the astronomy literature (Lima et al. 2008), which it not based on ratios, see Chapter 7 for more details. We do not show results of ratio-based estimators because the estimates of $f_L(\mathbf{x})$ are close to zero for many $\mathbf{x}$'s, inducing estimates of β that are infinity. Moreover, we do not compute the *kernel mean matching* estimate because it does not allow out-of-sample extrapolations, and hence does not permit the use of validation sets to tune parameters.

Figure 5.1 shows the estimated losses of the different methods of estimating the ratio $f_U(\mathbf{x})/f_L(\mathbf{x})$ when using 5,000 labeled and 5,000 unlabeled samples, and 10 photometric covariates $\mathbf{x}$. We use 60% of the data for training, 20% for validation and 20% for testing. Even though this example has a covariate space with as few as 10 dimensions, we can already see the benefits of the spectral series estimator. We leave more experiments to Chapter 7.

6

LIKELIHOOD FUNCTION ESTIMATION

6.1 Introduction

Estimation of the likelihood function is necessary when the complexity of the data-generation process prevents derivation of a sufficiently accurate analytical form for the likelihood function. Here we exploit the fact that, in many such situations, one can *simulate* data sets $\mathbf{x}$ under different parameters θ. This is often the case in statistical inference problems in the sciences, where the relationship between parameters of interest and observable data is complex, but accurate simulation models are available; see, for example, genetics (Beaumont 2010; Estoup et al. 2012) and astronomy (Cameron and Pettitt 2012; Weyant et al. 2013). Problems of this type have motivated recent interest in methods of *likelihood-free inference*, which includes methods of *Approximate Bayesian Computation (ABC)*; see Marin et al. (2012) for a review.

In our implementation, we redefine the likelihood function as $\mathcal{L}(\mathbf{x};\theta) \equiv f(\mathbf{x}|\theta)/g(\mathbf{x})$, where $g(\mathbf{x})$ is a density with support larger than that of $f(\mathbf{x}|\theta)$. This formulation differs from the standard definition of the likelihood by only a multiplicative term which is constant in θ, and hence $\mathcal{L}(\mathbf{x};\theta)$ can still be used for likelihood-based inference (including maximum likelihood estimation). In particular, the shape of the posterior for θ is unaffected. The challenge of estimating the likelihood is now a density ratio estimation problem. This approach will yield significant advantages

in cases where g is chosen to focus high probability on the low-dimensional subspace in which the data $\mathbf{x}$ lie. One natural choice is $g(\mathbf{x}) = \int f(\mathbf{x}|\theta)d\pi(\theta)$, where π is a well-chosen prior distribution for θ. The orthogonality of the spectral series with respect to g results in an efficient implementation of the estimator. Moreover, directly estimating the ratio $f(\mathbf{x}|\theta)/g(\mathbf{x})$ may itself be easier than estimating $f(\mathbf{x}|\theta)$, e.g., when the conditional distributions $f(\mathbf{x}|\theta)$ for different θ are similar,[1] or when they have similar support in high dimensions. To our knowledge, this is the first work that proposes a spectral series approach to non-parametric density estimation and likelihood inference in high dimensions, although there exist attempts of doing this in low dimensions (e.g., Diggle and Gratton 1984; Fan et al. 2013).

This chapter is organized as follows. In Section 6.2 we explain how to use spectral series to estimate a likelihood function. In Section 6.3, we present some theoretical guarantees on the estimator. Finally, in Section 6.4, we compare our estimators with other approaches.

6.2 METHODOLOGY

Let $\theta \in \Theta$ be a p-dimensional parameter. In this context, $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ is a random vector representing a single sample observation. We will adopt a Bayesian perspective, and let F_θ be the marginal distribution for θ, i.e., the prior, and let G denote the marginal distribution for $\mathbf{x}$. Then, let $(\mathbf{x}_1^F, \theta_1), \ldots, (\mathbf{x}_{n_F}^F, \theta_{n_F})$ be an i.i.d. sample from the joint distribution of $\mathbf{x}$ and θ. Further, let $\mathbf{x}_1^G, \ldots, \mathbf{x}_{n_G}^G$ be an i.i.d. sample from G. Our objective is to estimate the ratio

$$\mathcal{L}(\mathbf{x}; \theta) \equiv \frac{f(\mathbf{x}|\theta)}{g(\mathbf{x})}, \tag{6.1}$$

1 A trivial example: If $\mathbf{x}$ is independent of θ, $f(\mathbf{x}|\theta)/g(\mathbf{x}) = 1$ is a constant function, whereas $f(\mathbf{x}|\theta) = f(\mathbf{x})$ may be a harder to estimate (nonsmooth) function. Similarly, if $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$, and $\mathbf{x}_1$ is independent of $\mathbf{x}_2$ given θ, it can be shown $f(\mathbf{x}|\theta)/g(\mathbf{x})$ does not depend on $\mathbf{x}_2$, but $f(\mathbf{x}|\theta)$ does. Hence, $f(\mathbf{x}|\theta)$ is typically harder to estimate, because it is a function of more variables.

where $f(\mathbf{x}|\theta)$ is the conditional density of $\mathbf{x}$ given θ, and $g(\mathbf{x})$ is the marginal density for $\mathbf{x}$. This is, up to a multiplicative factor that is not a function of θ, the standard definition of the likelihood function.

To estimate $\mathcal{L}(x;\theta)$, we use a spectral series approach as in the other Chapters, but because the likelihood is a function of both $\mathbf{x}$ and θ, we consider the *tensor product* of a basis for $\mathbf{x}$ and a basis for θ, $\{\Psi_{i,j}\}_{i,j\in\mathbb{N}}$, where

$$\Psi_{i,j}(\mathbf{x},\theta) = \psi_j(\mathbf{x})\phi_i(\theta),\ i,j \in \mathbb{N}.$$

The construction of the separate bases $\{\psi_j\}_j$ and $\{\phi_i\}_i$ proceeds just as described in Section 2.2.1. Note that for θ, we consider the eigenfunctions $\{\phi_i\}_i$ of the operator $\mathbf{K}_\theta : L^2(\Theta, F_\theta) \longrightarrow L^2(\Theta, F_\theta)$:

$$\mathbf{K}_\theta(h)(\xi) = \int_\Theta K_\theta(\xi,\mu)h(\mu)dF_\theta(\mu),$$

where K_θ is not necessarily the same kernel as $K_{\mathbf{x}}$. That is, while $\{\psi_j\}_j$ is estimated using a Gram matrix based on $\mathbf{x}_1^G,\ldots,\mathbf{x}_{n_G}^G$, $\{\phi_i\}_i$ is estimated using $\theta_1,\ldots,\theta_{n_F}$.

Since $\{\phi_i\}_i$ is an orthonormal basis of functions in $L^2(\Theta, F_\theta)$, the tensor product $\{\Psi_{i,j}\}_{i,j}$ is an orthonormal basis for functions in $L^2(\Theta\times\mathcal{X}, F_\theta \times G)$[2].

The projection of $\mathcal{L}(\mathbf{x};\theta)$ onto $\{\Psi_{i,j}\}_{i,j}$ is given by

$$\sum_{i,j\in\mathbb{N}} \beta_{i,j}\Psi_{i,j}(\mathbf{x},\theta),$$

where

$$\begin{aligned}\beta_{i,j} &= \iint \mathcal{L}(\mathbf{x};\theta)\Psi_{i,j}(\mathbf{x},\theta)dG(\mathbf{x})dF_\theta(\theta)\\ &= \mathbb{E}_F\left[\Psi_{i,j}(\mathbf{x},\theta)\right].\end{aligned} \tag{6.2}$$

Hence, we define our likelihood function estimator by

$$\widehat{\mathcal{L}}(\mathbf{x};\theta) = \sum_{i=1}^{I}\sum_{j=1}^{J}\widehat{\beta}_{i,j}\widehat{\Psi}_{i,j}(\mathbf{x},\theta), \tag{6.3}$$

2 Notice that, contrary to what was done for conditional densities in Chapter 4, here we do not use a fixed basis neither for $\mathbf{x}$ nor for θ. The reason for this is that we want to avoid tensor products in higher dimensions.

where $\widehat{\beta}_{i,j} = \frac{1}{n_F} \sum_{k=1}^{n_F} \widehat{\Psi}_{i,j}\left(\mathbf{x}_k^F, \theta_k\right)$, and

$$\widehat{\Psi}_{i,j}(\mathbf{x}, \theta) = \widehat{\psi}_j(\mathbf{x})\widehat{\phi}_i(\theta)$$

is the estimator for $\Psi_{i,j}(\mathbf{x}, \theta)$, obtained via a Nyström extension as in Section 2.2.1. The tuning parameters I and J control the bias/variance tradeoff. Because we define the likelihood in terms of $g(\mathbf{x})$ (Eq. 6.1), we can take advantage of the orthogonality of the basis functions when estimating the coefficients $\beta_{i,j}$; see Eq. 6.2. The result is a simple and fast-to-implement procedure for estimating likelihood functions for high-dimensional data.

6.2.1 *Loss Function and Tuning of Parameters*

To evaluate the performance of a given estimator, we use the loss function

$$\begin{aligned} L\left(\widehat{\mathcal{L}}, \mathcal{L}\right) &\equiv \int \left(\widehat{\mathcal{L}}(\mathbf{x};\theta) - \mathcal{L}(\mathbf{x};\theta)\right)^2 dG(\mathbf{x})dF(\theta) \\ &= \int \widehat{\mathcal{L}}(\mathbf{x};\theta)^2 dG(\mathbf{x})dF(\theta) \\ &- 2\int \widehat{\mathcal{L}}(\mathbf{x};\theta)dF(\theta,\mathbf{x}) + K, \end{aligned} \tag{6.4}$$

where K does not depend on $\widehat{\mathcal{L}}$. We can estimate this quantity (up to K) by

$$\begin{aligned} \widehat{L}\left(\widehat{\mathcal{L}}, \mathcal{L}\right) =& \frac{1}{B}\sum_{l=1}^{B}\left[\frac{1}{\widetilde{n}}\sum_{k=1}^{\widetilde{n}}\left(\widehat{\mathcal{L}}\left(\widetilde{\mathbf{x}}_k^G|\widetilde{\theta}_k^{(l)}\right)\right)^2\right] \\ &- \frac{2}{\widetilde{n}}\sum_{k=1}^{\widetilde{n}}\widehat{\mathcal{L}}(\widetilde{\mathbf{x}}_k^F|\widetilde{\theta}_k), \end{aligned}$$

where $\widetilde{\mathbf{x}}_1^G, \ldots, \widetilde{\mathbf{x}}_{\widetilde{n}}^G$ is a validation sample from G; $(\widetilde{\theta}_1, \widetilde{\mathbf{x}}_1^F), \ldots, (\widetilde{\theta}_{\widetilde{n}}, \widetilde{\mathbf{x}}_{\widetilde{n}}^F)$ is a validation sample from the joint distribution of $\mathbf{x}$ and θ; $\widetilde{\theta}_1^{(l)}, \ldots, \widetilde{\theta}_{\widetilde{n}}^{(l)}$ for $l = 1, \ldots, B$ are random permutations of the original sample $\widetilde{\theta}_1, \ldots, \widetilde{\theta}_{\widetilde{n}}$; and B is a number limited only by computational considerations. We choose tuning parameters so as to minimize $\widehat{L}$.

Remark: As in this case of a ratio (Chapter 5), the likelihood function can more generally be defined using Radon-Nikodym derivatives: $\mathcal{L}(\mathbf{x};\theta) = \frac{dF}{dG}(\mathbf{x}|\theta)$, which

can handle cases such as, e.g., text data, functional data, and other data types common in modern applications, in which the distributions F and G are not dominated by Lebesgue measure. Again, the methodology we develop still applies as long as one is able to create a meaningful similarity function between two samples (more specifically, a kernel).

6.3 Theory

The bound we derive for the spectral series estimator of a likelihood function has the same nature as the bounds with a fixed kernel we derived in Chapters 3 and 4. In particular, the assumptions we make have a similar nature:

Assumption 6.1. $\int\int \mathcal{L}^2(\mathbf{x};\theta)dG(\mathbf{x})dF(\theta) < \infty$.

Assumption 6.2. $\lambda_1^{\mathbf{x}} > \lambda_2^{\mathbf{x}} > \ldots > \lambda_J^{\mathbf{x}} > 0$.

Assumption 6.3. $\lambda_1^{\theta} > \lambda_2^{\mathbf{x}} > \ldots > \lambda_I^{\theta} > 0$.

Assumption 6.4. *For all fixed* $\theta \in \Theta$, $\mathcal{L}(\mathbf{x};\theta) \in \{g \in \mathcal{H}_{K_{\mathbf{x}}} : \|g\|^2_{\mathcal{H}_{K_{\mathbf{x}}}} \leqslant c_\theta^2\}$ *where* c_θ*'s are such that* $c_{K_{\mathbf{x}}} \equiv \int_\Theta c_\theta^2 dF(\theta) < \infty$.

Assumption 6.5. *For all fixed* $\mathbf{x} \in \mathcal{X}$, $\mathcal{L}(\mathbf{x};\theta) \in \{h \in \mathcal{H}_{K_\theta} : \|h\|^2_{\mathcal{H}_{K_\theta}} \leqslant c_{\mathbf{x}}^2\}$ *where* $c_{\mathbf{x}}$*'s are such that* $c_{K_\theta} \equiv \int_{\mathcal{X}} c_{\mathbf{x}}^2 dG(\mathbf{x}) < \infty$.

Notice that, in these assumptions, the superscripts $\mathbf{x}$ and θ denote quantities associated with the eigenfunctions ψ_j's and ϕ_i's, respectively. See the theory section of Chapters 3 and 4 for an interpretation of these. Under these assumptions, we have the following Theorem:

Theorem 6.1. *Under Assumptions 6.1 – 6.5, the loss* $L\left(\widehat{\mathcal{L}}_{I,J}, \mathcal{L}\right)$ *is bounded by*

$$IJO_P\left(\max\left\{\frac{1}{\lambda_J^{\mathbf{x}}\Delta^2_{\mathbf{x},J}n_G}, \frac{1}{\lambda_I^{\theta}\Delta^2_{\theta,I}n_F}\right\}\right) + c_{K_\theta}O\left(\lambda_I^{\theta}\right) + c_{K_{\mathbf{x}}}O\left(\lambda_J^{\mathbf{x}}\right)$$

Similar interpretation holds for this bound as those from the other chapters. See Izbicki et al. (2014) for additional details and proofs, which follow along the same lines of those presented in the Appendix of this thesis.

6.4 NUMERICAL EXPERIMENTS

The general setup for using the estimator is as follows. We have data which are modeled as an i.i.d. sample $\mathbf{x}_1, \ldots, \mathbf{x}_m$ from the distribution $f(\mathbf{x}|\theta)$. Our goal is to infer the value θ. Although we are able to simulate from $f(\mathbf{x}|\theta)$ for fixed θ, we lack an analytical form for the likelihood function. Hence, we use the methodology of Section 6.2 to estimate $\mathcal{L}(\mathbf{x};\theta)$ from a *simulated sample*. Once we have an estimate $\widehat{\mathcal{L}}(\mathbf{x};\theta)$, we can approximate the likelihood of an *observed sample* according to

$$\widehat{\mathcal{L}}\left((\mathbf{x}_1, \ldots, \mathbf{x}_m);\theta\right) = \prod_{k=1}^{m} \widehat{\mathcal{L}}(\mathbf{x}_i;\theta).$$

This approximation can then be used in likelihood-based inference by, for example, plugging the expression into Bayes Theorem or by finding the maximum likelihood estimate.

In what follows we present five numerical examples where the ambient dimensionality of $\mathbf{x}$ is larger than its intrinsic dimensionality. In all experiments, we choose a uniform prior distribution on the parameter space.

Spiral. The data are i.i.d. observations of $(X^{(1)}, X^{(2)})$, where

$$X^{(1)} = \theta\cos\theta + N(0,1) \text{ and } X^{(2)} = \theta\sin\theta + N(0,1)$$

for $0 < \theta < 15$. Although the dimension of the sample space is 2, the data lie close to a one-dimensional spiral.

Klein Bottle. The data are i.i.d. observations of $(X^{(1)}, X^{(2)}, X^{(3)}, X^{(4)})$, where

$$\begin{cases} X^{(1)} = 2(\cos\theta_2 + 1)\cos\theta_1 + N(0,1) \\ X^{(2)} = 2(\cos\theta_2 + 1)\sin\theta_1 + N(0,1) \\ X^{(3)} = 2\sin\theta_2\cos\theta_1/2 + N(0,1) \\ X^{(4)} = 2\sin\theta_2\sin\theta_1/2 + N(0,1) \end{cases}$$

for $0 < \theta_1, \theta_2 < 2\pi$. The dimension of the sample is 4, but the data lie close to a two-dimensional Klein Bottle embedded in $\mathbb{R}^4$.

Transformed Images. In this example, we rotate and translate an image of a tiger, see the top row of Figure 6.1. The model parameters are (θ, ρ_x, ρ_y). The transformed images are centered at $(\rho_x + N_T(0, 10), \rho_y + N_T(0, 10))$[3] with rotation angle $(\theta + N(0, 10))$. The final images are cropped to 20×20 pixels, i.e., the sample space has dimension 400.

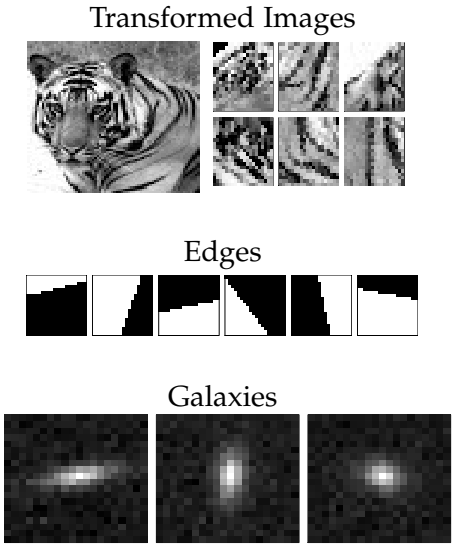


Figure 6.1: Some examples of data generated according to Section 6.4. (The top left image is the original image in "Transformed Images".)

Edges. Here we generate 20×20 images of binary edges from a model with two parameters, α and λ. The data are i.i.d. observations of an edge with rotation angle $\alpha + N(0, \pi/4)$ and displacement $\lambda + N_T(0, 0.5)$ from the center, see Figure 6.1 for some examples.

Simulated Galaxy Images. The last example is a simplified version of a key estimation problem in astronomy, namely that of shear estimation (Bridle et al. 2009). We use the GalSim Toolkit[4] to simulate realistic galaxy images. We sample two parameters: First, the orientation with respect to the x-axis of the image and, second, the axis ratio of the galaxies, which measures their ellipticity. To mimic a realistic situation, the observed data are low-resolution images of size 20×20.

3 N_T is the truncated normal to guarantee that the parameters are in the range of the image.
4 https://github.com/GalSim-developers/GalSim

Figure 6.1, bottom, shows some examples. These images have been degraded by observational effects such as background noise, pixelization, and blurring due to the atmosphere and telescope; see Figure 6.2

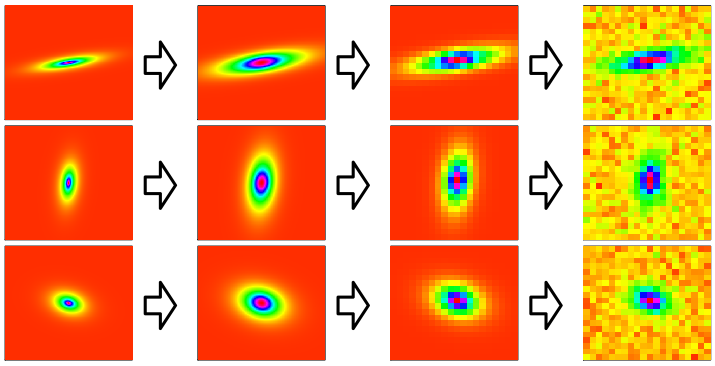


Figure 6.2: Examples of galaxies with different orientations and axis ratios. From left to right: High-resolution, uncontaminated galaxy image; effect of PSF caused by atmosphere and telescope; pixelated image; and observed image containing additional Poisson noise. We only observe images on the right.

We assume that the orientation and axis ratio of galaxy i are given by

$$a_i \sim \text{Laplace}(\alpha, 10) \text{ and } r_i \sim N_T(\rho, 0.1^2),$$

respectively. We seek to infer $\theta = (\alpha, \rho)$ based on an observed i.i.d. sample of images $\mathbf{x}$ contaminated by observational effects. Notice we do not observe a_i and r_i, but only $\mathbf{x}_i$, the 400-dimensional noisy image.

Methods. In all examples, the likelihood function is estimated based on $n_F = n_G = 5,000$ observations from the simulation model; 60% of the data are used for training and 40% for validation. We compare *Series*, our spectral series estimator from Section 6.2, with two state-of-the-art estimators of $f(\mathbf{x}|\theta)$. The first estimator is *KDE* – a kernel density estimator based on taking the ratio of kernel estimates of $f(\mathbf{x}, \theta)$ and $f(\theta)$. We use the implementation from the package "np" (Hayfield and Racine 2008) for R. For the Spiral and Klein bottle examples, we select the bandwidths of KDE via cross-validation. However, the high dimension of the other ex-

amples (Transformed, Edges, and Galaxy) makes a cross-validation approach computationally intractable and the density estimates numerically unstable. For these examples, we instead use the default reference rule for the bandwidth, and we reduce the dimensionality of the data with PCA with number of components chosen by minimizing the estimated loss (Eq. 6.4). The second estimator in our comparisons is *LS* – the direct least-squares conditional density estimator of Sugiyama et al. (2010b). This estimator is based on a direct expansion of the likelihood into a set of prespecified functions. This approach typically yields better results than estimators based on the ratio of random variables. Again, to avoid the problem of high dimensionality in the examples with $d > 4$, we also implement *PCA+LS*, the direct least-squares conditional density estimator after dimension reduction via PCA, with number of components chosen so as to minimize the estimated loss. PCA has the additional goal of decorrelating adjacent pixels in the images examples.

Table 6.1: Estimated L^2 loss (with standard errors) of the likelihood function estimators. Best-performing models with smallest loss are in bold fonts.

DATA	**DIM.**	**L^2 LOSS**				
		Series	*LS*	*PCA+LS*	*KDE*	*PCA+KDE*
Spiral	2	7.13 (0.14)	6.61 (0.12)	—	**2.95 (0.30)**	—
Klein Bottle	4	**1.45 (0.07)**	2.02 (0.06)	—	1.68 (0.11)	—
Transf. Images	400	**20.94 (0.03)**	26.91 (0.04)	27.12 (0.03)	—	26.62 (0.06)
Edges	400	**0.70 (0.03)**	1.77 (0.02)	1.55 (0.03)	—	1.60(0.02)
Galaxy Images	400	**40.94 (0.03)**	42.57 (0.01)	42.53 (0.01)	—	43.99 (0.04)

Results. In Tables 6.1 and 6.2, we present the estimated L^2 loss (Eq. 6.4), as well as the estimated average likelihood $\mathbb{E}_{(\mathbf{X},\theta)}[\widehat{\mathcal{L}}(\mathbf{X};\theta)]$ based on a test set with 3,000

Table 6.2: Estimated average likelihood (with standard errors) of the likelihood function estimators. Best-performing models with largest average likelihood are in bold fonts.

DATA	**DIM.**	**AVERAGE LIKELIHOOD**				
		Series	*LS*	*PCA+LS*	*KDE*	*PCA+KDE*
Spiral	2	16.54 (0.16)	19.49 (0.14)	—	**28.62 (0.01)**	—
Klein Bottle	4	5.62 (0.08)	4.96 (0.08)	—	**5.63 (0.13)**	—
Transf. Images	400	**8.31 (0.03)**	1.83 (0.03)	1.08 (0.02)	—	1.58 (0.06)
Edges	400	**3.69 (0.04)**	1.72 (0.02)	2.55 (0.03)	—	2.10 (0.02)
Galaxy Images	400	**4.63 (0.04)**	2.24 (0.01)	2.43 (0.02)	—	1.01 (0.04)

observations[5]. Both measures indicate that, while traditional methods have better performance in low dimensions, our spectral series method yields substantial improvements when the ambient dimensionality of the sample space is large. Note that even after dimension reduction, *LS* does not yield the same performance as *Series*. In fact, in some cases, a dimension reduction via PCA leads to less accurate estimates.

As a further illustration, Figure 6.3 shows the estimated likelihood function for samples of size $m = 10, 20, 30$ and 50 drawn from the galaxy image model with parameters $\alpha = 80^\circ$ and $\rho = 0.2$ (recall m is the observed data sample size). For comparison, we also include the true likelihood function (TRUTH), which is unavailable in practical applications[6]. It is apparent from the figure that the spectral series estimator comes closer to the truth than the other estimators, even without reducing the dimensionality of the galaxy images.

Furthermore, to quantify how the level sets of the likelihood function concentrate around the true parameters, we define the expected average distance of the es-

5 To make results comparable, we renormalize the estimated likelihood functions to integrate to 1 in θ.

6 Because the observed images are simulated, we know the true values of a_i and r_i.

timated likelihood function to the real parameter value, $\mathbb{E}_{\mathbf{x},\theta^*}\left[\int d(\theta^*,\theta)\widehat{\mathcal{L}}(\mathbf{x};\theta)d\theta\right]$[7], where the expectation is taken with respect to both θ^* and the observed data. Here we choose $d(\theta^*,\theta)$ to be the Euclidean distance between the vectors of parameters, standardized so that each component has minimum 0 and maximum 1.

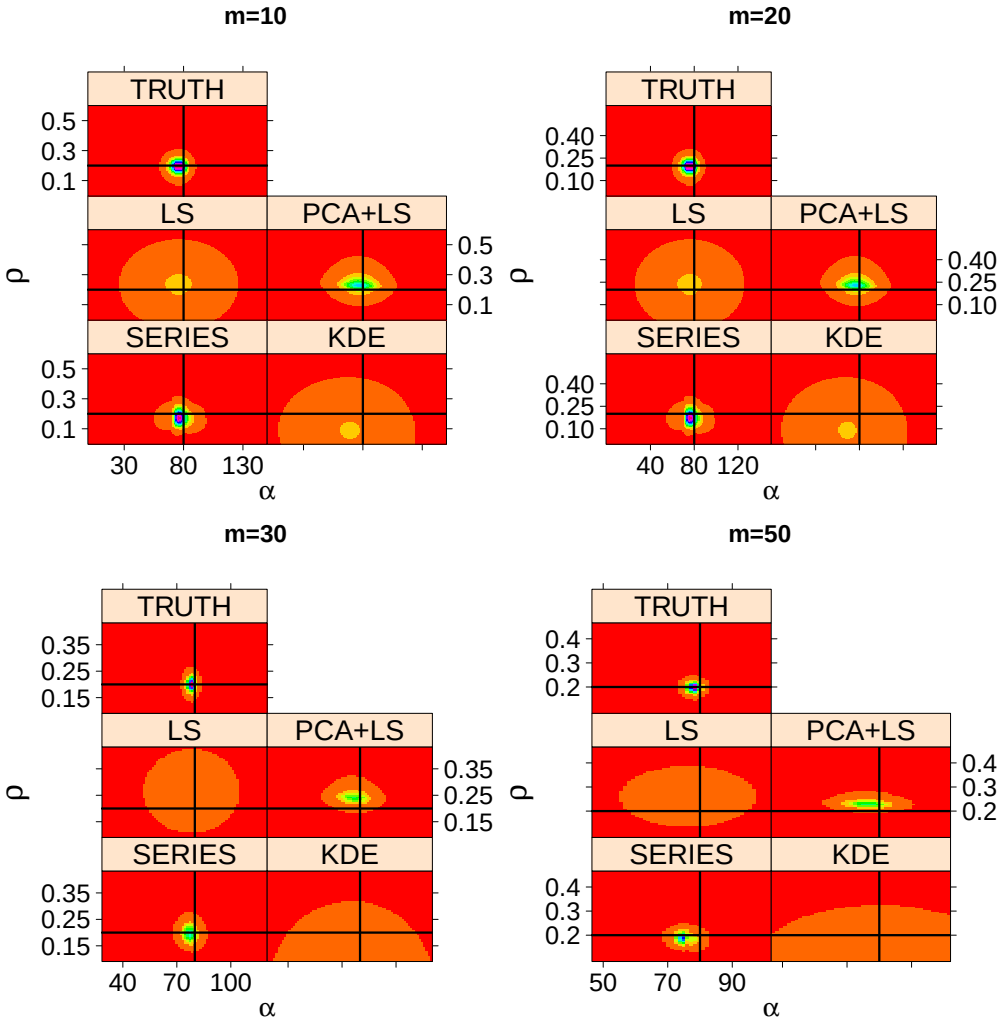


Figure 6.3: Comparison of level sets of estimated likelihood function $\mathcal{L}(\mathbf{x};(\alpha,\rho))$ for the galaxy example for 4 samples sizes. Horizontal and vertical lines are the true values of the parameters. In all cases, the spectral series estimator gets closer to the real distribution, which is uncomputable in practice.

7 As the prior distribution is uniform, this quantity is $\mathbb{E}_{\mathbf{x},\theta^*}\left[\int d(\theta^*,\theta)d\widehat{f}(\theta|\mathbf{x})\right]$.

As a final comparison of methods, we study how the above likelihood metric changes as a function of the sample size m of the observed data (the sample sizes n_F and n_G of the simulated data used to estimate the likelihood are held constant); see Figure 6.4 for results. Because $\mathcal{L}(\mathbf{x}; \theta)$ concentrates around the true parameter value θ^* for large sample sizes, we expect the average likelihood to decrease as m increases – if the likelihood estimates are reasonable. Indeed, we observe this behavior for all methods in the comparison for the problems with low dimensionality. However, for the problems with high dimensionality, this is no longer the case. On the other hand, the results indicate that *Series* is able to overcome the curse of dimensionality and recover the true θ^* parameter as the number of observations increases.

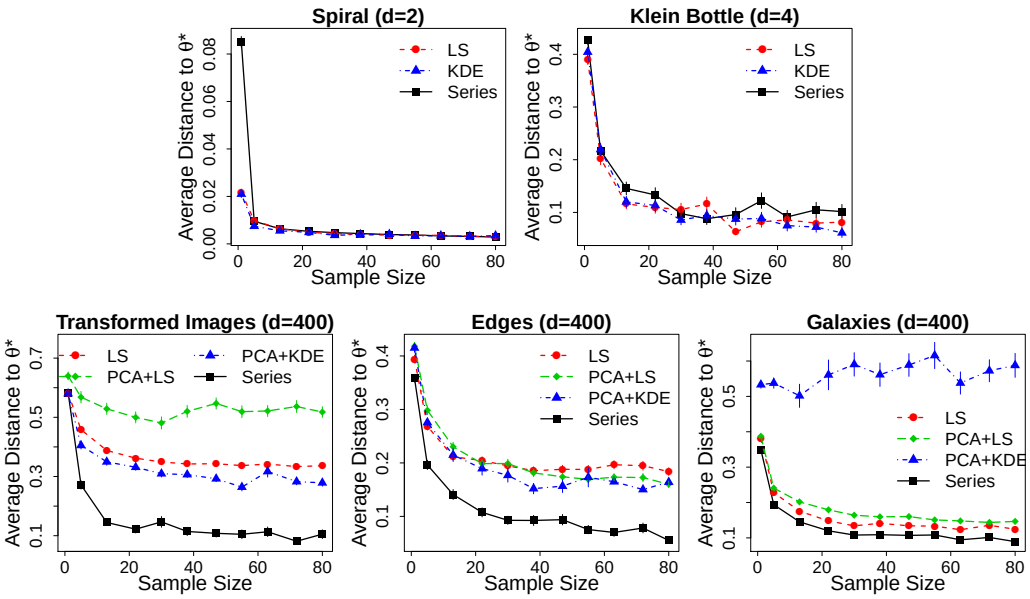


Figure 6.4: Average distance of estimated likelihoods to the true θ (and standard errors) as a function of the number of observed images for the galaxy data. While in low dimensions all estimators have similar performance, our approach performs better in high dimensions.

6.5 Future Work

In this chapter we saw the spectral series estimator of a likelihood function yields better estimates than other likelihood function estimators in high dimensions. We now describe some future directions that can be taken on this problem, which we plan to explore in a forthcoming paper.

6.5.1 *Non i.i.d. Data*

It would be desirable to extend the method to non i.i.d. data. If one is willing to use summary statistics as in standard ABC, this can be done by estimating the likelihood function $\mathcal{L}(\mathbf{s}(\mathbf{x}); \theta)$, where $\mathbf{s}(\mathbf{x}) = (s_1(\mathbf{x}), \ldots, s_p(\mathbf{x}))$ is a vector of summary statistics. In this framework, the kernel K would then measure the similarity between such vectors, $K(\mathbf{s}(\mathbf{x}), \mathbf{s}(\mathbf{y}))$.

6.5.2 *Likelihood Estimation versus ABC*

Some advantages of using conditional density estimation over traditional ABC can be explored. For example, framing the likelihood-free inference problem as a CDE problem allows one to select a subset from a collection of possible summary statistics. This can be done in the same fashion as we describe in the photometric redshift prediction problem in Chapter 7.

Another interesting feature is that one can take advantage of the fact that there is a natural way to check goodness-of-fit of the estimated likelihood function. More specifically, it is possible to compute p-values for the hypothesis that the estimated and the true likelihood functions are the same by using the following procedure:

1. Sample $\theta \sim \pi(\theta)$.
2. Sample $\mathbf{x}_1^\theta, \ldots, \mathbf{x}_n^\theta$ from $f(\mathbf{x}|\theta)$.

3. Using a rejection sampler, sample $\mathbf{x}_1^*, \ldots, \mathbf{x}_n^*$ from $f(\mathbf{x}|\theta)$ *under the hypothesis that* $\widehat{\mathcal{L}}(\mathbf{x}|\theta) = \mathcal{L}(\mathbf{x}|\theta)$. This can be done by repeating the following step until obtaining a sample of size n:
   - Let $M \equiv \sup_{\mathbf{x}} \widehat{\mathcal{L}}(\mathbf{x}|\theta)$.
   - Sample $\mathbf{x}_i \sim g(\mathbf{x})$, accept it with probability $\widehat{\mathcal{L}}(\mathbf{x}|\theta)/M$.
4. Calculate the p-value of the hypothesis that $\mathbf{x}_1^\theta, \ldots, \mathbf{x}_n^\theta$ comes from the same distribution as $\mathbf{x}_1^*, \ldots, \mathbf{x}_n^*$ (using, e.g., Gretton et al. 2007).

One might than test whether the collection of p-values obtained for all θ's comes from a uniform distribution. Notice that defining the likelihood function as the ratio $f(\mathbf{x}|\theta)/g(\mathbf{x})$ is the key for being able to sample from the conditional density. It would be interesting to test if such method has reasonable performance in practice; for example, if it has good power properties.

Finally, it would be desirable to systematically compare the performance of the spectral series approach – both computational and statistical – with traditional ABC methods, including MCMC-ABC (Marjoram et al. 2003) and SMC-ABC (Sisson et al. 2007).

6.5.3 *Likelihood Estimation via Regression*

Because in most problems parameters are not expected to lie on a lower-dimensional space, estimating the basis for θ in the likelihood function estimator of Equation 6.3 may be an unnecessary step. A possible way of avoiding this is to fix $\theta_0 \in \Theta$, and expand the likelihood function in the basis $(\psi_j(\mathbf{x}))_j$. The expansion is given by

$$\mathcal{L}(\mathbf{x}; \theta_0) = \sum_{j \geqslant 1} \beta_j^{\theta_0} \psi_j(\mathbf{x}),$$

where

$$\beta_j^{\theta_0} = \int_{\mathcal{X}} \psi_j(\mathbf{x}) \mathcal{L}(\mathbf{x}; \theta_0) \mathrm{d}P(\mathbf{x}) = \int_{\mathcal{X}} \psi_j(\mathbf{x}) \frac{f(\mathbf{x}|\theta_0)}{g(\mathbf{x})} \mathrm{d}P(\mathbf{x}) = \mathbb{E}[\psi_j(\mathbf{X})|\theta_0]$$

This indicates that the coefficients may be estimated by regressing $\psi_j(\mathbf{X})$ on θ using the simulated sample $(\theta_1, \mathbf{x}_1), \ldots, (\theta_n, \mathbf{x}_n)$.

A second way of avoiding estimation of this basis is to use, for example, the basis proposed by Sugiyama et al. (2010b) for θ together with spectral series for $\mathbf{x}$. This would require, however, new methods for computing the estimator efficiently, because such basis is not orthogonal with respect to $\pi(\theta)$. Alternatively, if θ is small-dimensional, one may use tensor products of Fourier bases.

Part III

MAIN APPLICATION

7

PHOTOMETRIC REDSHIFT PREDICTION UNDER SELECTION BIAS

7.1 Introduction

Technological advances over the last two decades have ushered in the era of "precision cosmology," with the construction of catalogs that contain data on upwards of 10^8 galaxies (e.g., Aihara et al. 2011). Cosmologists use these data to place progressively tighter constraints on the parameters of the ΛCDM model, the leading model explaining the structure and evolution of the Universe (see, e.g., Springel et al. 2006). This data flood will only intensify in the next two decades: for instance, the Large Synoptic Survey Telescope is expected to collect data for up to 10^{10} galaxies during its ten years of observations that will begin c. 2020 (Ivezić et al. 2008).

To constrain cosmological parameters, cosmologists need to estimate galaxy *redshifts*, a proxy for distance that may be precisely estimated via *spectroscopic* data. Unfortunately, spectroscopy is resource intensive, and is generally only applied to the brightest galaxies in a survey. As a result, more than 99 percent of current galaxy observations are done via *photometry*, essentially a low-resolution spectroscopy. Photometry, however, does not allow redshifts to be estimated with as much precision as spectroscopy does, see Section 7.2 for more details.

The goal of *photometric redshift estimation* (or photo-z estimation) is to construct probability density functions (pdfs) for ensembles of galaxies, conditional upon their photometric covariates. Cosmologists use the resulting ensemble of pdfs to inform cosmological analyses. Initially, they viewed photo-z estimation as a regression problem in which one sought $\mathbb{E}[Z|\mathbf{x}]$, where $\mathbf{x}$ is the vector of photometric covariates of the galaxy and Z represents its redshift. However, some have come to realize the importance of estimating $f(z|\mathbf{x})$, the conditional density of the redshift based on photometric magnitudes: $f(z|\mathbf{x})$ can be asymmetrical and multi-modal, thus making $\mathbb{E}[Z|\mathbf{x}]$ not effective in summarizing $f(z|\mathbf{x})$ (see, e.g., Ball and Brunner 2010 and Wittman 2009). Using an estimate $\widehat{f}(z|\mathbf{x})$ can substantially improve cosmological analyses based on redshift estimates (e.g., Wittman 2009).

There are two ways by which cosmologists estimate $f(z|\mathbf{x})$. In the first method, *template fitting* (e.g., Fernández-Soto et al. 1998), one estimates $f(z|\mathbf{x})$ for a given galaxy by directly comparing its data with a suite of idealized photometric data sets for different types of galaxies at different redshifts. In this chapter, our interest lies in the second method, *empirical redshift estimation*. In this method, one uses machine learning techniques to train an estimator of $f(z|\mathbf{x})$ utilizing spectroscopically derived redshifts for galaxies and their associated photometric data (see, e.g., Ball and Brunner 2010; Zheng and Zhang 2012; Kind and Brunner 2013).

Standard machine learning methods should not be naively used on such data: Because spectroscopy can generally be applied to only the brightest galaxies in a survey, in general it is not reasonable to assume photometric data are identically distributed to the spectroscopic data. That is, there is a selection bias that favors spectroscopic data sets with less faint samples than the photometric data sets (see, e.g., Oyaizu et al. 2008; Ball and Brunner 2010, and Figure 7.1). Although not taking into account selection bias may substantially decrease the performance of estimators and artificially diminish their nominal errors, few works on redshift prediction deal with this issue explicitly. One exception is the estimator of $f(z|\mathbf{x})$ developed

by Cunha et al. (2009). The development of improved methods for mitigating this problem is the subject of this chapter.

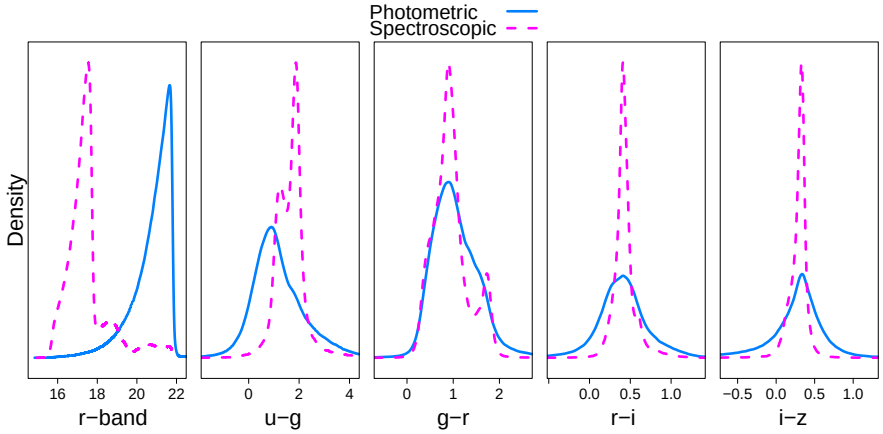


Figure 7.1: Distribution of r-band `model` magnitudes (upper left) and the four colors (i.e., differences of the `model` magnitudes in adjacent photometric bins) for spectroscopic and photometric SDSS data sets. For more details, see Section 7.2.

Contribution

In this chapter, we introduce a statistically rigorous framework for estimating photometric redshift distributions under selection bias, which more generally can be applied to other regression and conditional density estimation problems where there is a difference in how training and target data are selected. We design appropriate loss functions for the selection bias setting, and show how these can be estimated using a spectroscopic (training) and a photometric (target) sample (eqs. 7.5 and 7.9). The set-up allows for a principled way of choosing tuning parameters, comparing and combining different conditional density estimators, as well as performing variable selection. Here, we describe in detail how to go about each of these problems and suggest non-parametric procedures that are both flexible and practical for large databases.

In particular, we propose two alternative non-parametric estimators to the conditional density estimator of Cunha et al. (2009). We compare our new estimators to that of Cunha et al. (2009) using SDSS data and demonstrate that they yield im-

proved results. Moreover, we introduce a principled method of combining two or more conditional density estimators for optimal performance under different sampling schemes. We compare various methods for estimating importance weights – an important quantity used to correct methods so that they work under selection bias – and describe how to test the goodness-of-fit of the final density estimates.

The organization of this chapter is as follows. In Section 7.2 we describe the data that we use to assess previous and new methodologies, and in Section 7.3, we lay out the statistical problem of density estimation under selection bias. We explain how to match training and target samples using importance weights, and we compare different schemes for estimating these weights. In Section 7.4, we shift to the problem of constructing conditional density estimators under selection bias. We describe how to use the estimated importance weights to build an appropriate loss function for evaluating conditional density estimators under selection bias, and we propose and compare different density estimators designed for this setting. We also discuss variable selection, and how to evaluate goodness-of-fit of density estimates. In Section 7.5 we show an application of our methods to the problem of galaxy-galaxy lensing. Finally, in Section 7.6, we summarize and discuss the main results.

7.2 DATA

Redshift estimators are built using two types of data: *spectroscopic* data, where both the covariates $\mathbf{x}$ and the labels z are known, and *photometric* data, where only the covariates $\mathbf{x}$ are known and which is where the estimated $f(z|\mathbf{x})$'s will be applied. Below we describe the data used to test our methods, the *SDSS photometric sample*, the *spectroscopic sample*, and the *simulated photometric samples*.

SDSS photometric sample. To construct our photometric data set, we use Sloan Digital Sky Survey (SDSS; York et al. 2000). Since 2000, SDSS has utilized a dedicated 2.5-meter telescope at Apache Point, New Mexico, to collect data on over 200

million galaxies that are spread over nearly one-quarter of the sky. We extract SDSS DR8 (Aihara et al. 2011) galaxies from from a ≈72 square-degree patch of sky (RA $\in [168°,192°]$ and $\delta \in [-1.5°,1.5°]$), and filter the data according to the prescription of Sheldon et al. (2012). Our filtered data set numbers 538,974 galaxies.

In photometry, as image data are collected, different photometric filters are sequentially placed into the telescope's light path, with each one allowing only those photons in a specific wavelength band to pass through. The five SDSS bands–denoted u, g, r, i, and z–span the range of wavelengths from 3.5×10^{-7} meters (ultraviolet light) through the optical regime to 9×10^{-7} meters (infrared light). Within each image are sources that are detected and classified via pipeline software. *Photometric fluxes*–i.e., the galaxy fluxes in each of the ugriz bands–are then estimated for each detected galaxy by effectively drawing a boundary around it and summing the light intensity within that boundary. There are several boundary-definition algorithms or *magnitude systems* used in SDSS pipeline processing; in this work, we follow Sheldon et al. (2012) and use as our covariates the eight colors estimated using the `model` and `cmodel` algorithms. The logarithm of the ratios of fluxes in adjacent bands are dubbed *photometric colors* and are the covariates we use for the analysis. We also use the raw r magnitude (i.e., the logarithm of the flux in r band) in both `model` and `cmodel` measurements. We scale the 10 covariates so that they have mean 0 and standard deviation 1.

Spectroscopic sample. As mentioned above, SDSS has collected photometric data for over 200 million galaxies. Of these, some one million have been the subject of follow-up *spectroscopic* observations, in which fiber optics are used to redirect a galaxy's light to two spectrographs that are sensitive to blue and red light, respectively. Via pipeline processing, the dispersed light is mapped to ≈3,500 pixels, allowing emission and absorption spikes that are smeared out by photometry to be finely resolved. These spikes, which are caused by upward (absorption) or downward (emission) transitions of electrons between atomic energy levels, occur at known wavelengths. One can use the wavelength ratios for two or more ob-

served spikes to infer which transitions they represent; once that information is known, redshift estimation is trivial. Because the observed spikes are generally narrow and numerous, the typical precision in spectroscopic redshift estimation is $\Delta z/z \sim 10^{-6}$, i.e., we may safely make the simplifying assumption that for spectroscopic redshifts, there is no measurement error.

We follow Sheldon et al. (2012), and use a spectroscopic data (i.e., colors and redshifts) from a variety of sources (but mostly from SDSS Data Release 8). We use Sheldon et al.'s spectroscopic data set directly (435,875 galaxies; E. Sheldon, private communication). The goal of using additional spectroscopic sources besides SDSS is to have fainter galaxies on the labeled sample, and hence cover the same region of observables from the photometric sample.

Simulated photometric samples. In addition to the photometric sample from SDSS, we also use the spectroscopic data to create *simulated* photometric sets. This is done because (i) it provides a wider range of data sets for comparing the methods (ii) contrary to the real SDSS photometric set, the redshifts for such samples are known, allowing more thorough comparisons, and (iii) it is possible to study robustness to selection bias of different methods.

More specifically, we create a photo-z prediction setting under 3 different scenarios – with no, moderate and strong selection bias, respectively. We use rejection sampling to construct these sets that are shifted relative to the spectroscopic sample. More precisely: Let x^r denote the `r model` magnitude, scaled to be between 0 and 1. The larger the `r model` magnitude, the fainter the galaxy. We assume that $p(\mathbf{x})$, the probability that a data point in the spectroscopic sample is also included in the unlabeled data, depends on $\mathbf{x}$ through x^r only; i.e. $p(\mathbf{x}) = p(x^r)$. We consider three different sampling schemes:

Scheme 1: $p(x^r) \propto \mathtt{Beta}(1,1) \equiv \mathtt{Unif}(0,1)$,

Scheme 2: $p(x^r) \propto \mathtt{Beta}(13,4)$,

Scheme 3: $p(x^r) \propto \texttt{Beta}(18, 4)$.

Figure 7.2 shows the resulting distributions of the r-band `model` magnitude for labeled and unlabeled data. The amount of covariate shift corresponds to what one might observe in practice for astronomical surveys. Scheme 1 corresponds to no selection bias, with labeled and unlabeled data following the same distribution. In contrast, for Scheme 3, there is a strong selection bias with a large proportion of the galaxies in the unlabeled set being significantly fainter (shifted toward large r-band magnitude) than the galaxies in the labeled set. Sampling scheme 2 represents a case in-between 1 and 3.

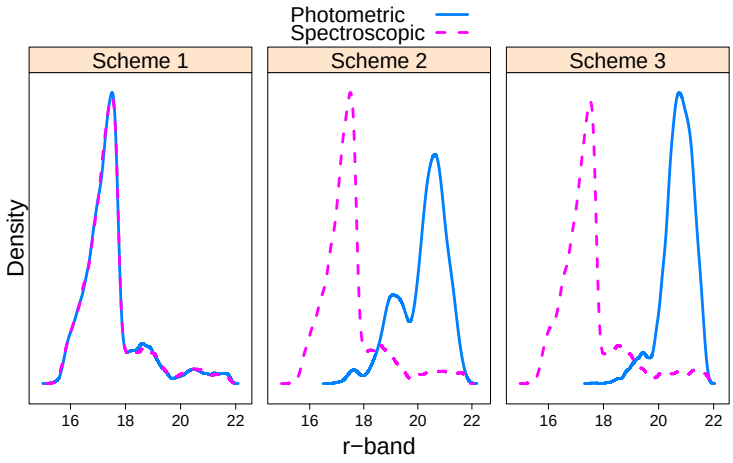


Figure 7.2: Distribution of r-band `model` magnitude under three different sampling schemes of spectroscopic data.

7.3 Selection Bias, Covariate Shift and Importance Weights

A popular method for correcting for selection bias in *regression* and *classification* is by sample reweighting. In this section, we define our notation, explain the main ideas behind importance weights, and provide a framework for comparing different methods for estimating these weights.

7.3.1 *Problem Formulation*

Let $\mathbf{x} \in \mathbb{R}^d$ denote the covariates of interest — in our application, the photometric colors, magnitudes, and other observables. Let $z \in \mathbb{R}$ denote the redshift of a galaxy. Without loss of generality, we assume $z \in [0, 1]$. Suppose we observe an i.i.d. sample $(\mathbf{x}^L_1, z^L_1), \ldots, (\mathbf{x}^L_{n_L}, z^L_{n_L})$ with labels from spectroscopic measurements and an i.i.d. unlabeled sample $\mathbf{x}^U_1, \ldots, \mathbf{x}^U_{n_U}$ with only photometric data (also known as target data). Our goal is to build an estimator $f(z|\mathbf{x})$ that has good performance on the unlabeled target data. Although the standard assumption in machine learning is that labeled and unlabeled data have similar distributions, we saw in Figure 7.1 that this is not the case in redshift surveys.

Let P_L denote the distribution on the labeled sample and let P_U denote the distribution on the unlabeled sample, i.e., $(\mathbf{x}^L, z^L) \sim P_L$ and $(\mathbf{x}^U, z^U) \sim P_U$ (where the z^U's are never observed in practice). In the machine learning literature on *data set shift* (Quionero-Candela et al. 2009), the aim is to understand how a $P_L \neq P_U$ shift affects learning algorithms and to design classifiers that perform well for the target distribution P_U. This is a hopeless problem if P_L has no relationship to P_U (Gretton et al. 2010). Hence, some assumptions have to be made regarding how P_L and P_U are related. Different assumptions have been proposed in the literature for different types of data set shift; see, e.g., Quionero-Candela et al. (2009) and Moreno-Torres et al. (2012) for an overview. In our photo-z application, it is often reasonable to assume that whether a sample is spectroscopically labeled or not only depends on observable covariates, such as the photometric colors in a given magnitude system (Lima et al. 2008; Sheldon et al. 2012). In other words, the assumption is that

$$P(S = 1|\mathbf{x}, z) = P(S = 1|\mathbf{x}), \tag{7.1}$$

where $\mathbf{x}$ denotes the observed covariates, and S is a random variable that assumes the value 1 if a given datum is labeled and 0 otherwise. In the statistics literature

(Rubin 1976; Moreno-Torres et al. 2012), this condition is commonly referred to as *missing at random* (MAR). MAR bias implies *covariate shift*, defined as

$$f_L(\mathbf{x}) \neq f_U(\mathbf{x}), \quad f_L(z|\mathbf{x}) = f_U(z|\mathbf{x}). \tag{7.2}$$

Under certain conditions on the support of $f_L(\mathbf{x})$ and $f_U(\mathbf{x})$, covariate shift also implies MAR bias (Moreno-Torres et al. 2012). Here we use both terms interchangeably to refer to the assumption in Equation 7.1.

At a first glance, it may seem that MAR bias would not pose a problem for density estimation: Because $f(z|\mathbf{x})$ is the same for both labeled and unlabeled samples, it appears reasonable that a good estimator of $f(z|\mathbf{x})$ constructed using labeled data also has good performance for unlabeled data. But this is generally not true — because the loss function for estimating $f(z|\mathbf{x})$ as well as other quantities depends on the marginal distribution of $\mathbf{x}$. In other words: An estimator that is good with respect to $f_L(\mathbf{x})$ might not be good with respect to $f_U(\mathbf{x})$. We will return to this point in Section 7.3.2.

In non-parametric regression and classification, covariate shift is sometimes corrected by the so-called *importance weights* $\beta(\mathbf{x}) := f_U(\mathbf{x})/f_L(\mathbf{x})$[1] (Sugiyama et al. 2008). In what follows (Section 7.3.2), we investigate the problem of how to best estimate importance weights from data. Then (in Section 7.4), we explain how to incorporate importance weights into nonparametric conditional density estimation.

7.3.2 *Estimating Importance Weights*

In Chapter 5 we described some of the most common approaches of estimating the importance weights $\beta(\mathbf{x})$. In astronomy, researchers have explored nearest

1 For the problem to be well-defined, $P_L(\mathbf{x})$ has to *dominate* $P_U(\mathbf{x})$, i.e., $P_L(\mathbf{x}) \gg P_U(\mathbf{x})$. This assumption is necessary but does not alway hold in practice. In our work, we choose the photometric data so that they have covariates in the same domain as the spectroscopic data (Section 7.2). A related and complementary approach (Ball and Brunner 2010) is to first apply machine-learning techniques to the unlabeled data that fall in the restricted domain and then extrapolate to data outside this region by template methods.

neighbor techniques for reweighting non-representative training samples. Lima et al. (2008) and Cunha et al. (2009) suggest estimating $\beta(\mathbf{x})$ with the following nearest-neighbor estimator (which we denote by β-*NN*):

$$\widehat{\beta}(\mathbf{x}) = \frac{1}{N}\frac{n_L}{n_U}\sum_{k=1}^{n_U} \mathbb{I}\left(\mathbf{x}_k^U \in V_{\mathbf{x}}^N\right), \tag{7.3}$$

where

$$V_{\mathbf{x}}^N = \{\mathbf{y} \in \mathbb{R}^d : d(\mathbf{y},\mathbf{x}) \leqslant d(\mathbf{x}_{(N)}^L, \mathbf{x})\}$$

is the region of the feature space with points closer to $\mathbf{x}$ than $\mathbf{x}_{(N)}^L$, the Nth nearest neighbor of $\mathbf{x}$ in the labeled data.

Our ultimate goal is good photo-z prediction but it can be tricky to choose the best method for estimating importance weights. The empirical loss of importance-weighted learning algorithms depends on the accuracy of these estimates, i.e., on the values $\widehat{\beta}(\mathbf{x})$ themselves. As a result, one cannot simply choose the weighting method that minimizes the empirical error of the final density estimates $\widehat{f}(z|\mathbf{x})$. To address this problem, here we employ a two-step approach: First, we select the best estimator of $\beta(\mathbf{x})$ using an appropriate loss (which is the topic of this section) and then we use the estimated weights to select the best conditional density estimator. As we shall see in Section 7.4, for importance-weighted statistical procedures, one needs to have accurate estimates of $\beta(\mathbf{x})$ *at the labeled points*, or more generally in the regions of the feature space where the density of labeled points is large. It follows that a natural loss function for a given estimator $\beta(\mathbf{x})$ is

$$\begin{aligned} L(\widehat{\beta}, \beta) :&= \int \left(\widehat{\beta}(\mathbf{x}) - \beta(\mathbf{x})\right)^2 dP_L(\mathbf{x}) \\ &= \int \widehat{\beta}^2(\mathbf{x}) dP_L(\mathbf{x}) - 2\int \widehat{\beta}(\mathbf{x}) dP_U(\mathbf{x}) + K \end{aligned} \tag{7.4}$$

where K is a constant that does not depend on $\widehat{\beta}(\mathbf{x})$. Notice this is the same loss as that we used for a ratio estimator in Section 5.2.1. Again, given a labeled validation

sample $\tilde{\mathbf{x}}_1^L, \ldots, \tilde{\mathbf{x}}_{\tilde{n}_L}^L$ and an unlabeled validation sample $\tilde{\mathbf{x}}_1^U, \ldots, \tilde{\mathbf{x}}_{\tilde{n}_U}^U$, we estimate $L(\widehat{\beta}, \beta)$ by

$$\widehat{L}(\widehat{\beta}, \beta) = \frac{1}{\tilde{n}_L} \sum_{k=1}^{\tilde{n}_L} \widehat{\beta}^2 \left(\tilde{\mathbf{x}}_k^L\right) - 2\frac{1}{\tilde{n}_U} \sum_{k=1}^{\tilde{n}_U} \widehat{\beta} \left(\tilde{\mathbf{x}}_k^U\right). \tag{7.5}$$

We choose the model that minimizes $L(\widehat{\beta}, \beta)$ on a validation set. This provides a principled and simple method for choosing tuning parameters in, for example, the nearest neighbor method by Lima et al. (2008) and Cunha et al. (2009).

7.3.3 *Variable Selection*

Although $\beta(\mathbf{x})$ may depend on all observable covariates $\mathbf{x}$, selecting a subset of covariates $\mathbf{x}_0$ to estimate it may produce better estimators because, as in standard nonparametric regression (Wasserman 2006), this will lead to estimators with less variance. Hence, variable selection can increase the performance of estimators of the weights $\beta(\mathbf{x})$

We propose to perform variable selection for the importance weights by first estimating $\beta(\mathbf{x})$ using different subsets of covariates of $\mathbf{x}$, and then, for each of these, estimating loss (7.4) using Equation 7.5. We then pick the subset that leads to the smallest estimated error. Because there are 2^{10} subsets of covariates (the four colors and r-band in both `model` and `cmodel` magnitude systems), we use a forward stepwise-type model search (Hastie et al. 2001), starting with the estimator with no covariates, $\widehat{\beta}(\mathbf{x}) \equiv 1$.

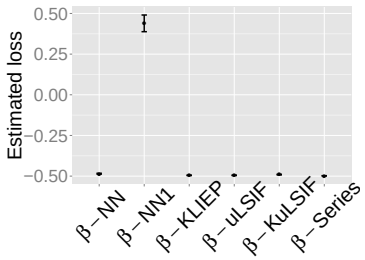
7.3.4 *Comparison of Estimators of* β

We now compare six estimators of importance weights: β-*NN* denotes the nearest neighbor estimator from Lima et al. (2008), but with the smoothing parameter N chosen so as to minimize our proposed empirical loss function (Equation 7.5); β-*NN1* is the nearest neighbor approach with $N = 1$ as in Loog (2012); β-*KLIEP* and β-*uLSIF* are the importance weight estimators suggested by Sugiyama et al. (2008) and Kanamori et al. (2009), respectively, implemented using MATLAB code

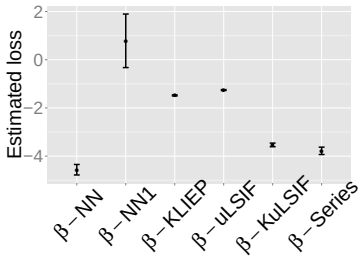
provided by the authors[2]. We also implement β-*KuLSIF*, a kernelized version of β-*uLSIF* (Kanamori et al. 2012), and β-*Series*, the spectral series estimator from Chapter 5.

Following Lima et al. (2008), our covariates are the colors and the r-band magnitude in the `model` magnitude system. We now present the results of the analyses, and defer the discussion to the end of the section.

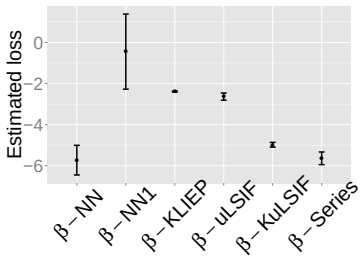
First, we use the simulated photo-z prediction setting to compare the performance of the importance weights estimators. We use 10,000 labeled samples, and an unlabeled sample of the same size. From each sample (labeled and unlabeled), we randomly chose 2,800 data points for training, 1,200 for validation and 6,000 for testing. Figure 7.3 summarizes the results. Using the variable selection technique from Section 7.3.3 on the nearest neighbors estimator leads us to choose the variables displayed in Table 7.1.



(a) Scheme 1



(b) Scheme 2



(c) Scheme 3

Figure 7.3: Comparison of different estimators of importance weights $\beta(\mathbf{x})$ for varying degrees of covariate shift. The plots display the estimated loss $\widehat{L}(\widehat{\beta}, \beta)$. Bars correspond to mean plus and minus standard error.

2 http://sugiyama-www.cs.titech.ac.jp/~sugi/software

We now perform the same analyses on the SDSS photometric set. As Figure 7.1 shows, the shift in the covariates is large. Therefore, about 80% of the spectroscopic samples have $\widehat{\beta}(\mathbf{x}) = 0$ (using β-*NN*). If these are not removed, they will cause the effective sample size to decrease substantially, and hence lead to bad conditional density estimates (see Section 7.4). In order to avoid this, we use a similar procedure as suggested by Lima et al. (2008). We first randomly select 15,000 spectroscopic and 15,000 photometric samples to estimate $\widehat{\beta}(\mathbf{x})$ using all `model` magnitude covariates. We then randomly select 15,000 additional spectroscopic samples in which $\widehat{\beta}(\mathbf{x}) \neq 0$. The analyses we present, unless mentioned otherwise, are based on this new spectroscopic sample and the original 15,000 photometric samples. Figure 7.4 shows the distribution of the covariates in `model` magnitude for the photometric and the spectroscopic sample. The shift is in fact smaller than in Figure 7.1. We use 3,500 samples of each data set for training, 1,500 for validation and the remaining for testing. Notice this is a different set from that used in Chapter 5, where we did not remove samples where $\widehat{\beta}(\mathbf{x}) = 0$.

Table 7.1: Selected covariates for importance weights estimators for each dataset (nearest neighbor estimator)

Dataset	`model`					`cmodel`				
	r	$u-g$	$g-r$	$r-i$	$i-z$	r	$u-g$	$g-r$	$r-i$	$i-z$
Scheme 1			X							
Scheme 2	X					X			X	
Scheme 3	X	X								
SDSS	X	X				X	X	X		

We first reestimate the importance weights in the new spectroscopic set. Figure 7.5 shows the estimated losses of the methods for estimation of $\beta(\mathbf{x})$ using all covariates from `model` magnitude. The chosen variables for the nearest neighbors estimator are displayed in the last row of Figure 7.1. The number of neighbors

chosen by minimizing Equation 7.5 was 8, not far from 5, the number used by Cunha et al. (2009). The loss of the final importance weight estimator is -2.41 ($\pm$ 0.08), smaller than -2.16 ($\pm$ 0.05) from the model using all 10 covariates, and -1.97 ($\pm$ 0.04) when using the 5 covariates from `model` magnitude only.

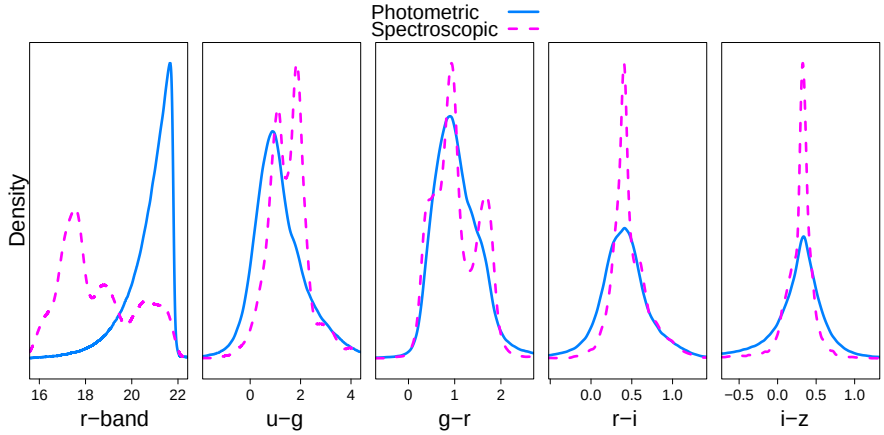


Figure 7.4: Distribution of r-band `model` magnitude and the 4 colors from `model` magnitude for the spectroscopic and photometric (SDSS) data sets *after* removing samples with initial estimated importance weight 0. Compare it to Figure 7.1; the distribution of the spectroscopic data indeed gets closer to that of the photometric sample.

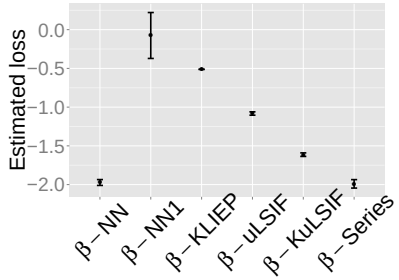


Figure 7.5: Estimated losses of the estimators of importance weights $\beta(\mathbf{x})$ for SDSS data using all 5 covariates from `model` magnitude. Bars correspond to mean plus and minus standard error.

Discussion. With cross-validated tuning parameters, β-*NN* and β-*Series* are the methods of choice for these data. They dominate the other estimators both for the SDSS photometric sample and for the three simulated selection bias schemes. In

some cases, β-*NN* is even better than β-*Series*[3]. This result is especially note-worthy as the nearest neighbor estimator has received little attention in the machine learning literature on data set shift.

Table 7.1 indicates our variable selection technique yields good results. In particular, for Scheme 1, where the weights do not depend on the covariate (they are constant), only one covariate was selected; for Schemes 2 and 3, where the weights depend only on `model` r-band, either this covariate or `cmodel` r-band was selected. This makes sense, because `model` r-band has 0.99 correlation with `cmodel` r-band. Moreover, the variable selection technique substantially increases the performance of the estimator of the importance weights.

Through the rest of the paper, we will use the β-*NN* method to estimate importance weights. The next section explains how one can use these weights to improve conditional density estimators under selection bias.

7.4 Conditional Density Estimation under Covariate Shift

We now switch the focus to the problem of estimating the conditional density $f(z|\mathbf{x})$, assuming an estimate of the weights $\beta(\mathbf{x})$ has already been computed.

Typically, conditional density estimators are designed to minimize the loss

$$\iint \left(\widehat{f}(z|\mathbf{x}) - f(z|\mathbf{x})\right)^2 dP_L(\mathbf{x})dz \tag{7.6}$$

where there is an implicit assumption that $P_L = P_U$. With labeled data, one can directly estimate this loss (up to a constant) according to Chapter 4:

$$\frac{1}{\widetilde{n}_L}\sum_{k=1}^{\widetilde{n}_L}\int \widehat{f}^2\left(z|\widetilde{\mathbf{x}}_k^L\right) dz - 2\frac{1}{\widetilde{n}_L}\sum_{k=1}^{\widetilde{n}_L}\widehat{f}\left(\widetilde{z}_k^L|\widetilde{\mathbf{x}}_k^L\right). \tag{7.7}$$

However, if $P_L \neq P_U$, a problem arises. Minimizing the loss over labeled data results in good estimates in high-density regions of the labeled data, whereas we

3 Notice, however, that for the example of Chapter 5, in which we use a smaller sample size and a larger number of covariates, β-*Series* had better performance than β-*NN*.

are really interested in good estimates in high-density regions of the *unlabeled* target data. In other words, we seek estimators that minimize the loss

$$L(\widehat{f}, f) := \iint \left(\widehat{f}(z|\mathbf{x}) - f(z|\mathbf{x})\right)^2 dP_U(\mathbf{x})dz \tag{7.8}$$

with respect to P_U instead of P_L. But we do not know the z of the unlabeled data. This is where the importance weights are helpful: Under the covariate shift assumption $f_U(z|\mathbf{x}) = f_L(z|\mathbf{x})$, one can rewrite the modified loss (7.8) up to a constant as

$$\begin{aligned} L(\widehat{f}, f) &= \iint \widehat{f}^2(z|\mathbf{x})dP_U(\mathbf{x})dz - 2\iint \widehat{f}(z|\mathbf{x})f(z|\mathbf{x})dP_U(\mathbf{x})dz \\ &= \iint \widehat{f}^2(z|\mathbf{x})dP_U(\mathbf{x})dz - 2\iint \widehat{f}(z|\mathbf{x})\beta(\mathbf{x})dP_L(z,\mathbf{x})dz. \end{aligned}$$

(The last equality follows from $f_U(z|\mathbf{x})dP_U(\mathbf{x}) = f_L(z|\mathbf{x})\beta(x)dP_L(\mathbf{x}) = \beta(x)dP_L(z,\mathbf{x})$.) Hence, in a setting with selection bias, we propose the empirical (reweighted) loss function

$$\widehat{L}(\widehat{f}, f) = \frac{1}{\widetilde{n}_U}\sum_{k=1}^{\widetilde{n}_U}\int \widehat{f}^2\left(z|\widetilde{\mathbf{x}}_k^U\right)dz - 2\frac{1}{\widetilde{n}_L}\sum_{k=1}^{\widetilde{n}_L}\widehat{f}\left(\widetilde{z}_k^L|\widetilde{\mathbf{x}}_k^L\right)\widehat{\beta}\left(\widetilde{\mathbf{x}}_k^L\right). \tag{7.9}$$

This error estimate corrects for covariate shift and it can be computed from validation data that contain both labeled and unlabeled examples.

So how can we use this result in practice? In what follows, we present three different conditional density estimators designed to work under covariate shift (NN_{CS}, ker-NN_{CS}, and $Series_{CS}$); these estimators all minimize the reweighted empirical loss (7.9) on the validation data. Furthermore, in Section 7.4.4, we propose a general technique for combining two or more estimators to further decrease the loss. For model selection and for choosing tuning parameters, we use (7.9) together with an estimate of β (Eqs. 7.3 and 7.5). However, to assess the density estimators, we will for the *simulated* data examples (where we have access to the true labels $\widetilde{z}^U$) use the more accurate error estimate

$$\widehat{L}(\widehat{f}, f) = \frac{1}{\widetilde{n}_U}\sum_{k=1}^{\widetilde{n}_U}\int \widehat{f}^2\left(z|\widetilde{\mathbf{x}}_k^U\right)dz - 2\frac{1}{\widetilde{n}_U}\sum_{k=1}^{\widetilde{n}_U}\widehat{f}\left(\widetilde{z}_k^U|\widetilde{\mathbf{x}}_k^U\right), \tag{7.10}$$

which does not involve estimating β.

7.4.1 *Nearest Neighbor Histogram (NN_{CS})*

In Cunha et al. (2009), the authors propose a weighted nearest neighbor histogram for estimating the redshift distribution of photometric galaxy samples. We use the notation NN_{CS} to denote their histogram method but with tuning parameters chosen according to our scheme. The details are as follows.

Suppose we want to estimate the density $f(z|\mathbf{x})$ at a given point $\mathbf{x}$. Let $\mathcal{N}_N(\mathbf{x})$ denote the set of N nearest neighbors of $\mathbf{x}$ in the labeled data. Divide $[0,1]$ into B equal-sized bins, and let $b(z)$ denote the bin that includes z for $z \in [0,1]$. The weighted *nearest neighbor* histogram is given by

$$\widehat{f}(z|\mathbf{x}) \propto \sum_{k \in \mathcal{N}_N(\mathbf{x})} \widehat{\beta}\left(\mathbf{x}_k^L\right) \mathbb{I}\left(z_k^L \in b(z)\right).$$

The idea is to histogram the labeled examples $\{z_k^L : k \in \mathcal{N}_N(\mathbf{x})\}$ that are close to $\mathbf{x}$, and assign a weight $\widehat{\beta}(\mathbf{x}_k^L)$ to each labeled example that reflects how representative the example is of the target distribution.

In Cunha et al. (2009), the number of nearest neighbors N is set to 100, the number of bins B is chosen according to the application, and the importance weights $\widehat{\beta}(\mathbf{x}_k^L)$ are the β-*NN* estimates in Equation 7.3 based on 5 neighbors of the labeled point $\mathbf{x}_k^L$. Here, we instead choose the tuning parameters that minimize an estimate of the generalization error. We choose the parameters N and B that minimize the empirical loss in Equation 7.9 for the validation data, and we use Equation 7.5 to select the optimal number of nearest neighbors for estimating $\widehat{\beta}(\mathbf{x})$.

7.4.2 *Kernel Nearest Neighbor Estimator (ker-NN_{CS})*

To further improve upon the estimator NN_{CS}, we suggest using a smoothing kernel K_ϵ instead of a histogram. Hence, as an alternative to NN_{CS}, we propose the weighted *kernel nearest neighbor* estimator

$$\widehat{f}(z|\mathbf{x}) \propto \sum_{k \in \mathcal{N}_N(\mathbf{x})} \widehat{\beta}\left(\mathbf{x}_k^L\right) K_\epsilon\left(z - z_k^L\right).$$

In this work, we choose a Gaussian kernel $K_\epsilon(z - z_k) = e^{-(z-z_k)^2/4\epsilon}$ with bandwidth ϵ. As before, we base our choice of tuning parameters on the estimated generalization errors in (7.8) and (7.5). Note that setting $\widehat{\beta}(\mathbf{x}) \equiv 1$ for all $\mathbf{x} \in \mathcal{X}$ corresponds to the traditional kernel nearest neighbors estimator *not* corrected for selection bias (Zhao and Liu 1985); we denote this estimator by *ker-NN*.

7.4.3 *Spectral Series CDE under Covariate Shift (Series$_{CS}$)*

We can also adapt the spectral series estimator for the selection bias setting. To do that, here we use the *labeled* data to build the estimator, i.e., we use the labeled data to build the Gram Matrix

$$\mathbb{G}^L = \begin{bmatrix} K(\mathbf{x}_1^L, \mathbf{x}_1^L) & K(\mathbf{x}_1^L, \mathbf{x}_2^L) & \cdots & K(\mathbf{x}_1^L, \mathbf{x}_{n_L}^L) \\ K(\mathbf{x}_2^L, \mathbf{x}_1^L) & K(\mathbf{x}_2^L, \mathbf{x}_2^L) & \cdots & K(\mathbf{x}_2^L, \mathbf{x}_{n_L}^L) \\ \vdots & \vdots & \ddots & \vdots \\ K(\mathbf{x}_{n_L}^L, \mathbf{x}_1^L) & K(\mathbf{x}_{n_L}^L, \mathbf{x}_2^L) & \cdots & K(\mathbf{x}_{n_L}^L, \mathbf{x}_{n_L}^L) \end{bmatrix},$$

and also to estimate the coefficients. However, we choose the tuning parameters I, J, and ϵ that minimize an estimate of the loss in Eq. (7.8), which makes use of the unlabeled data and is more appropriate to the selection bias setting. We denote this importance-weighted spectral series estimator by *Series$_{CS}$*. We denote the estimator from Chapter 4 *not corrected for covariate shift* by *Series*. For both *Series* and *Series$_{CS}$*, we use the renormalization scheme from Section 4.2.2, as well as its technique for removing spurious bumps for both estimators.

7.4.4 *Combining Multiple Estimators (Comb$_{CS}$)*

Finally, we present a simple procedure for combining, or *stacking*, multiple estimators or models to further improve prediction performance. Suppose $\widehat{f}_1(z|\mathbf{x}), \ldots,$

$\widehat{f}_p(z|\mathbf{x})$ are different (cross-validated) estimators of $f(z|\mathbf{x})$. We define the combined estimator, $Comb_{CS}$, as a weighted average

$$\widehat{f}^{\boldsymbol{\alpha}}(z|\mathbf{x}) = \sum_{k=1}^{p} \alpha_k \widehat{f}_k(z|\mathbf{x}).$$

The choice of weights is crucial. We compute the empirical loss $\widehat{L}(\widehat{f}^{\boldsymbol{\alpha}}, f)$ on the validation data according to Eq. 7.9, and we seek the weights that minimize this loss under the constraints $\alpha_i \geqslant 0, \forall 1, \ldots, p$ and $\sum_{i=1}^{p} \alpha_i = 1$. The optimal vector weight vector $\boldsymbol{\alpha} = [\alpha_i]_{i=1}^{p}$ is the solution to a standard quadratic programming problem:

$$\underset{\boldsymbol{\alpha}:\alpha_i \geqslant 0, \sum_{i=1}^{p} \alpha_i = 1}{\arg\min} \quad \boldsymbol{\alpha}' \mathbb{B} \boldsymbol{\alpha} - 2\boldsymbol{\alpha}' b \tag{7.11}$$

where $\mathbb{B}$ is the $p \times p$ matrix $\left[\frac{1}{\tilde{n}_U} \sum_{k=1}^{\tilde{n}_U} \int \widehat{f}_i(z|\tilde{\mathbf{x}}_k^U) \widehat{f}_j(z|\tilde{\mathbf{x}}_k^U) dz\right]_{i,j=1}^{p}$ and b is the vector $\left[\frac{1}{\tilde{n}_L} \sum_{k=1}^{\tilde{n}_L} \widehat{f}_i(\tilde{z}_k^L|\tilde{\mathbf{x}}_k^L) \widehat{\beta}(\tilde{\mathbf{x}}_k^L)\right]_{i=1}^{p}$.

7.4.5 *Variable Selection*

The same technique from Section 7.3.3 can be used to improve on estimators of $f(z|\mathbf{x})$: an estimator $\widehat{f}(z|\mathbf{x}_0)$ built using any subset of covariates $\mathbf{x}_0$ can also be regarded as an estimator of $f(z|\mathbf{x})$. We also use forward stepwise-type model search to select which covariate to use to estimate $f(z|\mathbf{x})$, using Equation 7.9 to estimate loss (7.8). We initialize the procedure with an estimate of the marginal distribution $f(z)$.

7.4.6 *Goodness-of-fit*

Although the techniques we developed so far allow one to pick the best model among a set of candidates (using loss (7.8)), they do not indicate how reasonable the final estimates are. Here we describe two goodness-of-fit techniques that can give additional insights into these estimates. They are adaptations of the ideas from Section 4.4 to the selection bias setting.

Denote by $\widehat{F}_{z|\mathbf{x}_i}$ the estimated conditional cumulative distribution function for z given the covariates $\mathbf{x}_i$. The first evaluation is based on building a QQPlot. For each c in a grid of values in $[0,1]$ and each testing sample i, compute $Q_i^c = \widehat{F}^{-1}_{z|\mathbf{x}_i}(c)$. Define

$$\widehat{c} = \frac{1}{n}\sum_{i=1}^{n} \widehat{\beta}(x_i^L)\mathbb{I}(z_i^L \leqslant Q_i^c).$$

If the estimates of $f(z|\mathbf{x})$ are reasonable, $\widehat{c} \approx c$. We plot a graph of $\widehat{c}$'s versus c's and see how close they are to the diagonal.

The second diagnostic measure is a coverage plot. For each α in a grid of values in $[0,1]$ and each testing sample i, let A_i be a set such that $\int_{A_i} \widehat{f}(z|\mathbf{x}_i)dz = \alpha$. Here we choose the set A_i with the smallest area (the Highest Density Region). Define

$$\widehat{\alpha}_i = \frac{1}{n}\sum_{i=1}^{n} \widehat{\beta}(\mathbf{x}_i^L)\mathbb{I}(z_i^L \in A_i).$$

If the estimates of the conditional density are reasonable, $\widehat{\alpha} \approx \alpha$. We plot a graph of $\widehat{\alpha}$'s versus α's and see how close they are to the diagonal.

These diagnostic measures are not enough to distinguish *all* bad estimates from the good estimates. Nonetheless, they are useful to detect most bad estimates and can be used as an additional tool to the methods developed here.

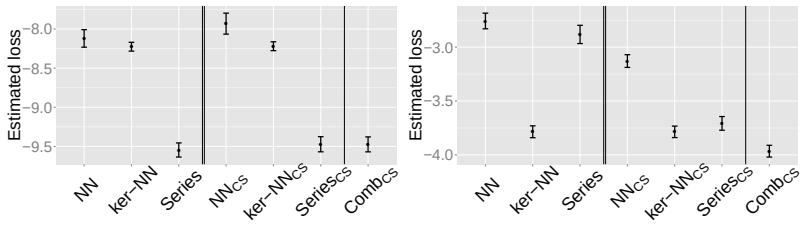
7.4.7 *Comparison of Conditional Density Estimators*

First, we use the simulated photo-z prediction setting to compare the performance of the various estimators of $f(z|\mathbf{x})$. We leave the discussion of the results to the end of the section. As before, our covariates are the four colors and the r-band magnitude in the `model` magnitude system. The weights $\beta(\mathbf{x})$ are estimated using the β-*NN* approach (Equation 7.3).

In total, we consider 7 density estimators: The first three methods do not take selection bias into account. They are *NN*, the nearest neighbor estimator from Section 7.4.1 with uniform weights $\widehat{\beta}(\mathbf{x}) \equiv 1$; *ker-NN*, the kernel nearest neighbor estimator from Section 7.4.2 with uniform weights; and *Series*, the spectral series estimator not corrected for covariate shift. The tuning parameters of these three estimators

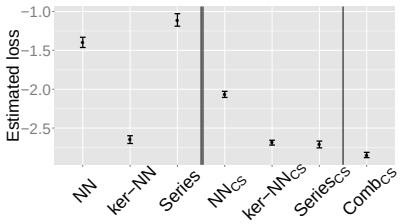
are chosen so as to minimize the empirical loss in Equation 7.7 over the labeled examples in the validation set.

The next three estimators are NN_{CS}, the nearest neighbors estimator from Section 7.4.1; *ker-NN*$_{CS}$, the kernel nearest neighbor estimator from Section 7.4.2; and *Series*$_{CS}$, the spectral series estimator corrected for covariate shift from Section 7.4.3. Finally, the last estimator is the combined estimator $Comb_{CS}$ from Section 7.4.4 by choosing a linear combination of *ker-NN*$_{CS}$ and *Series*$_{CS}$. The tuning parameters of these four estimators were chosen by minimizing the loss estimated in Equation 7.9 for the validation data.



(a) Scheme 1

(b) Scheme 2



(c) Scheme 3

Figure 7.6: Estimated losses of different density estimators in a simulated photo-z prediction setting with (a) no, (b) moderate, and (c) large covariate shifts. Bars correspond to mean plus and minus standard error.

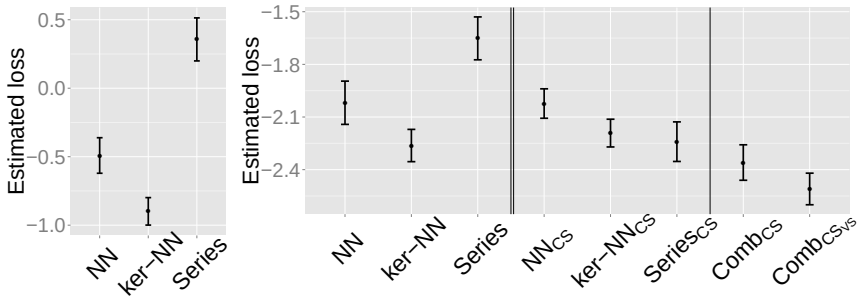
We split the data into training, validation and testing as in Section 7.3.4. To assess the performance of the cross-validated estimators, we compute the empirical loss in Equation 7.10 for the testing data. Figure 7.6 summarizes the results on the test data for the simulated photometric sets. We note that using exact instead of estimated importance weights yield plots similar to 7.6. Hence we omit these

results. For Scheme 1, the combined estimator gives average weight $\alpha = 0.96$ to $Series_{CS}$; this number is 0.50 for Scheme 2 and 0.43 in Scheme 3. Using the variable selection technique from Section 7.4.5 on the combined estimator leads us to choose the variables displayed in Table 7.2.

Table 7.2: Selected covariates for conditional density estimation for each dataset (combined estimator)

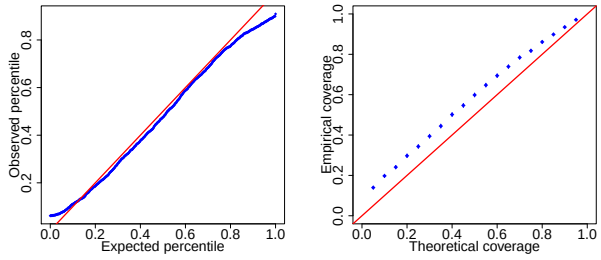
Dataset	model					cmodel				
	r	u − g	g − r	r − i	i − z	r	u − g	g − r	r − i	i − z
Scheme 1	X	X	X	X	X			X	X	X
Scheme 2	X		X	X		X				
Scheme 3	X		X	X		X				
SDSS	X		X	X		X				

We now perform the same analyses on the SDSS photometric set. We split the data into training, validation and testing as in Section 7.3.4. First, we show the results if we *do not* remove samples where the estimated importance weight are 0 (recall Section 7.3.4). Results are displayed in Figure 7.7a, where we use all covariates from `model` magnitude. On the other hand, Figure 7.7b shows the results if we proceed as suggested in Section 7.3.4 and substitute such samples for new ones where $\widehat{\beta}(\mathbf{x}) \neq 0$. The last row of Table 7.2 shows the variables that were selected under this scheme for the combined estimator. The loss of the final estimator was -2.51 ($\pm$0.09), smaller than -2.36 ($\pm$0.10), the loss achieved by the combined estimator using all 5 covariates from `model` magnitude. The weight given to $Series_{CS}$ was $\alpha = 0.53$. Finally, Figure 7.8 shows the goodness-of-fit tests of the final estimates of the combined model using the selected covariates.



(a) Original Sample (b) After removing $\widehat{\beta}(\mathbf{x}) = 0$

Figure 7.7: Estimated losses of conditional density estimators. Left: we use the original 15,000 spectroscopic samples. Right: we use 15,000 spectroscopic samples in which the initial estimates of the importance weights (which were then recomputed using the new sample) were different than zero. Notice that these plots have different scales.



(a) QQPlot (b) Coverage Plot

Figure 7.8: Goodness-of-fit plots for the final model, after variable selection was performed on the combined estimator.

Discussion. Our main findings are:

1. The kernel nearest-neighbor estimators (*ker-NN* and *ker-NN*$_{\mathrm{CS}}$) consistently perform better than the nearest-neighbor histogram estimators (*NN* and *NN*$_{\mathrm{CS}}$, respectively) for these data.

2. Kernel nearest neighbors, *ker-NN*, is remarkably robust to selection bias.

3. The spectral series method is sensitive to selection bias — but once corrected for covariate shift, $Series_{CS}$ is one of the best estimators. By combining *ker-NN*$_{CS}$ and $Series_{CS}$ into $Comb_{CS}$ we can improve the performance further. Variable selection improves the results even more.

4. It is important to remove samples with estimated importance weight zero (Figure 7.7); not taking selection bias into account leads to bad estimates.

The reason why *ker-NN* is so robust to selection bias is that, to use the terminology of Zadrozny (2004), it is a *local learner*. If $N(\mathbf{x}_0)$ is a sufficiently small neighborhood of $\mathbf{x}_0$, then under covariate shift (7.2), it holds that

$$f_L(z|\mathbf{x} \in N(\mathbf{x}_0)) \approx f_U(z|\mathbf{x} \in N(\mathbf{x}_0)).$$

The kernel nearest-neighbor estimator would then yield good density estimates even without a correction for selection bias, because then $\widehat{f}_L(z|\mathbf{x} \in N(\mathbf{x}_0)) \approx \widehat{f}_U(z|\mathbf{x} \in N(\mathbf{x}_0))$. For similar reasons, the histogram estimator *NN* is less robust than *ker-NN* (although it is also a local learner): Smoothing by binning $[0, 1]$ requires a larger number of neighbors, and hence larger neighborhoods than smoothing using a kernel for finite samples. For example, in Scheme 3 the number of neighbors chosen via cross validation is 35 for *NN* compared to only 8 for *ker-NN*. In these larger neighborhoods about a point $\mathbf{x}_0$, the above approximation may no longer be valid.

From a statistical point of view, it is interesting that *ker-NN*$_{CS}$ has good performance despite being extremely simple. As we saw in Chapter 4, conditional density estimation has considered to be a hard problem even when there are as few as 3 covariates (Fan et al. 2009), and has motivated several approaches in the absence of selection bias. We believe the good performance is because *ker-NN*$_{CS}$ is able to *adapt* to the intrinsic dimensionality of the data. That is, it is able to automatically find the lower dimensional structure on the 10 covariates, which are highly redundant. It has been shown that this is in fact the case for nearest neighbors *regression* (Kpotufe 2011).

We also observe that while *Series*$_{CS}$ has better performance than *ker-NN*$_{CS}$ when there is no selection bias (see results for Scheme 1, and also Chapter 4), they yield similar results otherwise. This is possibly because, when labeled and unlabeled sample are far from each other, the Nyström Extension of Equation (2.4) does not yield very good estimates of the eigenfunctions on the unlabeled sample, as these were built using the labeled data. A similar reason explains why using spectral series does not dominate nearest neighbors when estimating a density ratio: in order to estimate the coefficients of the expansion, one needs to use the Nyström Extension to evaluate the basis on the unlabeled data, although the basis was estimated using labeled data only. Notice this does not happen in the other spectral series estimators. In Chapter 8 we discuss how one may proceed to avoid this problem.

It is also interesting that when there is selection bias, less variables are chosen (Table 7.2). This is because the large covariate shift makes the effective sample size smaller in Schemes 2) and 3) (essentially many labeled samples have weight zero; see also discussion in Shimodaira 2000 and Gretton et al. 2010), and hence the variance of the estimators are larger. This is exactly the reason why substituting labeled samples with $\widehat{\beta}(\mathbf{x}) = 0$ to samples where $\widehat{\beta}(\mathbf{x}) \neq 0$ improves the results substantially.

Finally, the QQPlot is reasonable for almost all percentiles, except in the tails, which indicates that the fitted estimates have bigger tails than the real conditional densities. This is in agreement with the coverage plot, which shows that the empirical coverage is around 8% larger than the nominal one. This indicates that there is still room for improvement on the final estimates.

As an illustration, in Figure 7.9 we display the final conditional density estimates of six galaxies from the *spectroscopic* sample, along with the real redshifts and estimated importance weights. We present the results from *Series*$_{CS}$, *ker-NN*$_{CS}$ and *Comb*$_{CS}$. It also shows the final conditional density estimates of six galaxies from the *photometric* sample. As expected, most of the estimates are multimodal

and asymmetric, indicating the regression $\mathbb{E}[Z|\mathbf{x}]$ would not be a good summary of these.

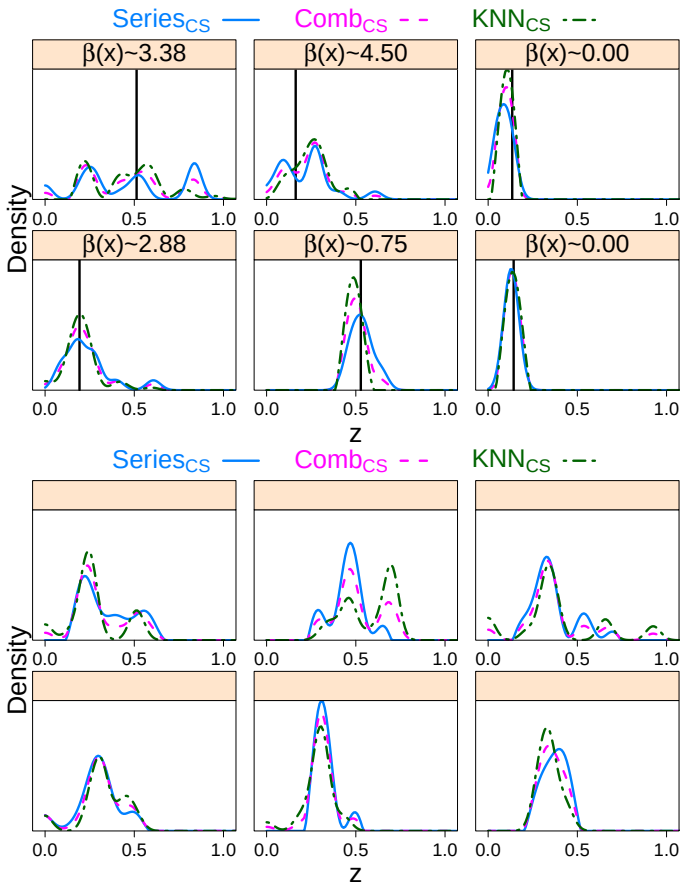


Figure 7.9: Top: Examples of estimated densities on spectroscopic sample and estimated importance weights. Vertical lines indicate observed spectroscopic redshift. Bottom: Examples of estimated densities on photometric sample.

7.5 EXAMPLE: GALAXY-GALAXY LENSING

As a proof-of-concept, in this section we apply the methods we develop to the galaxy-galaxy weak lensing analysis from Sheldon et al. (2012). The goal is to estimate the *critical surface density*, $\Sigma(z_l, z_s)$, which determines the lensing strength of a given lens-source pair, see Mandelbaum et al. (2008) for additional details. More precisely, we need to estimate $\Sigma^{-1}(z_l, z_s)$ for a set of source galaxies with

unknown redshifts z_s's. Here, z_l is the desired lens redshift, assumed to be fixed. A simple estimate is given by $\Sigma^{-1}(z_l, \widehat{z}_s)$, where $\widehat{z}_s$ is a point estimate of the source galaxy redshift z_s, typically obtained via regression. However, because $\mathbb{E}[g(Z)|\mathbf{x}] \neq g(\mathbb{E}[Z|\mathbf{x}])$ (here, $g(z) := \Sigma^{-1}(z_l, z)$), the estimator

$$\widehat{\Sigma^{-1}}(z_l, z_s) = \int \Sigma^{-1}(z_l, z)\widehat{f}(z|\mathbf{x})\,dz$$

usually yields better results if $\widehat{f}(z|\mathbf{x})$ is a good estimate of $f(z|\mathbf{x})$. We notice that another interesting approach to estimate $\mathbb{E}[g(Z)|\mathbf{x}]$ effectively is to directly regress the variable $W \equiv g(Z)$ on $\mathbf{x}$: This also avoids the problem that $\mathbb{E}[g(Z)|\mathbf{x}] \neq g(\mathbb{E}[Z|\mathbf{x}])$. Hence, we also implement an estimator of this type in the experiment that follows.

As in Sheldon et al. (2012), we use data from DEEP2 EGS Region (Weiner et al. 2005). Besides the estimators we described in this chapter (*NN*, *ker-NN*, *Series*, and *Comb*), we also implement *NN-7*, the nearest neighbors estimators with 7 neighbors (the value used by Sheldon et al. 2012 for this particular application), and *Photo-z*, a nearest neighbors *regression* estimator of z. That is, *Photo-z* is simply a plug-in estimator of the type $g(\widehat{z})$. Finally, we also compute *Series Reg.*, the spectral series *regression* estimator of $W \equiv g(Z)$ on $\mathbf{x}$ (i.e., contrary to *Photo-z*, *Series Reg.* is not a plug-in estimator).

We evaluate the performance of these estimators via the measures defined by Mandelbaum et al. (2008), named bias and variance ratio. We notice these are not the standard definitions of bias and variance from statistics. Small values of bias and large values of variance ratio indicate good performance. We use 500 samples for training, 500 for validation, and 382 for testing.

Results are shown in Figure 7.10. We do not display the combined estimator because it is essentially the same as Series in this case ($\alpha = 1$).

We observe the following key facts:

- The plug-in-based estimator, *Photo-z*, has a large bias;

- *ker-NN* has performance in between *NN* and *Series*, which is in agreement with the analysis of Section 7.4.7: when there is no selection bias, *Series* has best performance;

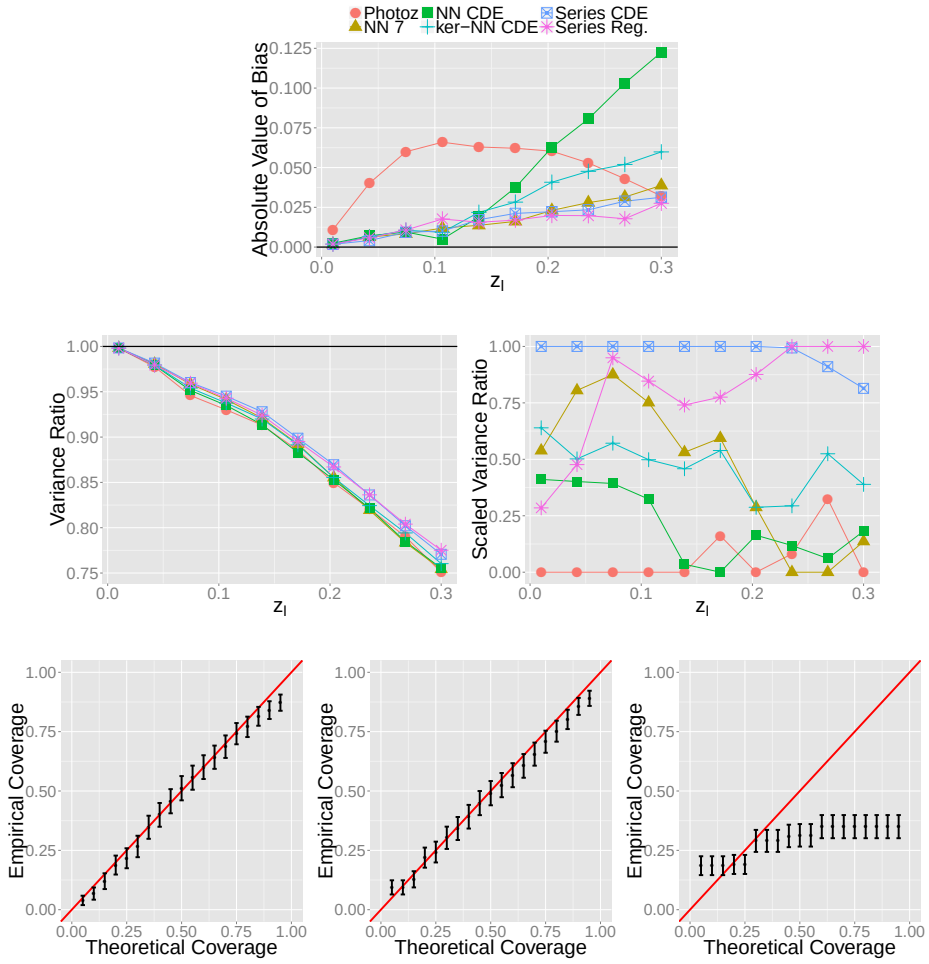
- In terms of bias, the estimators with best performance are *Series*, *Series Reg.*, and *NN-7*;

- In terms of variance, the estimators with best performance are *Series* and *Series Reg.*;

- Nearest neighbors with 7 neighbors performs better than that with 27 neighbors, the value chosen via the technique described in Section 7.4. On the other hand, 7 neighbors yields a bad density estimator (e.g., it does not have good coverage, see Figure 7.10).

How can NN with 7 neighbors give better estimates of the critical surface density than NN with 27 neighbors, which has better goodness-of-fit? The reason is that estimating $f(z|\mathbf{x})$ by NN and then computing $\int g(z)\widehat{f}(z|\mathbf{x})\,dz$ is essentially equivalent to estimating $\mathbb{E}[g(Z)|\mathbf{x}]$ *directly* by performing nearest neighbors *regression* of $W \equiv g(Z)$ on $\mathbf{x}$, the same task performed by *Series Reg.* Hence, 7 yields a good *regression* estimator of $g(Z)$, but a bad density estimator. This is in agreement with the fact that when tuning *NN* using the loss function for *regression* estimators (Chapter 3) we recover the value 7. Moreover, overall *Series Reg* has better performance than *NN-7*, which is in agreement with the results from Chapter 3.

These results suggest that one may consider directly regressing the response of interest, $g(z)$, on the photometric covariates as an alternative to performing conditional density estimation. Nonetheless, this alternative procedure requires a new tuning of parameters for each g of interest; for example, other numbers than 7 will be optimal depending on the function one is estimating: Generally, smoother functions will requires a larger k for optimality. On the other hand, estimating the conditional density requires the tuning to be performed only once.



(d) Series (e) NN (f) NN 7

Figure 7.10: Galaxy-galaxy lensing using DEEP2. Scaled Variance Ratio is Variance Ratio normalized to be have minimum at 0 and maximum at 1. *Series* and *Series-Reg* yield estimates with smaller biases and variances than the other approaches. Using 7 neighbors for NN as in Sheldon et al. (2012) yields similar performance in terms of bias, however it gives unreasonable density estimates.

7.6 Conclusions

Here, we proposed and analyzed non-parametric methods for estimating conditional densities under selection bias. More specifically, we worked with the Missing at Random assumption in the context of photometric redshift prediction.

To match training (spectroscopic) and target (photometric) samples, we used a reweighting scheme based on importance weights. We found that the nearest neighbors estimator developed in Cunha et al. (2009) is very effective for estimating these weights, even when compared to state-of-the-art approaches of density ratio estimation from the machine learning literature. We provided a principled way of choosing the tuning parameter of the importance weight estimator, and introduced two new conditional density estimators, *kernel nearest neighbors* (Section 7.4.2) and *Series* (Section 7.4.3), both with better performance than the photo-z prediction method by Cunha et al. (2009). We found that the kernel nearest neighbors estimator is relatively robust to departures from i.i.d. situations even when not corrected for selection bias. When taking selection bias into account, the kernel nearest neighbors and the spectral series estimators yield similar performance. We proposed principled ways of tuning their parameters under selection bias, and we described how to combine two or more estimators for optimal performance. In particular, we saw that such procedures lead to better inference on galaxy-galaxy lensing problems. We also proposed a scheme for variable selection when estimating importance weights and conditional densities. Most likely, variable selection will be essential in next-generation surveys that include additional covariates (e.g., surface brightness or sizes of galaxies, Lima et al. 2008, or other magnitudes such as grizYJHKs, Oyaizu et al. 2008).

In summary, for our study of SDSS galaxy data, we found that the following procedure gave the best photo-z estimates: First, compute initial estimates of the importance weights $\beta(\mathbf{x})$ using nearest neighbors (Section 5.4). Then, to increase the effective sample size, remove data points where the estimated weights $\widehat{\beta}(\mathbf{x})$ are zero, substituting them for new samples with $\widehat{\beta}(\mathbf{x}) \neq 0$, and reestimate the importance weights. Finally, to estimate the redshift distribution $f(z|\mathbf{x})$ under selection bias, use the technique from Section 7.4.4 to combine the weighted kernel nearest neighbors and spectral series estimators (ker-NN_{CS} and $\text{Seriest}_{\text{CS}}$, respectively) into a final estimator Comb_{CS}. For optimal performance, use the variable selec-

tion scheme in Sections 7.3.3 and 7.4.5 to decide which covariates to include in the estimation of both $\beta(\mathbf{x})$ and $f(z|\mathbf{x})$.

The goodness-of-fit techniques we propose indicate that the final conditional density estimates are reasonable on SDSS data, but that there is still room for improvement. We expect that one would be able to achieve even better results by using larger sample sizes as well as by aggregating more conditional density estimators. One could also iterate the procedure of removing samples where $\beta(\mathbf{x}) = 0$.

Finally, although the scope of this chapter is conditional density estimation, taking into account selection bias is also important for regression. Much of the proposed work can directly be adapted to regression estimators of, for example, photometric redshift as the regression function $\mathbb{E}[Z|\mathbf{x}] = \int z f(z|\mathbf{x}) d\mathbf{x}$.

Part IV

CONCLUSIONS AND FUTURE WORK

8

CONCLUSIONS AND FUTURE WORK

In this thesis we developed a general nonparametric framework for estimating functions defined on a high-dimensional space. The method is based on expanding the function of interest using spectral bases, the eigenfunctions of a kernel-based operator. We applied this technique to four distinct problems, and saw several examples where it yields estimates comparable to or better than those from state-of-the-art techniques. We also saw that the spectral bases adapt to the domain of the data, yielding improved rates when data live on a space with small intrinsic dimensionality but is embedded in a high-dimensional space. In particular, we showed that if a function is smooth with respect to the data distribution, we only need the first few eigenfunctions to approximate it well, i.e., smoothness implies a sparse representation for the spectral basis. Moreover, we saw that, for many problems, spectral series methods have better computational properties than competing methods, one of the reasons being the orthogonality of the eigenfunctions with respect to the data distribution. Additionally, we discussed improvements that make the estimators scale to larger datasets with almost no loss in accuracy. We saw that the spectral series method is also very flexible: It can handle different data types well, and it can also make use of unlabeled data in a semi-supervised learning setting. Moreover, it automatically provides basis functions that can be used for data visualization. Finally, we presented a real application to the problem

of photometric redshift prediction, where the methods we develop lead to more accurate inference than traditional approaches.

8.1 FUTURE WORK

The methodology we developed in this thesis motivates several interesting future directions.

8.1.1 *Confidence Bands*

First, it would be desirable to construct confidence sets for the estimates given by the spectral series method. This is of particular interest on the likelihood estimation problem, where having such estimates may be a first step towards quantifying the level of approximation on the inferences. This would represent a substantial advantage over standard ABC methods, where it is typically hard to quantify such uncertainties.

A first step towards building confidence bands over spectral series estimators is to notice they are *linear smoothers*. Consider for example the regression estimator with Kernel PCA basis (see Chapter 3). Then,

$$\widehat{r}(\mathbf{x}) = \sum_{i=1}^{I} \widehat{\beta}_i Y_i = \sum_{i=1}^{I} \left(\frac{1}{n} \sum_{k=1}^{n} \widehat{\psi}_i(\mathbf{x}_k) \widehat{\psi}_i(\mathbf{x}) \right) Y_i = \sum_{k=1}^{n} \left(\frac{1}{n} \sum_{i=1}^{I} \widehat{\psi}_i(\mathbf{x}_k) \widehat{\psi}_i(\mathbf{x}) \right) Y_i = \\ = \sum_{k=1}^{n} l_i(\mathbf{x}) Y_i,$$

where $l_i(\mathbf{x}) = n^{-1} \sum_{i=1}^{I} \widehat{\psi}_i(\mathbf{x}_k) \widehat{\psi}_i(\mathbf{x})$. One may then use the techniques discussed in Chapter 5 of Wasserman (2006) and in Sun and Loader (1994) to build bands with the desired coverage for $\mathbb{E}[\widehat{r}(\mathbf{x})]$.

8.1.2 *Spectral Series Estimators and Selection Bias*

As we saw in Chapters 3, 4 and 7, the spectral series estimator typically yields better estimates than nearest neighbors methods. However, when there is selection

bias, they give similar results. As we discussed, this is possibly because in this case the Nyström extension of the basis to the testing samples is not very precise, mainly because the basis is build using the labeled sample, which is far from the unlabeled one. A solution to this may be to use both samples to estimate the basis. More specifically, one might build $(\psi_j(\mathbf{x}))_j$ sampling from the population $\gamma P_U(\mathbf{x}) + (1-\gamma)P_L(\mathbf{x})$, where γ can be chosen by minimizing the estimated loss. Then, in the context of the conditional density estimation problem, the coefficients of the expansion would be given by

$$\begin{aligned}\beta_{i,j} &= \iint \phi_i(z)\psi_j(\mathbf{x})f(z|\mathbf{x})\,d\left(\gamma P_U(\mathbf{x}) + (1-\gamma)P_L(\mathbf{x})\right)\\ &= \gamma\iint \phi_i(z)\psi_j(\mathbf{x})f(z|\mathbf{x})P_U(\mathbf{x}) + (1-\gamma)\iint \phi_i(z)\psi_j(\mathbf{x})f(z|\mathbf{x})\,dP_L(\mathbf{x})\\ &= \gamma\mathbb{E}_L[\phi_i(Z)\psi_j(\mathbf{X})\beta(\mathbf{X})] + (1-\gamma)\mathbb{E}_L[\phi_i(Z)\psi_j(\mathbf{X})],\end{aligned}$$

which can be estimated using the ideas presented in Chapter 7. Similar ideas can be used to improve the spectral series estimator of a density ratio (Chapter 5).

8.1.3 *Rates of Convergence*

From a theoretical perspective, an interesting question is whether our bounds for spectral series methods – which are of order $n^{-1/O(p^2)}$ when there is finite unlabeled data[1] – can be improved to $n^{-1/O(p)}$. We believe the answer is yes, with one of the reasons being that although the nearest neighbors regression estimator has a rate of the form $n^{-1/O(p)}$ (Kpotufe 2011), spectral series typically has better performance in our experiments. One way to achieve improved bounds may be to use the fact that we empirically observe that the eigenfunctions of the kernel operators are better estimated on the sample points. If this is indeed the case, the current bounds on the estimated coefficients are too conservative: To estimate the coefficients it is not necessary to use the Nyström extension, although we do not make use of this in the proofs we present in the appendices. Hence, having better bounds for $\widehat{\psi}_j(\mathbf{x})$'s on the sample points may lead to improved rates. A second

1 Recall that p is the intrinsic dimension of the data.

path that may lead one to achieve better bounds is to note that the $\widehat{\psi}_j$'s are still a basis of functions (in a broad sense) even if they are not close to ψ_j.

Still regarding the adaptation of our method to low-dimensional structure, an interesting question is whether assuming data live *close to* a submanifold of the original space still yields improved bounds. Similarly, it would be desirable to evaluate if improved rates can be obtained under more general assumptions on the data domain.

8.1.4 *Other Applications of Spectral Series*

Finally, it would be interesting to investigate how to apply the spectral series method to other statistical problems. An interesting application may be to smooth tests (Neyman 1937; Kallenberg and Ledwina 1997; Bera and Ghosh 2002), where one typically uses a *one-dimensional* Fourier basis, or Lagrange polynomials. Spectral series may allow such tests to be extended to higher dimensions. One way this can be done is by using density ratio estimation (Chapter 5). More precisely, assume

$$\mathbf{X}_1, \dots, \mathbf{X}_n \overset{\text{i.i.d.}}{\sim} F,$$

and say we are interested in testing the hypothesis $H_0 : F = F_0$. Assume we are able to sample from F_0, and that $F \ll F_0 \ll \lambda$, where λ is the Lebesgue measure. Then H_0 is equivalent to

$$H_0 : \frac{f(\mathbf{x})}{f_0(\mathbf{x})} \equiv 1.$$

Now, let $(\widehat{\psi}_i)_i$ be the estimated diffusion spectral basis (see Section 2.2.4 for details; here, for simplicity, we omit in the notation the dependence of the basis and related quantities on ϵ), estimated using a simulated sample from F_0. Let

$$\frac{f(\mathbf{x})}{f_0(\mathbf{x})} = \sum_i \beta_i \psi_i(\mathbf{x}),$$

where

$$\beta_i = \int \frac{f(\mathbf{x})}{f_0(\mathbf{x})} \psi_i(\mathbf{x}) dS(\mathbf{x}) = \int \frac{f(\mathbf{x})}{f_0(\mathbf{x})} \psi_i(\mathbf{x}) s(\mathbf{x}) dF_0(\mathbf{x}) = \mathbb{E}_F[\psi_i(\mathbf{X}) s(\mathbf{X})].$$

The key for adapting the smooth testing to this framework is to recall that, for the diffusion basis, $\psi_0(\mathbf{x}) = \widehat{\psi}_0(\mathbf{x}) \equiv 1$. Hence, the hypothesis H_0 can be rewritten as

$$H_0 : \forall i > 0,\ \beta_i = 0.$$

This motivates the use of the test statistic

$$T = \sum_{i=1}^{I} \widehat{\beta}_i^2,$$

where $\widehat{\beta}_i$ is the estimated projection coefficient on the eigenfunction i, given by

$$\widehat{\beta}_i = \frac{1}{n} \sum_{k=1}^{n} \widehat{\psi}_i(\mathbf{x}_k)\widehat{s}(\mathbf{x}_k).$$

One would then reject the null when $T > K$, where K is chosen so that the test has the desired level α. The cutoff K can be chosen by sampling from F_0. This yields a simple procedure for performing smooth tests in sample spaces with more than one dimension.

As an illustration, the left plot of Figure 8.1 shows the power function of this test for testing $H_0 : \mu = 0$ given a sample $X_1, \ldots, X_{50} \overset{\text{i.i.d.}}{\sim} N(\mu, 1)$ for a fixed $\alpha = 5\%$. We use sample of size 5,000 to estimate the basis, an expansion with $I = 5$ terms, and a bandwidth

$$\epsilon = \text{median}_{i,j} \left(d^2(X'_i, X'_j)/4 \right),$$

where $X'_1, \ldots, X'_{5000}$ is the sample used to estimate the basis functions. We also include a comparison with the data-driven smooth test from Kallenberg and Ledwina (1997) (Neyman's smooth test),[2] and with the likelihood ratio test (the Z test). We observe that the spectral series smooth test has power comparable to that of the traditional smooth test. Moreover, as expected, its power is inferior to the Z-test, because the latter assumes a parametric form for the likelihood function. A similar phenomenon happens in the right plot of that figure, where we test the hypothesis that $H_0 : \boldsymbol{\mu} = (0, 0, 0)$ based on 150 samples from three dimensional Gaussian with covariance matrix equals to the identity. The power is plotted as a function of the

2 We use the package *ddst* from R, http://cran.fhcrc.org/web/packages/ddst

mean vector $\boldsymbol{\mu} = (\mu, \mu, \mu)$. Notice that because the sample space has dimension three, it is not possible to use the traditional smooth test for this problem.

Some aspects to be investigated are:

- How to choose the truncation point I?
- How to choose the bandwidth ϵ?
- How to adapt this framework to test composite hypotheses?
- What are the power properties of this test? In particular, how does it compare to other tests from the literature?

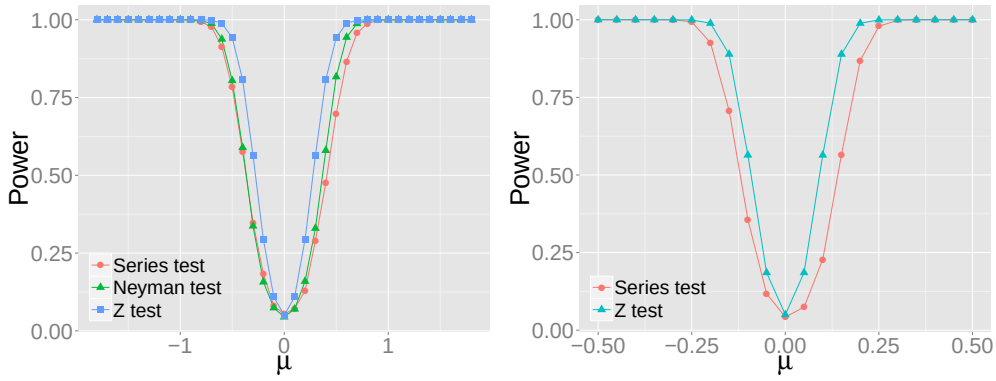


Figure 8.1: Power curves of tests for the mean of a Gaussian. Significance levels are $\alpha = 5\%$ for all the tests. Left: $\mathbf{x} \in \mathbb{R}$. Right: $\mathbf{x} \in \mathbb{R}^3$. The Z test is more powerful, however it assumes a parametric form for the likelihood function. On the other hand, the standard adaptive Neyman smooth test has power comparable to that obtain via spectral series.

A second possible application is to inverse problems (O'Sullivan 1986; Wasserman 2006). Here, traditional solutions use Fourier basis or wavelets to expand the regression function (Donoho and Johnstone 1995; Abramovich and Silverman 1998; Abramovich et al. 2000). Spectral series may offer a natural extension to higher dimensional spaces.

Part V

APPENDICES

A

APPENDIX: BOUNDS ON THE REGRESSION ESTIMATOR

We start by stating some useful lemmas.

Lemma A.1. $\forall x \in \mathcal{X}$,

$$\frac{a}{b} \leqslant s_\epsilon(\mathbf{x}) \leqslant \frac{b}{a}$$

Proof. $\forall x \in \mathcal{X}$,

$$\frac{\inf_{\mathbf{x}\in\mathcal{X}} p_\epsilon(\mathbf{x})}{\sup_{\mathbf{x}\in\mathcal{X}} p_\epsilon(\mathbf{x})} \leqslant s_\epsilon(\mathbf{x}) \leqslant \frac{\sup_{\mathbf{x}\in\mathcal{X}} p_\epsilon(\mathbf{x})}{\inf_{\mathbf{x}\in\mathcal{X}} p_\epsilon(\mathbf{x})},$$

where $a \int_{\mathcal{X}} K_\epsilon(\mathbf{x},\mathbf{y})d\mathbf{y} \leqslant p_\epsilon(\mathbf{x}) \leqslant b \int_{\mathcal{X}} K_\epsilon(\mathbf{x},\mathbf{y})d\mathbf{y}$. □

Lemma A.2. *For* $g \in L^2(\mathcal{X}, P)$,

$$L_{\text{bias}} \leqslant \frac{b}{a} \sum_{j>J} |\beta_{\epsilon,j}|^2.$$

Proof. From the orthogonality property of the basis functions ψ_j, we have that

$$\int_{\mathcal{X}} |g(\mathbf{x}) - g_{\epsilon,J}(\mathbf{x})|^2 dS_\epsilon(\mathbf{x}) = \sum_{j>J} |\beta_{\epsilon,j}|^2.$$

The result follows from Lemma A.1. □

Lemma A.3. *(Coifman and Lafon, 2006, Proposition 3) For* $g \in C^3(\mathcal{X})$ *and* $x \in \mathcal{X} \setminus \partial\mathcal{X}$,

$$-\lim_{\epsilon\to 0} G_\epsilon^* = \triangle.$$

If $\mathcal{X}$ *is a compact* C^∞ *submanifold of* $\mathbb{R}^d$, *then* $\triangle$ *is the psd Laplace-Beltrami operator of* $\mathcal{X}$ *defined by* $\triangle g(\mathbf{x}) = -\sum_{j=1}^r \frac{\partial^2 g}{\partial s_j^2}(\mathbf{x})$ *where* $(s_1, \ldots, s_r)$ *are the normal coordinates of the tangent plane at* x.

Lemma A.4. *Under the same assumptions as in Proposition A.1, it holds that*

$$\|\varphi_{\epsilon,j} - \widehat{\varphi}_{\epsilon,j}\|_{L^2(\mathcal{X},P)} = O_P\left(\frac{\gamma_n}{\delta_{\epsilon,j}}\right),$$

where $\gamma_n = \sqrt{\frac{\log(1/\epsilon_n)}{n\epsilon_n^{d/2}}}$ *and* $\delta_{\epsilon,j} = \lambda_{\epsilon,j} - \lambda_{\epsilon,j+1}$.

Proof. From Giné and Guillou (2002), $\sup_x |\widehat{p}_\epsilon(x) - p_\epsilon(x)| = O_P(\gamma_n)$. Hence,

$$\sup_x |\widehat{s}_\epsilon(x) - s_\epsilon(x)| = O_P(\gamma_n).$$

By using Proposition A.1, we conclude that

$$\int_{\mathcal{X}} |\widehat{\psi}_{\epsilon,j}(x)|^2 dP(x) \leqslant 2\int_{\mathcal{X}} |\widehat{\psi}_{\epsilon,j}(x) - \psi_{\epsilon,j}(x)|^2 dP(x) + 2\int_{\mathcal{X}} |\psi_{\epsilon,j}(x)|^2 dP(x) = O_P\left(\frac{\gamma_n^2}{\delta_{\epsilon,j}^2}\right) + C,$$

where C is a constant. Write

$$\begin{aligned}
|\varphi_{\epsilon,j}(x) - \widehat{\varphi}_{\epsilon,j}(x)|^2 &= |\psi_{\epsilon,j}(x)s_\epsilon(x) - \widehat{\psi}_{\epsilon,j}(x)\widehat{s}_\epsilon(x)|^2 \\
&\leqslant 2|\psi_{\epsilon,j}(x) - \widehat{\psi}_{\epsilon,j}(x)|^2|s_\epsilon(x)|^2 + 2|s_\epsilon(x) - \widehat{s}_\epsilon(x)|^2|\widehat{\psi}_{\epsilon,j}(x)|^2.
\end{aligned}$$

Hence,

$$\begin{aligned}
\|\varphi_{\epsilon,j} - \widehat{\varphi}_{\epsilon,j}\|^2_{L^2(\mathcal{X},P)} &\leqslant 2\sup_x |s_\epsilon(x)|^2 \, \|\widehat{\psi}_{\epsilon,j} - \psi_{\epsilon,j}\|^2_{L^2(\mathcal{X},P)} + 2\sup_x |\widehat{s}_\epsilon(x) - s_\epsilon(x)|^2 \, \|\widehat{\psi}_{\epsilon,j}\|^2_{L^2(\mathcal{X},P)} \\
&= O_P\left(\frac{\gamma_n^2}{\delta_{\epsilon,j}^2}\right) + O_P(\gamma_n^2)\left(O_P\left(\frac{\gamma_n^2}{\delta_{\epsilon,j}^2}\right) + C\right) \\
&= O_P\left(\frac{\gamma_n^2}{\delta_{\epsilon,j}^2}\right).
\end{aligned}$$

□

Lemma A.5. $\forall 0 \leqslant j \leqslant J$, *it holds that*

$$\left|\frac{1}{n}\sum_{i=1}^n Y_i(\widehat{\varphi}_{\epsilon,j}(X_i) - \varphi_{\epsilon,j}(X_i)) - \int_{\mathcal{X}} r(x)(\widehat{\varphi}_{\epsilon,j}(x) - \varphi_{\epsilon,j}(x))dP(x)\right| = O_P\left(\frac{1}{\sqrt{n}}\right).$$

Proof. Let $S = \frac{1}{n}\sum_{i=1}^n Y_i(\widehat{\varphi}_{\epsilon,j}(X_i) - \varphi_{\epsilon,j}(X_i))$ and $I = \int_{\mathcal{X}} r(x)(\widehat{\varphi}_{\epsilon,j}(x) - \varphi_{\epsilon,j}(x))dP(x)$.

According to Chebyshev's inequality, for any $M > 0$,

$$\begin{aligned}
\mathbb{P}\left(|S - I| \geqslant M \,|\, \tilde{X}_1, \ldots, \tilde{X}_n\right) &\leqslant \frac{\mathbb{V}\left(S - I|\tilde{X}_1, \ldots, \tilde{X}_n\right)}{M^2} \\
&\leqslant \frac{\mathbb{V}\left(Y_1(\widehat{\varphi}_{\epsilon,j}(X_1) - \varphi_{\epsilon,j}(X_1)) \,\middle|\, \tilde{X}_1, \ldots, \tilde{X}_n\right)}{nM^2}.
\end{aligned}$$

Hence, for any $M > 0$,

$$\mathbb{P}\left(|S - I| \geqslant M\right) \leqslant \frac{\mathbb{V}\left(Y_1(\widehat{\varphi}_{\epsilon,j}(\mathbf{X}_1) - \varphi_{\epsilon,j}(\mathbf{X}_1))\right)}{nM^2} \leqslant \frac{\sigma}{nM^2}\left(\mathbb{E}|\widehat{\varphi}_{\epsilon,j}(\mathbf{X}_1) - \varphi_{\epsilon,j}(\mathbf{X}_1)|^2\right)^{1/2},$$

where we in the last inequality apply Cauchy-Schwartz. Under assumption (A4), we conclude the result of the lemma. □

A.1 Bias

Proof of Proposition 3.1. Note that $\mathcal{J}_\epsilon(g) = \sum_j \nu_{\epsilon,j}^2 |\beta_{\epsilon,j}|^2$. Hence,

$$\frac{\mathcal{J}_\epsilon(g)}{\nu_{\epsilon,J+1}^2} = \sum_j \frac{\nu_{\epsilon,j}^2}{\nu_{\epsilon,J+1}^2}|\beta_{\epsilon,j}|^2 \geqslant \sum_{j>J} \frac{\nu_{\epsilon,j}^2}{\nu_{\epsilon,J+1}^2}|\beta_{\epsilon,j}|^2 \geqslant \sum_{j>J} |\beta_{\epsilon,j}|^2 = \int_X |g(\mathbf{x}) - g_{\epsilon,J}(\mathbf{x})|^2 dS_\epsilon(\mathbf{x}).$$

The last result follows from Lemma A.2. □

Proof of Lemma 3.1. By Green's first identity

$$\int_{\mathcal{X}} g\, \nabla^2 g dS(\mathbf{x}) + \int_{\mathcal{X}} \nabla g \cdot \nabla g dS(\mathbf{x}) = \oint_{\partial \mathcal{X}} g(n \cdot \nabla g) dS(\mathbf{x}) = 0,$$

where n is the normal direction to the boundary $\partial\mathbf{X}$, and the last surface integral vanishes due to the Neumann boundary condition. It follows from Lemma A.3 that

$$\lim_{\epsilon\to 0} \mathcal{J}_\epsilon^*(g) = -\lim_{\epsilon\to 0} \int_{\mathcal{X}} g(\mathbf{x}) G_\epsilon^* g(\mathbf{x}) dS_\epsilon(\mathbf{x}) = \int_{\mathcal{X}} g(\mathbf{x}) \triangle g(\mathbf{x}) dS(\mathbf{x}) = \int_{\mathcal{X}} \|\nabla g(\mathbf{x})\|^2 dS(\mathbf{x}). \square$$

Proof of Theorem 3.1. We have that

$$c^2 \geqslant \int_{\mathcal{X}} \|\nabla g(\mathbf{x})\|^2 dS(\mathbf{x}) = \int_{\mathcal{X}} g(\mathbf{x}) \triangle g(\mathbf{x}) dS(\mathbf{x}) = \sum_j \nu_j^2 \beta_j^2,$$

where $\nu_j^2 = O(j^{2s})$. Hence, $g \in W_{\mathcal{B}}(s, c)$ and by Theorem 9.1 in Mallat (2009), $\|g - g_J\|^2 = o(J^{-2s})$. □

A.2 VARIANCE

One can view the matrix $\mathbb{A}_\epsilon$ (defined in Equation 2.6) as a perturbation of the integral operator A_ϵ due to finite sampling. We would like to bound the difference $\psi_{\epsilon,j} - \widehat{\psi}_{\epsilon,j}$, where $\psi_{\epsilon,j}$ are the eigenvectors of A_ϵ, and $\widehat{\psi}_{\epsilon,j}$ are the Nyström extensions (Eq. 2.9) of the eigenvectors of $\mathbb{A}_\epsilon$. One strategy proposed by Rosasco et al. (2010) is to introduce two new integral operators that are related to A_ϵ and $\mathbb{A}_\epsilon$, but that both act on an auxiliary[1] RKHS $\mathcal{H}$ of smooth functions. Define $A_\mathcal{H}, \widehat{A}_\mathcal{H} : \mathcal{H} \to \mathcal{H}$ where

$$\begin{aligned} A_\mathcal{H} f(\mathbf{x}) &= \frac{\int K_\epsilon(x,y)\langle f, K(\cdot,y)\rangle_\mathcal{H} dP(\mathbf{y})}{\int K_\epsilon(\mathbf{x},\mathbf{y}) dP(\mathbf{y})} = \int a_\epsilon(\mathbf{x},\mathbf{y})\langle f, K(\cdot,y)\rangle_\mathcal{H}\ dP(\mathbf{y}) \\ \widehat{A}_\mathcal{H} f(\mathbf{x}) &= \frac{\sum_{i=1}^n K_\epsilon(x,X_i)\langle f, K(\cdot,X_i)\rangle_\mathcal{H}}{\sum_{i=1}^n K_\epsilon(x,X_i)} = \int \widehat{a}_\epsilon(\mathbf{x},\mathbf{y})\langle f, K(\cdot,y)\rangle_\mathcal{H}\ d\widehat{P}_n(\mathbf{y}), \end{aligned}$$

and K is the reproducing kernel of $\mathcal{H}$. Define the following operator norm: $\|A\|_\mathcal{H} = \sup_{f\in\mathcal{H}} \|Af\|_\mathcal{H}/\|f\|_\mathcal{H}$ where $\|f\|_\mathcal{H}^2 = \langle f, f\rangle_\mathcal{H}$. Now suppose the weight function K_ϵ is sufficiently smooth with respect to $\mathcal{H}$ (Assumption 1 in Rosasco et al. 2010); this condition is for example satisfied by a Gaussian kernel on a compact support $\mathcal{X}$. By Propositions 13.3 and 14.3 in Rosasco et al. (2010), we can then relate the functions $\psi_{\epsilon,j}$ and $\widehat{\psi}_{\epsilon,j}$, respectively, to the eigenfunctions $u_{\epsilon,j}$ and $\widehat{u}_{\epsilon,j}$ of $A_\mathcal{H}$ and $\widehat{A}_\mathcal{H}$. We have that

$$\|\psi_{\epsilon,j} - \widehat{\psi}_{\epsilon,j}\|_{L^2(\mathcal{X},P)} = C_1 \|u_{\epsilon,j} - \widehat{u}_{\epsilon,j}\|_{L^2(\mathcal{X},P)} \leqslant C_2 \|u_{\epsilon,j} - \widehat{u}_{\epsilon,j}\|_\mathcal{H} \tag{A.1}$$

for some constants C_1 and C_2. According to Theorem 6 in Rosasco et al. (2008) for eigenprojections of positive compact operators, it holds that

$$\|u_{\epsilon,j} - \widehat{u}_{\epsilon,j}\|_\mathcal{H} \leqslant \frac{\|A_\mathcal{H} - \widehat{A}_\mathcal{H}\|_\mathcal{H}}{\delta_{\epsilon,j}}, \tag{A.2}$$

where $\delta_{\epsilon,j}$ is proportional to the eigengap $\lambda_{\epsilon,j} - \lambda_{\epsilon,j+1}$. As a result, we can bound the difference $\|\psi_{\epsilon,j} - \widehat{\psi}_{\epsilon,j}\|_{L^2(\mathcal{X},P)}$ by controlling the deviation $\|A_\mathcal{H} - \widehat{A}_\mathcal{H}\|_\mathcal{H}$.

1 This auxiliary space only enters the intermediate derivations and plays no role in the error analysis of the algorithm itself.

We choose the auxiliary RKHS $\mathcal{H}$ to be a Sobolev space with a sufficiently high degree of smoothness so that certain assumptions ((A5)-(A6) below) are fulfilled. Let $\mathcal{H}^s$ denote the Sobolev space of order s with vanishing gradients at the boundary; that is, let

$$\mathcal{H}^s = \{f \in L^2(\mathcal{X}) \mid D^\alpha f \in L^2(\mathcal{X}) \ \forall |\alpha| \leqslant s,\ D^\alpha f|_{\partial\mathcal{X}} = 0 \ \forall |\alpha| = 1\},$$

where $D^\alpha f$ is the weak partial derivative of f with respect to the multi-index α, and $L^2(\mathcal{X})$ is the space of square integrable functions with respect to the Lebesgue measure. Let $C_b^3(\mathcal{X})$ be the set of uniformly bounded, three times differentiable functions with uniformly bounded derivatives whose gradients vanish at the boundary. Now suppose that $\mathcal{H} \subset C_b^3(\mathcal{X})$ and that

(A5) $\forall f \in \mathcal{H}, |\alpha| = s, \quad D^\alpha(\widehat{A}_{\mathcal{H}} f - A_{\mathcal{H}} f) = \widehat{A}_{\mathcal{H}} D^\alpha f - A_{\mathcal{H}} D^\alpha f,$

(A6) $\forall f \in \mathcal{H}, |\alpha| = s, \quad D^\alpha f \in C_b^3(\mathcal{X}).$

Under assumptions (A1)-(A6), we have

Lemma A.6. *Let* $\epsilon_n \to 0$ *and* $n\epsilon_n^{d/2}/\log(1/\epsilon_n) \to \infty$. *Then* $\|A_{\mathcal{H}} - \widehat{A}_{\mathcal{H}}\|_{\mathcal{H}} = O_P(\gamma_n)$, *where* $\gamma_n = \sqrt{\frac{\log(1/\epsilon_n)}{n\epsilon_n^{d/2}}}$.

Proof. Uniformly, for all $f \in C_b^3(\mathcal{X})$, and all x in the support of P,

$$|A_\epsilon f(x) - \widehat{A}_\epsilon f(x)| \leqslant |A_\epsilon f(x) - \tilde{A}_\epsilon f(x)| + |\tilde{A}_\epsilon f(x) - \widehat{A}_\epsilon f(x)|$$

where $\tilde{A}_\epsilon f(x) = \int \widehat{a}_\epsilon(x, y) f(y) dP(y)$. From Giné and Guillou (2002),

$$\sup_x \frac{|\widehat{p}_\epsilon(x) - p_\epsilon(x)|}{|\widehat{p}_\epsilon(x) p_\epsilon(x)|} = O_P(\gamma_n).$$

Hence,

$$\begin{aligned} |A_\epsilon f(x) - \tilde{A}_\epsilon f(x)| &\leqslant \frac{|\widehat{p}_\epsilon(x) - p_\epsilon(x)|}{|\widehat{p}_\epsilon(x) p_\epsilon(x)|} \int |f(y)| k_\epsilon(x, y) dP(y) \\ &= O_P(\gamma_n) \int |f(y)| K_\epsilon(x, y) dP(y) \\ &= O_P(\gamma_n). \end{aligned}$$

Next, we bound $\tilde{A}_\epsilon f(x) - \widehat{A}_\epsilon f(x)$. We have

$$\begin{aligned} \tilde{A}_\epsilon f(x) - \widehat{A}_\epsilon f(x) &= \int f(y) \widehat{a}_\epsilon(x, y) (d\widehat{P}_n(y) - dP(y)) \\ &= \frac{1}{p(x) + o_P(1)} \int f(y) K_\epsilon(x, y) (d\widehat{P}_n(y) - dP(y)). \end{aligned}$$

Now, expand $f(\mathbf{y}) = f(\mathbf{x}) + r_n(\mathbf{y})$ where $r_n(\mathbf{y}) = (\mathbf{y}-\mathbf{x})^T \nabla f(u_y)$ and u_y is between $\mathbf{y}$ and $\mathbf{x}$. So,

$$\int f(\mathbf{y}) K_\epsilon(\mathbf{x},\mathbf{y})(d\widehat{P}_n(\mathbf{y}) - dP(\mathbf{y})) =$$
$$f(\mathbf{x}) \int K_\epsilon(\mathbf{x},\mathbf{y})(d\widehat{P}_n(\mathbf{y}) - dP(\mathbf{y})) + \int r_n(\mathbf{y}) K_\epsilon(\mathbf{x},\mathbf{y})(d\widehat{P}_n(\mathbf{y}) - dP(\mathbf{y})).$$

By an application of Talagrand's inequality to each term, as in Theorem 5.1 of Giné and Koltchinskii (2006), we have

$$\int f(\mathbf{y}) K_\epsilon(\mathbf{x},\mathbf{y})(d\widehat{P}_n(\mathbf{y}) - dP(\mathbf{y})) = O_P(\gamma_n).$$

Thus, $\sup_{f \in C_b^3(\mathcal{X})} \|\widehat{A}_\epsilon f - A_\epsilon f\|_\infty = O_P(\gamma_n)$.

The Sobolev space $\mathcal{H}$ is a Hilbert space with respect to the scalar product

$$\langle f, g\rangle_{\mathcal{H}} = \langle f, g\rangle_{L^2(\mathcal{X})} + \sum_{|\alpha|=s} \langle D^\alpha f, D^\alpha g\rangle_{L^2(\mathcal{X})}.$$

Under assumptions (A5)-(A6),

$$\begin{aligned}
\sup_{f\in\mathcal{H}:\|f\|_{\mathcal{H}}=1} \|\widehat{A}_\epsilon f - A_\epsilon f\|_{\mathcal{H}}^2 &\leqslant \sup_{f\in\mathcal{H}} \sum_{|\alpha|\leqslant s} \|D^\alpha(\widehat{A}_\epsilon f - A_\epsilon f)\|_{L^2(\mathcal{X})}^2 \\
&= \sum_{|\alpha|\leqslant s} \sup_{f\in\mathcal{H}} \|\widehat{A}_\epsilon D^\alpha f - A_\epsilon D^\alpha f\|_{L^2(\mathcal{X})}^2 \\
&\leqslant \sum_{|\alpha|\leqslant s} \sup_{f\in C_b^3(\mathcal{X})} \|\widehat{A}_\epsilon f - A_\epsilon f\|_{L^2(\mathcal{X})}^2 \leqslant C \sup_{f\in C_b^3(\mathcal{X})} \|\widehat{A}_\epsilon f - A_\epsilon f\|_\infty^2.
\end{aligned}$$

for some constant C. Hence,

$$\sup_{f\in\mathcal{H}} \frac{\|\widehat{A}_\epsilon f - A_\epsilon f\|_{\mathcal{H}}}{\|f\|_{\mathcal{H}}} = \sup_{f\in\mathcal{H}, \|f\|_{\mathcal{H}}=1} \|\widehat{A}_\epsilon f - A_\epsilon f\|_{\mathcal{H}} \leqslant C' \sup_{f\in C_b^3(\mathcal{X})} \|\widehat{A}_\epsilon f - A_\epsilon f\|_\infty = O_P(\gamma_n).$$

□

For $\epsilon_n \to 0$ and $n\epsilon_n^{d/2}/\log(1/\epsilon_n) \to \infty$, it then holds that:

Proposition A.1. *$\forall 0 \leqslant j \leqslant J$,*

$$\|\psi_{\epsilon,j} - \widehat{\psi}_{\epsilon,j}\|_{L^2(\mathcal{X},P)} = O_P\left(\frac{\gamma_n}{\delta_{\epsilon,j}}\right),$$

where $\delta_{\epsilon,j} = \lambda_{\epsilon,j} - \lambda_{\epsilon,j+1}$.

Proof. From Lemma A.1 and Equation A.2, we have that

$$\|\psi_{\epsilon,j} - \widehat{\psi}_{\epsilon,j}\|_{\epsilon} \leqslant \sqrt{\frac{b}{a}} \|\psi_{\epsilon,j} - \widehat{\psi}_{\epsilon,j}\|_{L^2(\mathcal{X},P)} \leqslant C \frac{\|A_{\mathcal{H}} - \widehat{A}_{\mathcal{H}}\|_{\mathcal{H}}}{\lambda_{\epsilon,j} - \lambda_{\epsilon,j+1}}$$

for some constant C that does not depend on n. The result follows from Lemma A.6.

□

Using this, we can then bound the estimated coefficients.

Lemma A.7. $\forall 0 \leqslant j \leqslant J$,

$$|\widehat{\beta}_{\epsilon,j} - \beta_{\epsilon,j}|^2 = O_P\left(\frac{1}{n}\right) + O_P\left(\frac{\gamma_n^2}{\delta_{\epsilon,j}^2}\right).$$

Proof. Note that $\psi_{\epsilon,j}(\mathbf{x})s_{\epsilon}(\mathbf{x}) = \varphi_{\epsilon,j}(\mathbf{x})$ and

$$\begin{aligned}
\widehat{\beta}_{\epsilon,j} &= \frac{1}{n}\sum_{i=1}^{n} Y_i \widehat{\psi}_{\epsilon,j}(\mathbf{X}_i)\widehat{s}_{\epsilon}(\mathbf{X}_i) \\
&= \frac{1}{n}\sum_{i=1}^{n} Y_i \varphi_{\epsilon,j}(\mathbf{X}_i) + \frac{1}{n}\sum_{i=1}^{n} Y_i \left(\widehat{\varphi}_{\epsilon,j}(\mathbf{X}_i) - \varphi_{\epsilon,j}(\mathbf{X}_i)\right) \\
&= \beta_{\epsilon,j} + O_P\left(\frac{1}{\sqrt{n}}\right) + \frac{1}{n}\sum_{i=1}^{n} Y_i \left(\widehat{\varphi}_{\epsilon,j}(\mathbf{X}_i) - \varphi_{\epsilon,j}(\mathbf{X}_i)\right).
\end{aligned}$$

Let $S = \frac{1}{n}\sum_{i=1}^{n} Y_i \left(\widehat{\varphi}_{\epsilon,j}(\mathbf{X}_i) - \varphi_{\epsilon,j}(\mathbf{X}_i)\right)$ and $I = \int_{\mathcal{X}} r(\mathbf{x})(\widehat{\varphi}_{\epsilon,j}(\mathbf{x}) - \varphi_{\epsilon,j}(\mathbf{x}))dP(\mathbf{x})$. We conclude that

$$\begin{aligned}
\frac{1}{2}|\widehat{\beta}_{\epsilon,j} - \beta_{\epsilon,j}|^2 &\leqslant O_P\left(\frac{1}{n}\right) + |S - I|^2 + |I|^2 \\
&\leqslant O_P\left(\frac{1}{n}\right) + |S - I|^2 + \left(\int_{\mathcal{X}} |r(\mathbf{x})|^2 dP(\mathbf{x})\right)\left(\int_{\mathbf{X}} |\varphi_{\epsilon,j}(\mathbf{x}) - \widehat{\varphi}_{\epsilon,j}(\mathbf{x})|^2 dP(\mathbf{x})\right) \\
&= O_P\left(\frac{1}{n}\right) + O_P\left(\frac{\gamma_n^2}{\delta_{\epsilon,j}^2}\right),
\end{aligned}$$

where the second inequality follows from Cauchy-Schwartz, and the last equality is due to Lemmas A.5 and A.4. □

□

We can now prove Proposition 3.2:

Proof. Let $\tilde{r}_{\epsilon,J}(\mathbf{x}) = \sum_{j=0}^{J} \beta_{\epsilon,j}\widehat{\psi}_{\epsilon,j}(\mathbf{x})$. Write

$$\begin{aligned} |r_{\epsilon,J}(\mathbf{X}_i) - \widehat{r}_{\epsilon,J}(\mathbf{X}_i)|^2 &= |r_{\epsilon,J}(\mathbf{X}_i) - \tilde{r}_{\epsilon,J}(\mathbf{X}_i) + \tilde{r}_{\epsilon,J}(\mathbf{X}_i) - \widehat{r}_{\epsilon,J}(\mathbf{X}_i)|^2 \\ &\leqslant 2|r_{\epsilon,J}(\mathbf{X}_i) - \tilde{r}_{\epsilon,J}(\mathbf{X}_i)|^2 + 2|\tilde{r}_{\epsilon,J}(\mathbf{X}_i) - \widehat{r}_{\epsilon,J}(\mathbf{X}_i)|^2 \end{aligned}$$

We bound the contribution to L_{var} from each of these two terms separately:

By using Cauchy's inequality and Proposition A.1, we have that

$$\int_{\mathcal{X}} |r_{\epsilon,J}(\mathbf{x}) - \tilde{r}_{\epsilon,J}(\mathbf{x})|^2 dP(\mathbf{x}) = \int_{\mathcal{X}} \left| \sum_{j=0}^{J} \beta_{\epsilon,j}(\psi_{\epsilon,j}(\mathbf{x}) - \widehat{\psi}_{\epsilon,j}(\mathbf{x})) \right|^2 dP(\mathbf{x})$$

$$\leqslant \left(\sum_{j=0}^{J} |\beta_{\epsilon,j}|^2 \right) \cdot \sum_{j=0}^{J} \left(\int_{\mathcal{X}} |\psi_{\epsilon,j}(\mathbf{x}) - \widehat{\psi}_{\epsilon,j}(\mathbf{x})|^2 dP(\mathbf{x}) \right) = J\, O_P\left(\frac{\gamma_n^2}{\Delta_{\epsilon,J}^2} \right).$$

By construction, it holds that $\frac{1}{n}\sum_i \widehat{\psi}_{\epsilon,j}(\widetilde{X}_i)\widehat{\psi}_{\epsilon,\ell}(\tilde{X}_i)\widehat{s}_\epsilon(\tilde{X}_i) = \delta_{j,\ell}$. Furthermore,

$$\begin{aligned} \int_{\mathcal{X}} \widehat{\psi}_{\epsilon,j}(\mathbf{x})\widehat{\psi}_{\epsilon,\ell}(\mathbf{x}) d\widehat{S}_\epsilon(\mathbf{x}) &= \frac{1}{n}\sum_i \widehat{\psi}_{\epsilon,j}(\mathbf{X}_i)\widehat{\psi}_{\epsilon,\ell}(\mathbf{X}_i)\widehat{s}_\epsilon(\mathbf{X}_i) + O_P\left(\frac{1}{\sqrt{n}}\right) \\ &= \frac{1}{n}\sum_i \widehat{\psi}_{\epsilon,j}(\tilde{\mathbf{X}}_i)\widehat{\psi}_{\epsilon,\ell}(\tilde{\mathbf{X}}_i)\widehat{s}_\epsilon(\tilde{\mathbf{X}}_i) + O_P\left(\frac{1}{\sqrt{n}}\right) \\ &= \delta_{j,\ell} + O_P\left(\frac{1}{\sqrt{n}}\right). \end{aligned}$$

for a sample $\mathbf{X}_1, \ldots, \mathbf{X}_n$ drawn independently from $\tilde{\mathbf{X}}_1, \ldots, \tilde{\mathbf{X}}_n$. Finally, from the orthogonality property of the $\widehat{\psi}_{\epsilon,j}$'s together with Lemmas A.7 and A.1, it follows that

$$\int_{\mathcal{X}} |\tilde{r}_{\epsilon,J}(\mathbf{x}) - \widehat{r}_{\epsilon,J}(\mathbf{x})|^2 dP(\mathbf{x}) = \int_{\mathcal{X}} \frac{1}{\widehat{s}_\epsilon(\mathbf{x})} \left| \sum_{j=0}^{J} (\beta_{\epsilon,j} - \widehat{\beta}_{\epsilon,j})\widehat{\psi}_{\epsilon,j}(\mathbf{x})\sqrt{\widehat{s}_\epsilon(\mathbf{x})} \right|^2 dP(\mathbf{x})$$

$$= \int_{\mathcal{X}} \frac{1}{\widehat{s}_\epsilon(\mathbf{x})} \left(\sum_{j=0}^{J} (\beta_{\epsilon,j} - \widehat{\beta}_{\epsilon,j})^2 \widehat{\psi}_{\epsilon,j}^2(\mathbf{x}) d\widehat{S}_\epsilon(\mathbf{x}) \right)$$

$$+ \int_{\mathcal{X}} \frac{1}{\widehat{s}_\epsilon(\mathbf{x})} \left(\sum_{j=0}^{J} \sum_{\ell=0, \ell\neq j}^{J} (\beta_{\epsilon,j} - \widehat{\beta}_{\epsilon,j})(\beta_{\epsilon,\ell} - \widehat{\beta}_{\epsilon,\ell})\widehat{\psi}_{\epsilon,j}(\mathbf{x})\widehat{\psi}_{\epsilon,\ell}(\mathbf{x}) d\widehat{S}_\epsilon(\mathbf{x}) \right)$$

$$
\begin{aligned}
&\leqslant \frac{b}{a}\sum_{j=0}^{J}(\beta_{\epsilon,j}-\widehat{\beta}_{\epsilon,j})^2\left(\int_{\mathcal{X}}\widehat{\psi}^2_{\epsilon,j}(\mathbf{x})d\widehat{S}_\epsilon(\mathbf{x})\right)\\
&+ \frac{b}{a}\sum_{j=0}^{J}\sum_{\ell=0,\ell\neq j}^{J}(\beta_{\epsilon,j}-\widehat{\beta}_{\epsilon,j})(\beta_{\epsilon,\ell}-\widehat{\beta}_{\epsilon,\ell})\left(\int_{\mathcal{X}}\widehat{\psi}_{\epsilon,j}(\mathbf{x})\widehat{\psi}_{\epsilon,\ell}(\mathbf{x})d\widehat{S}_\epsilon(\mathbf{x})\right)\\
&= \frac{b}{a}\sum_{j=0}^{J}(\beta_{\epsilon,j}-\widehat{\beta}_{\epsilon,j})^2\left(1+O_P\left(\frac{1}{\sqrt{n}}\right)\right)\\
&+ \frac{b}{a}\sum_{j=0}^{J}\sum_{\ell=0,\ell\neq j}^{J}(\beta_{\epsilon,j}-\widehat{\beta}_{\epsilon,j})(\beta_{\epsilon,\ell}-\widehat{\beta}_{\epsilon,\ell})\,O_P\left(\frac{1}{\sqrt{n}}\right)\\
&= J\left(O_P\left(\frac{1}{n}\right)+O_P\left(\frac{\gamma_n^2}{\Delta^2_{\epsilon,J}}\right)\right).
\end{aligned}
$$

The result follows. □

We now prove of Corollary 3.2.

Proof. The results of Coifman and Lafon (2006) and Giné and Koltchinskii (2006) form the basis of our Lemma A.3 and Lemma A.6. Because their results apply to the p-dimensional submanifold case, the term $\gamma_n = \sqrt{\frac{\log(1/\epsilon_n)}{n\epsilon_n^{d/2}}}$ in our derivations becomes $\gamma_n = \sqrt{\frac{\log(1/\epsilon_n)}{n\epsilon_n^{p/2}}}$. Furthermore, the eigenvalues of the Laplace-Beltrami operator $\triangle$ on an p-dimensional Riemannian manifold are $\nu_j^2 \sim j^{2/p}$ (Safarov and Vassilev 1997). Under the given assumptions, we then have that

$$
L(r,\widehat{r}) = O\left(\frac{1}{J^{2/p}}\right)+\frac{J}{(J+1)^{2/p}-J^{2/p}}O_P\left(\frac{\log n}{n}\right)^{\frac{2}{p+4}}.
$$

Using the mean- value theorem, one can show that $\frac{J}{(J+1)^{2/p}-J^{2/p}} = O(J^{2(1-1/p)})$. The main result follows. □

B

APPENDIX: BOUNDS ON THE CONDITIONAL DENSITY ESTIMATOR

We now present the proofs for the bounds from Chapter 4.

To simplify the proofs, we assume the functions $\psi_1, \ldots, \psi_J$ are estimated using an unlabeled sample $\widetilde{\mathbf{X}}_1, \ldots, \widetilde{\mathbf{X}}_m$, drawn independently from the sample used to estimate the coefficients $\beta_{i,j}$. Without loss of generality, this can be achieved by splitting the labeled sample in two. This split is only for theoretical purposes; in practice using all data to estimate the basis leads to better results. The technique also allows us to derive bounds for the semi-supervised learning setting described in the paper, and better understand the additional cost of estimating the basis. Define the following quantities:

$$f_{I,J}(z|\mathbf{x}) = \sum_{i=1}^{I}\sum_{j=1}^{J} \beta_{i,j}\phi_i(z)\psi_j(\mathbf{x}), \quad \beta_{i,j} = \iint \phi_i(z)\psi_j(\mathbf{x})f(z,\mathbf{x})\,d\mathbf{x}\,dz$$

$$\widehat{f}_{I,J}(z|\mathbf{x}) = \sum_{i=1}^{I}\sum_{j=1}^{J} \widehat{\beta}_{i,j}\phi_i(z)\widehat{\psi}_j(\mathbf{x}), \quad \widehat{\beta}_{i,j} = \frac{1}{n}\sum_{k=1}^{n} \phi_i(z_k)\widehat{\psi}_j(\mathbf{x}_k)$$

and note that

$$\begin{aligned}
&\iint \left(\widehat{f}_{I,J}(z|\mathbf{x}) - f(z|\mathbf{x})\right)^2 dP(\mathbf{x})\,dz \\
&\leqslant \iint \left(\widehat{f}_{I,J}(z|\mathbf{x}) - f_{I,J}(z|\mathbf{x}) + f_{I,J}(z|\mathbf{x}) - f(z|\mathbf{x})\right)^2 dP(\mathbf{x})\,dz \\
&\leqslant 2\left(VAR(\widehat{f}_{I,J}, f_{I,J}) + B(f_{I,J}, f)\right). \qquad \text{(B.1)}
\end{aligned}$$

where $B(f_{I,J}, f) := \int\int (f_{I,J}(z|\mathbf{x}) - f(z|\mathbf{x}))^2 \, dP(\mathbf{x})dz$ can be interpreted as a bias term (or approximation error) and $VAR(\widehat{f}_{I,J}, f_{I,J}) := \int\int \left(\widehat{f}_{I,J}(z|\mathbf{x}) - f_{I,J}(z|\mathbf{x})\right)^2 dP(\mathbf{x})dz$ can be interpreted as a variance term . First we bound the variance.

Lemma B.1. $\forall 1 \leqslant j \leqslant J$,

$$\int \left(\widehat{\psi_j}(\mathbf{x}) - \psi_j(\mathbf{x})\right)^2 dP(\mathbf{x}) = O_P\left(\frac{1}{\lambda_j \delta_j^2 m}\right),$$

where $\delta_j = \lambda_j - \lambda_{j+1}$.

For a proof of Lemma B.1 see for example Sinha and Belkin (2009).

Lemma B.2. $\forall 1 \leqslant j \leqslant J$, *there exists* $C < \infty$ *that does not depend on* m *such that*

$$E\left[\left(\widehat{\psi}_j(\mathbf{X}) - \psi_j(\mathbf{X})\right)^2\right] < C,$$

where $\mathbf{X} \sim P(\mathbf{x})$ *is independent of the sample used to construct* $\widehat{\psi}_j$.

Proof. Let $\delta \in (0, 1)$. From Sinha and Belkin (2009), it follows that

$$\mathbb{P}\left(\int \left(\widehat{\psi}_j(\mathbf{x}) - \psi_j(\mathbf{x})\right)^2 dP(\mathbf{x}) > \frac{16 \log\left(\frac{2}{\delta}\right)}{\delta_j^2 m}\right) < \delta,$$

and therefore $\forall \epsilon > 0$,

$$\mathbb{P}\left(\int \left(\widehat{\psi}_j(\mathbf{x}) - \psi_j(\mathbf{x})\right)^2 dP(\mathbf{x}) > \epsilon\right) < 2e^{-\frac{\delta_j^2 m \epsilon}{16}}.$$

Hence

$$\mathbb{E}\left[\left(\widehat{\psi}_j(\mathbf{X}) - \psi_j(\mathbf{X})\right)^2\right] = \mathbb{E}\left[\int \left(\widehat{\psi}_j(\mathbf{x}) - \psi_j(\mathbf{x})\right)^2 dP(\mathbf{x})\right] =$$
$$\int_0^\infty \mathbb{P}\left(\int \left(\widehat{\psi}_j(\mathbf{x}) - \psi_j(\mathbf{x})\right)^2 dP(\mathbf{x}) > \epsilon\right) d\epsilon \leqslant \int 2e^{-\frac{\delta_j^2 m \epsilon}{16}} d\epsilon < \int 2e^{-\frac{\delta_j^2 \epsilon}{16}} d\epsilon < \infty$$

□

Lemma B.3. $\forall 1 \leqslant j \leqslant J$ *and* $\forall 1 \leqslant j \leqslant J$, *there exists* $C < \infty$ *that does not depend on* m *such that*

$$\mathbb{E}\left[\mathbb{V}\left[\phi_i(Z)\left(\widehat{\psi}_j(\mathbf{X}) - \psi_j(\mathbf{X})\right) \middle| \widetilde{\mathbf{X}}_1, \ldots, \widetilde{\mathbf{X}}_m\right]\right] < C$$

Proof. Using that ϕ is bounded (Assumption 4.2), it follows that

$$\begin{aligned}
&\mathbb{E}\left[\mathbb{V}\left[\phi_i(Z)\left(\widehat{\psi}_j(\mathbf{X})-\psi_j(\mathbf{X})\right)\middle|\widetilde{\mathbf{X}}_1,\ldots,\widetilde{\mathbf{X}}_m\right]\right]\\
&\leqslant \mathbb{V}\left[\phi_i(Z)\left(\widehat{\psi}_j(\mathbf{X})-\psi_j(\mathbf{X})\right)\right] \leqslant \mathbb{E}\left[\phi_i^2(Z)\left(\widehat{\psi}_j(\mathbf{X})-\psi_j(\mathbf{X})\right)^2\right]\\
&\leqslant K\mathbb{E}\left[\left(\widehat{\psi}_j(\mathbf{X})-\psi_j(\mathbf{X})\right)^2\right]
\end{aligned}$$

for some $K<\infty$. The result follows from Lemma B.2. □

Lemma B.4. *$\forall 1\leqslant j\leqslant J$ and $\forall 1\leqslant j\leqslant J$,*

$$\left[\frac{1}{n}\sum_{k=1}^{n}\phi_i(Z_k)\left(\widehat{\psi}_j(\mathbf{X}_k)-\psi_j(\mathbf{X}_k)\right)-\iint\phi_i(z)\left(\widehat{\psi}_j(\mathbf{x})-\psi_j(\mathbf{x})\right)dP(z,\mathbf{x})\right]^2=O_P\left(\frac{1}{n}\right)$$

Proof. Let $A=\iint\phi_i(z)\left(\widehat{\psi}_j(\mathbf{x})-\psi_j(\mathbf{x})\right)dP(z,\mathbf{x})$. By Chebyshev's inequality it holds that $\forall M>0$

$$\begin{aligned}
&\mathbb{P}\left(\left|\frac{1}{n}\sum_{k=1}^{n}\phi_i(Z_k)\left(\widehat{\psi}_j(\mathbf{X}_k)-\psi_j(\mathbf{X}_k)\right)-A\right|^2>M\middle|\widetilde{\mathbf{X}}_1,\ldots,\widetilde{\mathbf{X}}_m\right)\leqslant\\
&\frac{1}{nM}\mathbb{V}\left[\phi_i(Z)\left(\widehat{\psi}_j(\mathbf{X})-\psi_j(\mathbf{X})\right)\middle|\widetilde{\mathbf{X}}_1,\ldots,\widetilde{\mathbf{X}}_m\right].
\end{aligned}$$

The conclusion follows from taking an expectation with respect to the unlabeled samples on both sides of the equation and using Lemma B.3.

□

Note that $\widehat{\psi}'$s are random functions, and therefore the proof of Lemma B.4 relies on the fact that these functions are estimated using a different sample than $\mathbf{X}_1,\ldots,\mathbf{X}_n$.

Lemma B.5. *$\forall 1\leqslant j\leqslant J$ and $\forall 1\leqslant i\leqslant I$,*

$$\left(\widehat{\beta}_{i,j}-\beta_{i,j}\right)^2=O_P\left(\frac{1}{n}\right)+O_P\left(\frac{1}{\lambda_j\delta_j^2 m}\right).$$

Proof. It holds that

$$\frac{1}{2}\left(\widehat{\beta}_{i,j}-\beta_{i,j}\right)^2\leqslant\left(\frac{1}{n}\sum_{k=1}^{n}\phi_i(Z_k)\psi_j(\mathbf{X}_k)-\beta_{i,j}\right)^2+\left(\frac{1}{n}\sum_{k=1}^{n}\phi_i(Z_k)(\widehat{\psi}_j(\mathbf{X}_k)-\psi_j(\mathbf{X}_k))\right)^2.$$

The first term is $O_P\left(\frac{1}{n}\right)$. Let A be as in the proof of Lemma B.4. By using Cauchy-Schwartz and that Lemma, the second term divided by two is bounded by

$$\begin{aligned}
&\frac{1}{2}\left(\frac{1}{n}\sum_{k=1}^{n}\phi_i(Z_k)\left(\widehat{\psi}_j(\mathbf{X}_k)-\psi_j(\mathbf{X}_k)\right)-A+A\right)^2\\
&\leqslant\left(\frac{1}{n}\sum_{k=1}^{n}\phi_i(Z_k)\left(\widehat{\psi}_j(\mathbf{X}_k)-\psi_j(\mathbf{X}_k)\right)-A\right)^2+A^2.\\
&\leqslant O_P\left(\frac{1}{n}\right)+\left(\iint\phi_i^2(z)dP(z,\mathbf{x})\right)\left(\iint\left(\widehat{\psi}_j(\mathbf{x})-\psi_j(\mathbf{x})\right)^2dP(z,\mathbf{x})\right).
\end{aligned}$$

The result follows from Lemma B.1 and the orthogonality of ϕ_i.

□

Lemma B.6. *[Sinha and Belkin 2009, Corollary 1] Under the stated assumptions,*

$$\int\widehat{\psi}_j^2(\mathbf{x})dP(\mathbf{x})=O_P\left(\frac{1}{\lambda_j\Delta_J^2m}\right)+1$$

and

$$\int\widehat{\psi}_i(\mathbf{x})\widehat{\psi}_j(\mathbf{x})dP(\mathbf{x})=O_P\left(\left(\frac{1}{\sqrt{\lambda_i}}+\frac{1}{\sqrt{\lambda_j}}\right)\frac{1}{\Delta_J\sqrt{m}}\right)$$

where $\Delta_J=\min_{1\leqslant j\leqslant J}\delta_j$.

Lemma B.7. *Let* $h(z|\mathbf{x})=\sum_{i=1}^{I}\sum_{j=1}^{J}\beta_{i,j}\phi_i(z)\widehat{\psi}_j(\mathbf{x})$. *Then*

$$\iint\left|\widehat{f}_{I,J}(z|\mathbf{x})-h(z|\mathbf{x})\right|^2dP(\mathbf{x})dz=IJ\left(O_P\left(\frac{1}{n}\right)+O_P\left(\frac{1}{\lambda_J\Delta_J^2m}\right)\right).$$

Proof.

$$\begin{aligned}
&\iint \left|\widehat{f}_{I,J}(z|\mathbf{x}) - h(z|\mathbf{x})\right|^2 dP(\mathbf{x})dz \\
&= \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{l=1}^{J}\left(\widehat{\beta}_{i,j} - \beta_{i,j}\right)\left(\widehat{\beta}_{i,l} - \beta_{i,l}\right)\int \widehat{\psi_j}(\mathbf{x})\widehat{\psi_l}(\mathbf{x})dP(\mathbf{x}) \\
&\leqslant \sum_{i=1}^{I}\sum_{j=1}^{J}\left(\widehat{\beta}_{i,j} - \beta_{i,j}\right)^2\int \widehat{\psi_j}^2(\mathbf{x})dP(\mathbf{x}) + \\
&+ \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{l=1,l\neq j}^{J}\left(\widehat{\beta}_{i,j} - \beta_{i,j}\right)\left(\widehat{\beta}_{i,l} - \beta_{i,l}\right)\int \widehat{\psi_j}(\mathbf{x})\widehat{\psi_l}(\mathbf{x})dP(\mathbf{x}) \leqslant \\
&\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\widehat{\beta}_{i,j} - \beta_{i,j}\right)^2\int \widehat{\psi_j}^2(\mathbf{x})dP(\mathbf{x}) + \\
&+ \left[\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\widehat{\beta}_{i,j} - \beta_{i,j}\right)^2\right]\left[\sqrt{\sum_{j=1}^{J}\sum_{l=1,l\neq j}^{J}\left(\int \widehat{\psi_j}(\mathbf{x})\widehat{\psi_l}(\mathbf{x})dP(\mathbf{x})\right)^2}\right],
\end{aligned}$$

where the last inequality follows from repeatedly using Cauchy-Schwartz. The result follows from Lemmas B.5 and B.6.

□

Lemma B.8. *Let* $h(z|x)$ *be as in Lemma B.7. Then*

$$\iint |h(z|\mathbf{x}) - f_{I,J}(z|\mathbf{x})|^2 dP(\mathbf{x})dz = JO_P\left(\frac{1}{\lambda_J \Delta_J^2 m}\right).$$

Proof. Using Cauchy-Schwartz inequality,

$$\begin{aligned}
&\iint |h(z|\mathbf{x}) - f_{I,J}(z|\mathbf{x})|^2 dP(\mathbf{x})dz \leqslant \iint \left|\sum_{i=1}^{I}\sum_{j=1}^{J}\beta_{i,j}\phi_i(z)\left(\psi_j(\mathbf{x}) - \widehat{\psi_j}(\mathbf{x})\right)\right|^2 dP(\mathbf{x})dz \\
&\leqslant \left\{\sum_{j=1}^{J}\int\left[\sum_{i=1}^{I}\beta_{i,j}\phi_i(z)\right]^2 dz\right\}\left\{\sum_{j=1}^{J}\int\left[\psi_j(\mathbf{x}) - \widehat{\psi_j}(\mathbf{x})\right]^2 dP(\mathbf{x})\right\} \\
&= \left\{\sum_{j=1}^{J}\sum_{i=1}^{I}\beta_{i,j}^2\right\}\left\{\sum_{j=1}^{J}\int\left[\psi_j(\mathbf{x}) - \widehat{\psi_j}(\mathbf{x})\right]^2 dP(\mathbf{x})\right\}.
\end{aligned}$$

The conclusion follows from Lemma B.1 and by noticing that $\sum_{j=1}^{J}\sum_{i=1}^{I}\beta_{i,j}^2 \leqslant ||f(z|\mathbf{x})||^2 < \infty$.

□

It is now possible to bound the variance term:

Theorem B.1. *Under the stated assumptions,*

$$VAR(\widehat{f}_{I,J}, f_{I,J}) = IJ\left(O_P\left(\frac{1}{n}\right) + O_P\left(\frac{1}{\lambda_J \Delta_j^2 m}\right)\right).$$

Proof. Let h be defined as in Lemma B.7. We have

$$\begin{aligned}\frac{1}{2}VAR(\widehat{f}_{I,J}, f_{I,J}) &= \frac{1}{2}\iint \left|\widehat{f}_{I,J}(z|\mathbf{x}) - h(z|\mathbf{x}) + h(z|\mathbf{x}) - f_{I,J}(z|\mathbf{x})\right|^2 dP(\mathbf{x})dz \\ &\leqslant \iint \left|\widehat{f}_{I,J}(z|\mathbf{x}) - h(z|\mathbf{x})\right|^2 dP(\mathbf{x})dz + \iint |h(z|\mathbf{x}) - f_{I,J}(z|\mathbf{x})|^2 dP(\mathbf{x})dz.\end{aligned}$$

The conclusion follows from Lemmas B.7 and B.8.

□

We now bound the bias term.

Lemma B.9. *For each $z \in [0,1]$, expand $g_z(\mathbf{x})$ in the basis ψ : $g_z(\mathbf{x}) = \sum_{j\geqslant 1} \alpha_j^z \psi_j(\mathbf{x})$, where $\alpha_j^z = \int g_z(\mathbf{x})\psi_j(\mathbf{x})dP(\mathbf{x})$. We have*

$$\alpha_j^z = \sum_{i\geqslant 1} \beta_{i,j}\phi_i(z) \text{ and } \int \left(\alpha_j^z\right)^2 dz = \sum_{i\geqslant 1} \beta_{i,j}^2.$$

Proof. It follows from projecting α_j^z onto the basis ϕ. □

Similarly, we have the following.

Lemma B.10. *For each $\mathbf{x} \in \mathcal{X}$, expand $h_{\mathbf{x}}(z)$ in the basis ϕ : $h_{\mathbf{x}}(z) = \sum_{i\geqslant 1} \alpha_i^{\mathbf{x}} \phi_i(z)$, where $\alpha_i^{\mathbf{x}} = \int h_{\mathbf{x}}(z)\phi_i(z)dz$. We have*

$$\alpha_i^{\mathbf{x}} = \sum_{j\geqslant 1} \beta_{i,j}\psi_i(\mathbf{x}) \text{ and } \int (\alpha_i^{\mathbf{x}})^2 dP(\mathbf{x}) = \sum_{j\geqslant 1} \beta_{i,j}^2.$$

Lemma B.11. *Using the same notation as Lemmas B.9 and B.10, we have*

$$\beta_{i,j} = \int \alpha_i^{\mathbf{x}}\psi_j(\mathbf{x})dP(\mathbf{x}) = \int \alpha_j^z \phi_i(z)dz.$$

Proof. Follows from plugging the definitions of $\alpha_i^{\mathbf{x}}$ and α_j^z into the expressions above and recalling the definition of $\beta_{i,j}$. □

Lemma B.12. $\sum_{i\geqslant I}\int \left(\alpha_i^x\right)^2 dP(\mathbf{x}) = O\left(\frac{1}{I^{2s}}\right)$.

Proof. By Lemma, B.10, $h_x(z) = \sum_{i\geqslant 1}\alpha_i^x\phi_i(z)$. As by Assumption 4.4 $h_x \in W_\phi(s_x, c_x)$,

$$\sum_{i\geqslant I} I^{2s_x}\left(\alpha_i^x\right)^2 \leqslant \sum_{i\geqslant I} i^{2s_x}\left(\alpha_i^x\right)^2 \leqslant c_x^2.$$

Hence

$$\sum_{i\geqslant I}\int \left(\alpha_i^x\right)^2 dP(\mathbf{x}) \leqslant \int \frac{c_x^2}{I^{2s_x}} dP(\mathbf{x}) \leqslant \frac{1}{I^{2\beta}} c^2.$$

□

Lemma B.13. $\sum_{j\geqslant J}\int \left(\alpha_j^z\right)^2 dz = c_K O(\lambda_J)$.

Proof. Note that $\|h_z(.)\|_{\mathcal{H}_K}^2 = \sum_{j\geqslant 1}\frac{\left(\alpha_j^z\right)^2}{\lambda_j}$ (Minh 2010). Using Assumption 4.5 and that the eigenvalues are decreasing it follows that

$$\sum_{j\geqslant J}\left(\alpha_j^z\right)^2 = \sum_{j\geqslant J}\left(\alpha_j^z\right)^2\frac{\lambda_j}{\lambda_j} \leqslant \lambda_J\|h_z(.)\|_{\mathcal{H}}^2 \leqslant \lambda_J c_z^2,$$

and therefore $\sum_{j\geqslant J}\int \left(\alpha_j^z\right)^2 dz \leqslant \lambda_J\int_z c_z^2 dz = c_K O(\lambda_J)$.

□

Theorem B.2. *Under the stated assumptions, the bias is bounded by*

$$B(f_{I,J}, f) = c_K O(\lambda_J) + O\left(\frac{1}{I^{2\beta}}\right).$$

Proof. By using orthogonality, we have that

$$\begin{aligned} B(f_{I,J}, f) &\stackrel{\text{def}}{=} \iint \left(f(z|\mathbf{x}) - f_{I,J}(z|\mathbf{x})\right)^2 dP(\mathbf{x})dz \leqslant \sum_{j>J}\sum_{i\geqslant 1}\beta_{i,j}^2 + \sum_{i>I}\sum_{j\geqslant 1}\beta_{i,j}^2 \\ &= \sum_{j\geqslant J}\int \left(\alpha_j^z\right)^2 dz + \sum_{i\geqslant I}\int \left(\alpha_i^x\right)^2 dP(\mathbf{x}), \end{aligned}$$

where the last equality comes from Lemmas B.9 and B.10. The theorem follows from Lemmas B.12 and B.13. □

The main Theorem of Chapter 4 for the kernel PCA basis follows from putting together Theorems B.1 and B.2 using the bias-variance decomposition of Equation B.1.

The proofs for the diffusion kernel follow the same strategy from those presented in Chapter 3. See also Izbicki and Lee (2014) for more details.

BIBLIOGRAPHY

F. Abramovich and B. W. Silverman. The vaguelette-wavelet decomposition approach to statistical inverse problems. *Biometrika*, 85:115–129, 1998.

F. Abramovich, T. C Bailey, and T. Sapatinas. Wavelet analysis and its statistical applications. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 49(1): 1–29, 2000.

H. Aihara, C. A Prieto, D. An, et al. The eighth data release of the Sloan Digital Sky Survey: first data from SDSS-III. *The Astrophysical Journal Supplement Series*, 193(2):29, 2011.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

A. Aswani, P. Bickel, and C. Tomlin. Regression on manifolds: Estimation of the exterior derivative. *The Annals of Statistics*, 39(1):48–81, 2011.

N.M. Ball and R.J. Brunner. Data mining and machine learning in astronomy. *International Journal of Modern Physics D*, 19:1049–1106, 2010.

D.M. Bashtannyk and R.J. Hyndman. Bandwidth selection for kernel conditional density estimation. *Computational Statistics & Data Analysis*, 36:279–298, 2000.

M. A. Beaumont. Approximate Bayesian computation in evolution and ecology. *Annual Review of Ecology, Evolution, and Systematics*, 41:379–406, 2010.

M. Belkin and P. Niyogi. Towards a theoretical foundation for Laplacian-based manifold methods. In *Proceedings of the 18th Annual Conference on Learning Theory*, volume 18, pages 486–500. Springer, 2005.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 6(15):1373–1396, 2003.

M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, volume 14, pages 585–591. The MIT Press, 2001.

R. Bellman. *Adaptive control processes: a guided tour*. Princeton University Press, 1961.

Y. Bengio, O. Delalleau, N. Le Roux, et al. Learning eigenfunctions links Spectral Embedding and Kernel PCA. *Neural Computation*, 16(10):2197–2219, 2004.

A. Bera and A. Ghosh. Neyman's smooth test and its applications in econometrics. *Handbook Of Applied Econometrics And Statistical Inference*, 165:177–230, 2002.

P. J. Bickel and B. Li. Local polynomial regression on unknown manifolds. In *IMS Lecture Notes–Monograph Series, Complex Datasets and Inverse Problems*, volume 54, pages 177–186. Institute of Mathematical Statisitcs, 2007.

O. Bousquet, O. Chapelle, and M. Hein. Measure based regularization. In *Advances in Neural Information Processing Systems 16*. Cambridge, MA, MIT Press, 2003.

S. Bridle, J. Shawe-Taylor, A. Amara, et al. Handbook for the GREAT08 Challenge: An image analysis competition for cosmological lensing. *The Annals of Applied Statistics*, 3(1):6–37, 2009.

S. Buchman. *High-Dimensional Adaptive Basis Density Estimation*. PhD thesis, Carnegie Mellon University, 2011.

S. M. Buchman, A. B. Lee, and C. M. Schafer. High-dimensional density estimation via sca: An example in the modelling of hurricane tracks. *Statistical Methodology*, 8(1):18–30, 2011.

E. Cameron and A.N. Pettitt. Approximate Bayesian computation for astronomical model analysis: a case study in galaxy demographics and morphological trans-

formation at high redshift. *Monthly Notices of the Royal Astronomical Society*, 425 (1):44–65, 2012.

E. Candès and T. Tao. The Dantzig selector: statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351, 2005.

G. Chagny. Warped bases for conditional density estimation. *Mathematical Methods of Statistics*, 22(4):253–282, 2013.

M. Y. Cheng and H. T. Wu. Local linear regression on manifolds and its geometric interpretation. *Journal of the American Statistical Association*, 108:1421–1434, 2013.

R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21:5–30, 2006.

R. R. Coifman, S. Lafon, A. B. Lee, et al. Geometric diffusions as a tool for harmonics analysis and structure definition of data: Diffusion maps. *Proc. of the National Academy of Sciences*, 102(21):7426–7431, 2005.

C. J. Conselice. The relationship between stellar light distributions of galaxies and their formation histories. *The Astrophysical Journal Supplement Series*, 147(1):1–28, 2003.

V. Corradi and N. R. Swanson. Predictive density evaluation. In *Handbook of Economic Forecasting*. North-Holland, 2006.

F. Cucker and D.X. Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, 2007.

C. E. Cunha, M. Lima, H. Oyaizu, J. Frieman, and H. Lin. Estimating the redshift distribution of photometric galaxy samples — II. Applications and tests of a new method. *Monthly Notices of the Royal Astronomical Society*, (396):2379–2398, 2009.

P. J. Diggle and R. J. Gratton. Monte carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(2): 193–227, 1984.

D. L. Donoho and C. Grimes. Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data. *Proc. of the National Academy of Sciences*, 100(10):5591–5596, 2003.

D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.

P. Drineas and M. W. Mahoney. On the Nyström Method for approximating a Gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

S. Efromovich. Conditional density estimation in a regression setting. *The Annals of Statistics*, 35(6):2504–2535, 2007.

S. Efromovich. Dimension reduction and adaptation in conditional density estimation. *Journal of the American Statistical Association*, 105(490):761–774, 2010.

S. Efromovich. *Nonparametric Curve Estimation: Methods, Theory and Applications*. Springer Series in Statistics. Springer, 1999.

A. Estoup, E. Lombaert, J. Marin, et al. Estimation of demo-genetic model probabilities with approximate Bayesian computation using linear discriminant analysis on summary statistics. *Molecular Ecology Resources*, 12(5):846–855, 2012.

J. Fan. Local linear regression smoothers and their minimax efficiencies. *Annals of Statistics*, 21:196–216, 1993.

J. Fan, Q. Yao, and H. Tong. Estimation of conditional densities and sensitivity measures in nonlinear dynamical systems. *Biometrika*, 83(1):189–206, 1996.

J. Fan, L. Peng, Q. Yao, and W. Zhang. Approximating conditional density functions using dimension reduction. *Acta Mathematicae Applicatae Sinica*, 25(3):445–456, 2009.

J. Fan, Y. Feng, and R. Song. Nonparametric independence screening in sparse ultra-high-dimensional additive models. *Journal of the American Statistical Association*, 106(494):544–557, 2011.

Y. Fan, D. J Nott, and S. A Sisson. Approximate Bayesian computation via regression density estimation. *Stat*, 2:34–48, 2013.

O. P. Faugeras. A quantile-copula approach to conditional density estimation. *Journal of Multivariate Analysis*, 100(9):2083–2099, 2009.

A. Fernández-Soto, K. M. Lanzetta, and A. Yahil. A new catalog of photometric redshifts in the Hubble Deep Field. *The Astrophysical Journal*, 513:34–50, 1998.

A. Fernández-Soto, K. M. Lanzetta, H. W. Chen, B. Levine, and N. Yahata. Error analysis of the photometric redshift technique. *Monthly Notices of the Royal Astronomical Society*, 330:889–894, 2001.

D. Filipović, L. P Hughston, and A. Macrina. Conditional density models for asset pricing. *International Journal of Theoretical and Applied Finance*, 15(1), 2012.

P. E. Freeman, J. A. Newman, A. B. Lee, J. W. Richards, and C. M. Schafer. Photometric redshift estimation using Spectral Connectivity Analysis. *Monthly Notices of the Royal Astronomical Society*, 398(4):2012–2021, 2009.

P. E. Freeman, R. Izbicki, A. B. Lee, et al. New image statistics for detecting disturbed galaxy morphologies at high redshift. *Monthly Notices of the Royal Astronomical Society*, 434:282–295, 2013.

A. Gauci, K. Z. Adami, and J. Abela. Machine learning for galaxy morphology classification. *arXiv preprint arXiv:1005.0390*, 2010. URL http://arxiv.org/abs/1005.0390.

E. Giné and A. Guillou. Rates of strong uniform consistency for multivariate kernel density estimators. *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, 38:907–921, 2002.

E. Giné and V. Koltchinskii. Empirical graph Laplacian approximation of Laplace-Beltrami operators: Large sample results. In *High Dimensional Probability: Proceedings of the Fourth International Conference*, IMS Lecture Notes, pages 1–22, 2006.

I. K. Glad, N.L. Hjort, and G. Ushakov. Correction of density estimators that are not densities. *Scandinavian Journal of Statistics*, 30(2):415–427, 2003.

G. Gneiting and M. Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 171(1):125–51, 2014.

A. Gretton, K. Borgwardt, B. Schölkopf, M. Rasch, and A. Smola. A kernel method for the two sample problem. In *Advances in Neural Information Processing Systems 19*, 2007.

A. Gretton, A. Smola, J. Huang, et al. Covariate shift by kernel mean matching. In J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, editors, *Dataset Shift in Machine Learning*, chapter 8. Cambridge, MA, MIT Press, 2010.

N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

P. Hall and Q. Yao. Approximating conditional distribution functions using dimension reduction. *Annals of Statistics*, 33(3):1404–1421, 2005.

P. Hall, J. S. Racine, and Q. Li. Cross-validation and the estimation of conditional probability densities. *Journal of the American Statistical Association*, 99:1015–1026, 2004.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations*. New York: Springer-Verlag, 2001.

T. Hayfield and J. S. Racine. Nonparametric econometrics: The np package. *Journal of Statistical Software*, 27(5):1–32, 2008.

M. Hein, J. Y. Audibert, and U. von Luxburg. Intrinsic dimensionality estimation of submanifolds in R^{d}. In *Proc. of the 22nd International Conference on Machine learning*, pages 289–296, 2005.

H. Hendriks. Nonparametric estimation of a probability density on a riemannian manifold using fourier expansions. *The Annals of Statistics*, pages 832–849, 1990.

S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori. Statistical outlier detection using direct density ratio estimation. *Knowledge and Information Systems*, 26(2):309–336, 2011.

M. Hoffmann and O. Lepski. Random rates in anisotropic regression. *The Annals of statistics*, 30(2):325–358, 2002.

E. P. Hubble. Extra-galactic nebulae. *Astrophysical Journal*, 64:321–369, 1926.

R. J. Hyndman, D. M. Bashtannyk, and G. K. Grunwald. Estimating and visualizing conditional densities. *Journal of Computational & Graphical Statistics*, 5: 315–336, 1996.

T. Ichimura and D. Fukuda. A fast algorithm for computing least-squares cross-validations for nonparametric conditional kernel density functions. *Computational Statistics Data Analysis*, 54(12):3404–3410, 2010.

Ž. Ivezić, T. Axelrod, W.N. Brandt, et al. Large synoptic survey telescope: from science drivers to reference design. *Serbian Astronomical Journal*, 176(1):1–13, 2008.

R. Izbicki and A. B. Lee. Nonparametric conditional density estimation in a high-dimensional regression setting. 2014. Under Review.

R. Izbicki, A. B. Lee, and C. M. Schafer. High-dimensional density ratio estimation with extensions to approximate likelihood computation. In *Proceedings of the*

Seventeenth International Conference on Artificial Intelligence and Statistics, pages 420–429, 2014.

M. Ji, T. Yang, B. Lin, R. Jin, and J. Han. A simple algorithm for semi-supervised learning with improved generalization error bound. In *Proc. of the 29th International Conference on Machine learning*, 2012.

I. T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

A. Kalda and S. Siddiqui. Nonparametric conditional density estimation of short-term interest rate movements: procedures, results and risk management implications. *Applied Financial Economics*, 23(8):671–684, 2013.

W. C. M. Kallenberg and T. Ledwina. Data-driven smooth tests when the hypothesis is composite. *Journal of the American Statistical Association*, 92:1094–1104, 1997.

T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10:1391–1445, 2009.

T. Kanamori, T. Suzuki, and M. Sugiyama. Statistical analysis of kernel-based least-squares density-ratio estimation. *Machine Learning*, 86(3):335–367, 2012.

M. C. Kind and R. J. Brunner. Tpz: photometric redshift pdfs and ancillary information by using prediction trees and random forests. *Monthly Notices of the Royal Astronomical Society*, 432(2):1483–1501, 2013.

V. Koltchinskii and M. Yuan. Sparsity in multiple kernel learning. *The Annals of Statistics*, 38(6):3660–3695, 2010.

S. Kpotufe. k-NN regression adapts to local intrinsic dimension. In *Advances in Neural Information Processing Systems 24*, pages 729–737. The MIT Press, 2011.

S. Kpotufe. *The curse of dimension in nonparametric regression*. PhD thesis, University of California, 2010.

J. Lafferty and L. Wasserman. Rodeo: sparse, greedy nonparametric regression. *The Annals of Statistics*, 36(1):28–63, 2008.

A. B. Lee and L. Wasserman. Spectral Connectivity Analysis. *Journal of the American Statistical Association*, 105(491):1241–1255, 2010.

A. B. Lee, D. Luca, and K. Roeder. A spectral graph approach to discovering genetic ancestry. *The Annals of Applied Statistics*, 4(1):179, 2010.

M. Lima, C.E. Cunha, H. Oyaizu, et al. Estimating the redshift distribution of photometric galaxy samples. *Monthly Notices of the Royal Astronomical Society*, (390):118–130, 2008.

H. Liu, J. Lafferty, and L. Wasserman. Sparse nonparametric density estimation in high dimensions using the rodeo. In *Eleventh International Conference on Artificial Intelligence and Statistics*, 2007.

M. Loog. Nearest neighbor-based importance weighting. In *IEEE International workshop on machine learning for signal processing*, 2012.

J. M. Lotz, J. Primack, and P. Madau. A new nonparametric approach to galaxy morphological classification. *The Astronomical Journal*, 128(1):163, 2004.

S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 3rd edition, 2009.

R. Mandelbaum, U. Seljak, C. M. Hirata, S. Bardelli, and M. Bardelli. Precision photometric redshift calibration for galaxy-galaxy weak lensing. *Monthly Notices of the Royal Astronomical Society*, (386):781–806, 2008.

A. Margolis. A literature review of domain adaptation with unlabeled data, March 2011. URL http://ssli.ee.washington.edu/~amargoli/review_Mar23.pdf.

J. M. Marin, P. Pudlo, C. P Robert, and R. J Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012.

P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré. Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328, 2003.

M. Meila and J. Shi. A random walks view on spectral segmentation. In *Proc. Eighth International Conference on Artificial Intelligence and Statistics*, 2001.

H. Q. Minh. Some properties of Gaussian Reproducing Kernel Hilbert Spaces and their implications for function approximation and learning theory. *Constructive Approximation*, 32(2):307–338, 2010.

H. Q. Minh, P. Niyogi, and Y. Yao. Mercer's theorem, feature maps, and smoothing. In *19th Annual Conference on Learning Theory*, 2006.

J. G. Moreno-Torres, T. Raeder, R. Alaíz-Rodríguez, N. V. Chawla, and F. Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012.

B. Nadler, A. Srebro, and X. Zhou. Semi-supervised learning with the graph Laplacian: The limit of infinite unlabelled data. In *Advances in Neural Information Processing Systems 23*, 2009.

H. H. Nam, H. Hachiya, and M. Sugiyama. Computationally efficient multi-label classification by least-squares probabilistic classifier. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 2077–2080. IEEE, 2012.

J. Neyman. » smooth test» for goodness of fit. *Scandinavian Actuarial Journal*, 1937 (3-4):149–199, 1937.

F. O'Sullivan. A statistical perspective on ill-posed inverse problems. *Statistical science*, 1(4):502–518, 1986.

H. Oyaizu, M. Lima, C.E. Cunha, H. Lin, and J. Frieman. Photometric redshift error estimators. *The Astrophysical Journal*, 689:709–720, 2008.

B. Pelletier. Kernel density estimation on riemannian manifolds. *Statistics & probability letters*, 73(3):297–304, 2005.

B. Pelletier. Non-parametric regression estimation on closed riemannian manifolds. *Journal of Nonparametric Statistics*, 18(1):57–67, 2006.

C. Y. Peng, L. C. Ho, C. D. Impey, and H. W. Rix. Detailed structural decomposition of galaxy images. *The Astronomical Journal*, 124(1):266, 2002.

J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman. Sparse additive models. *Journal of the Royal Statistical Society, Series B*, 71(5):1009–1030, 2009.

J. W. Richards, P. E. Freeman, A. B. Lee, and C. M. Schafer. Exploiting low-dimensional structure in astronomical spectra. *Astrophysical Journal*, 691:32–42, 2009.

L. Rosasco, M. Belkin, and E. D. Vito. A note on perturbation results for learning empirical operators. CSAIL Technical Report TR-2008-052, CBCL-274, Massachusetts Institute of Technology, 2008.

L. Rosasco, M. Belkin, and E. D. Vito. On learning with integral operators. *Journal of Machine Learning Research*, 11:905–934, 2010.

M. Rosenblatt. Conditional probability density and regression estimators. In P.R. Krishnaiah, editor, *Multivariate Analysis II*. 1969.

D. B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

M. Saerens, F. Fouss, L. Yen, and P. Dupont. The principal components analysis of a graph, and its relationships to spectral clustering. In *Proceedings of the 15th European Conference on Machine Learning (ECML 2004). Lecture Notes in Artificial Intelligence*, pages 371–383. Springer-Verlag, 2004.

Y. Safarov and D. Vassilev. *The asymptotic distribution of eigenvalues of partial differential operators*, volume 155. American Mathematical Society, 1997.

B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Cambridge, MA, MIT Press, 2001.

B. Schölkopf, A. J. Smola, and K. R. Müller. Cambridge, MA, MIT Press.

B. Schölkopf, A. Smola, and K. R. Müller. Kernel principal component analysis. In *Artificial Neural Networks - ICANN'97*, pages 583–588. Springer, 1997.

E. S. Sheldon, C. E. Cunha, R. Mandelbaum, J. Brinkmann, and B. A. Weaver. Photometric redshift probability distributions for galaxies in the SDSS DR8. *The Astrophysical Journal Supplement Series*, 201(2):32, 2012.

T. Shi, M. Belkin, and B. Yu. Data spectroscopy: eigenspace of convolution operators and clustering. *The Annals of Statistics*, 37, 6B:3960–3984, 2009.

H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

A. Singer. From graph to manifold Laplacian: The convergence rate. *Applied and Computational Harmonic Analysis*, 21:128–134, 2006.

K. Sinha and M. Belkin. Semi-supervised learning using sparse eigenfunction bases. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1687–1695. 2009.

S. A. Sisson, Y. Fan, and M. M. Tanaka. Sequential monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765, 2007.

V. Springel, C. S. Frenk, and S. D.M. White. The large-scale structure of the universe. *Nature*, 440:1137–1144, 2006.

I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, 2008.

I. Steinwart, D. R Hush, and C. Scovel. Optimal rates for regularized least squares regression. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

C. J. Stone. Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, 10(4):1040–1053, 1982.

M. Sugiyama, T. Suzuki, S. Nakajima, et al. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4): 699–746, 2008.

M. Sugiyama, T. Suzuki, and T. Kanamori. Density ratio estimation: A comprehensive review. *RIMS Kokyuroku*, pages 10–31, 2010a.

M. Sugiyama, I. Takeuchi, T. Suzuki, et al. Conditional density estimation via least-squares density ratio estimation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 781–788, 2010b.

M. Sugiyama, M. Yamada, P. von Bünau, et al. Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search. *Neural Networks*, 24:183–198, 2011.

J. Sun and C. R. Loader. Simultaneous confidence bands for linear regression and smoothing. *The Annals of Statistics*, pages 1328–1345, 1994.

I. Takeuchi, Q. V. Le, T. D. Sears, A. J. Smola, and C. Williams. Nonparametric quantile estimation. *Journal of Machine Learning Research*, 7:7–1231, 2006.

I. Takeuchi, K. Nomura, and T. Kanamori. Nonparametric conditional density estimation using piecewise-linear solution path of kernel quantile regression. *Neural Computation*, 21(2):533–559, 2009.

J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

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

G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

L. Wasserman. *All of Nonparametric Statistics*. Springer-Verlag New York, Inc., 2006.

B. J. Weiner, A. C. Phillips, S.M. Faber, et al. The DEEP groth strip galaxy redshift survey. III. Redshift catalog and properties of galaxies. *The Astrophysical Journal*, 620(2):595, 2005.

A. Weyant, C. Schafer, and W. M. Wood-Vasey. Likelihood-free cosmological inference with type ia supernovae: Approximate Bayesian computation for a complete treatment of uncertainty. *The Astrophysical Journal*, 764(2):116, 2013.

R. A. Windhorst, S. H. Cohen, N. P. Hathi, et al. The hst wfc3 early release science data: Panchromatic faint object counts for 0.2-2 μm wavelength. *The Astrophysical Journal Supplement Series*, 193(2), 2011.

D. Wittman. What lies beneath: Using p(z) to reduce systematic photometric redshift errors. *The Astrophysical Journal Letters*, 700(2), 2009.

S. Wold, M. Sjöström, and L. Eriksson. PLS-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, 58:109–130, 2001.

G. B. Ye and D. X. Zhou. Learning and approximation by gaussians on riemannian manifolds. *Advances in Computational Mathematics*, 29(3):291–310, 2008.

D. G. York, J. Adelman, J. E. Anderson Jr, et al. The Sloan Digital Sky Survey: Technical summary. *The Astronomical Journal*, 120(3):1579, 2000.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–67, 2006.

B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the 21st international conference on Machine learning*, page 114, 2004.

L. Zhao and Z. Liu. Strong consistency of the kernel estimators of conditional density function. *Acta Mathematica Sinica*, 1(4):314–318, 1985.

H. Zheng and Y. Zhang. Review of techniques for photometric redshift estimation. In *Software and Cyberinfrastructure for Astronomy II*, volume 8451, 2012.

X. Zhou and N. Srebro. Error analysis of Laplacian eigenmaps for semi-supervised learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 901–908, 2011.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, pages 912–919, 2003.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67(2):301–320, 2005.

L. Zwald and G. Blanchard. On the convergence of eigenspaces in kernel principal component analysis. In *Advances in Neural Information Processing Systems 18*. Cambridge, MA, MIT Press, 2005.