7-2-2011

# Adaptive weighting for flexible estimation in nonparametric regression models.

Alvaro Nosedal-Sanchez

Alvaro Nosedal-Sanchez
*Candidate*

Mathematics and Statistics
*Department*

This dissertation is approved, and it is acceptable in quality and form for publication:

*Approved by the Dissertation Committee:*

, Chairperson

EJ Bedrick

# Adaptive Weighting for Flexible Estimation in Nonparametric Regression Models.

by

**Alvaro Nosedal-Sánchez**

B.S., Universidad Nacional Autónoma de México, 2000
M.S., Statistics, University of New Mexico, 2007

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Statistics

The University of New Mexico

Albuquerque, New Mexico

May, 2011

# Dedication

*To Erika, the Nosedals, the Sanchezs and the Hernandezs.*

*"I have found inspiration: You have willed me to succeed, sometimes even in my lowest moments. And I have found generosity: You have given me your shoulders to stand on, to reach for my dreams, dreams I could have never reached without you. I have found you, and I will take you and the memory of you with me for the rest of my life."*

*–Andre Agassi*

# Acknowledgments

Foremost, I would like to thank my adviser and friend, Dr. Curt Storlie, for his many suggestions, constant support during this research and for introducing me to rock climbing. Doing research and climbing with him showed me that it takes a lot of work to make it to the top. I look forward to years of future collaboration. I am also thankful to Dr. Ron Christensen for a great course in Linear Models, for being my tennis partner and for his friendship. My statistical thinking, my serve and my return will always be influenced by him. I would also like to thank Dr. Gabriel Huerta for his friendship, for the opportunity of working under his supervision at the Statistics Clinic and, especially, for his encouragement to apply to UNM. I would also like to extend my gratitude to Dr. Vageli Coutsias, who expressed his interest in my work and lend me a book in Functional Analysis that gave me a better perspective about the background needed for my research. Thanks are also due to Dr. Ed Bedrick, who always gave me good advise and friendly encouragement. I am also thankful to Dr. Michele Guindani, for his friendship and because thanks to him I started working with Curt, Grazie!

I should also mention that my graduate studies in UNM were supported in part by the National Research and Technology Council of Mexico (CONACYT).

Of course, I am grateful to my parents, Dolores Sánchez Góngora and Víctor Nosedal Blancas, for their patience and *love*. Without them this work would never have come into existence (literally).

Finally, I wish to thank the following: Erika, my beautiful wife (for her love); Vicky and Ed (for a fun introduction to the art of doubles); Erik and Dave (for being such a great study group), Zibo, Zazil, Siu Kei, Flor, Pedro, Jaime, Shadi, Gaby, Jean, Eva, Qiao, Danielle, Glenn, Angela, Fletcher, Pablo and Ana (for all the good times we had together); and, last but not least, my brothers, Benjamin, Jenaro and Alex Villagrán, (for being always there for me).

Albuquerque, New Mexico		Al Nosedal-Sánchez
Fall, 2010

# Adaptive Weighting for Flexible Estimation in Nonparametric Regression Models.

by

**Alvaro Nosedal-Sánchez**

ABSTRACT OF DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Statistics

The University of New Mexico

Albuquerque, New Mexico

May, 2011

# Adaptive Weighting for Flexible Estimation in Nonparametric Regression Models.

by

**Alvaro Nosedal-Sánchez**

B.S., Universidad Nacional Autónoma de México, 2000
M.S., Statistics, University of New Mexico, 2007

PhD, Statistics, University of New Mexico, 2011

## Abstract

Nonparametric Regression has proven to be a very useful methodology, with applications to a large list of modern problems such as computer models, image data, environmental processes, to name a few. The nonparametric regression model is given by

$$y_i = f_0(x_i) + \epsilon_i, \; i = 1, 2, ..., n \tag{0.1}$$

where $f_0$ is an unknown regression function and $\epsilon_i$ are independent error terms.

Smoothing splines are among the most popular methods for estimation of $f_0$ due to their good empirical performance and sound theoretical support [4, 26, 6, 30]. It is usually assumed without loss of generality that the domain of $f_0$ is [0,1]. Let

$f^{(m)}$ denote the $m^{th}$ derivative of $f$. The smoothing spline estimate $\hat{f}$ is the unique minimizer of

$$\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int_0^1 \left(f^{(m)}(x)\right)^2 dx \tag{0.2}$$

over all functions, $f$, in $m^{th}$ order Sobolev space,

$$S^m = \{f : f^{(j)} \text{ is absolutely continuous for } j = 1, ..., m-1 \text{ and } f^{(m)} \in L_2\}$$

The minimizer of (0.2) trades off fidelity to the data (in terms of residual sum of squares) against smoothness of the reconstructed curve (in terms of the integrated squared derivative of order $m$, where $m$ is typically taken to be two). The smoothing spline solution uses a global smoothing parameter $\lambda$ which implies that the true underlying mean process has a constant degree of smoothness. We suggest a more general, "spatially adaptive", framework that accommodates varying degrees of roughness by seeking solutions where the smoothness penalty depends on the region of the domain being fitted. We derive the solution within a Reproducing Kernel Hilbert Space framework [31, 10].

We propose a method which breaks down the interval $[0, 1]$ into $p$ disjoint sub-intervals. Then we define $p$ functional components in $[0, 1]$, which have two important features. First, the purpose of each of these $p$ components is to estimate the true function locally, i.e., in only one of the sub-intervals. Second, even though all components are defined on the entire domain, i.e. $[0, 1]$, a component has curvature only in one of the afore mentioned intervals. The $p$ local estimates are then added together to produce a function estimate over the entire $[0, 1]$ interval. This is similar in spirit to the method of [22]. However, in the proposed method, the additional flexibility that comes from finding these $p$ local functional estimates does not come at any additional computational cost. In spite of having $p$ components there is no need to specify (e.g., choose via cross validation) $p$ smoothing parameters. Theory from

COmponent Selection and Shrinkage Operator (COSSO) [15], reduces the problem of specifying these $p$ smoothing parameters to specifying only one smoothing parameter without a loss in flexibility. In fact, empirical studies indicate superior performance of COSSO in the additive model framework over that for the traditional additive model [12], see [29], for example.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The overarching goal of this dissertation is to contribute to the literature about the use of reproducing kernel Hilbert spaces (RKHS) for penalized regression and present an innovative approach to Nonparametric Regression. The work presented here focuses on two areas: 1) Making the theory of RKHS more accessible to statisticians; and 2) Development of a new method in locally adaptive functional estimation that allows for a solution to have a degree of roughness (or smoothness) that depends on the region of the domain being fitted. The paragraphs below, summarize the two stand-alone dissertation chapters.

Penalized Regression procedures have become very popular ways to estimate complex functions (e.g., [31],[13],[6]). These procedures use an estimator that is the solution to a minimization problem. In any minimization problem, there are the following questions: Does the solution exist? Is it unique? How can we find it? If the problem is posed in the reproducing kernel Hilbert space (RKHS) framework that we discuss on Chapter 2, then the solution is guaranteed to exist, it is unique and it takes a particularly simple form. Reproducing kernel Hilbert spaces and reproducing kernels play a central role in penalized regression. In the first section of Chapter 2 we present and solve simple problems while gently introducing key concepts. Section 2 takes the reader from a basic understanding of fields through Banach Spaces and Hilbert Spaces. In Section 3, we provide elementary theory for RKHS along with some examples. Section 4 discusses Penalized Regression with RKHS. Two specific examples involving ridge regression and the cubic smoothing spline are given with R codes to solidify the concepts. Chapter 2 will be submitted to *Statistical Science* and is currently under review.

Nonparametric Regression is a very useful approach to a large list of modern problems such as computer models, image data, environmental processes, to name a few. The nonparametric regression model is given by

$$y_i = f_0(x_i) + \epsilon_i, \ i = 1, 2, ..., n \tag{1.1}$$

where $f_0$ is an unknown regression function and $\epsilon_i$ are independent error terms.

Smoothing splines are among the most popular methods for estimation of $f_0$ due to their good empirical performance and sound theoretical support ([4], [26], [6], [30], and many others). It is usually assumed without loss of generality that the domain of $f_0$ is [0,1]. Let $f^{(m)}$ denote the $m^{th}$ derivative of $f$. The smoothing spline estimate $\hat{f}$ is the unique minimizer of

$$\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int_0^1 \left(f^{(m)}(x)\right)^2 dx \tag{1.2}$$

over all functions, $f$, in $m^{th}$ order Sobolev space,

$$S^m = \{f : f^{(j)} \text{ is absolutely continuous for } j = 1, ..., m-1 \text{ and } f^{(m)} \in L_2\}$$

The minimizer of (1.2) trades off fidelity to the data in terms of residual sum of squares against smoothness of the reconstructed curve in terms of the integrated squared derivative of order $m$, where $m$ is typically taken to be two. The smoothing spline solution uses a global smoothing parameter $\lambda$ which implies that the true underlying mean process has a constant degree of smoothness. In Chapter 3 we suggest a more general, ”spatially adaptive”, framework that accommodates varying degrees of roughness by seeking solutions where the smoothness penalty depends on the region of the domain being fitted. We derive the solution within a Reproducing Kernel Hilbert Space. When applying the method to data we propose to breakdown the interval $[0, 1]$ in $p$ intervals. Then we define $p$ functional components in $[0, 1]$, which have two important features. First, the purpose of each of these $p$ components is to estimate the true function locally, that is, in only one of the intervals. Second, even though all components are defined on the entire domain, i.e. $[0, 1]$, a component has curvature only in one of the afore mentioned intervals. The $p$ local estimates are then added together to produce function estimate over the entire $[0, 1]$ interval. The additional flexibility that comes from finding these $p$ local functional estimates does not come at an additional computational cost. In spite of having $p$ components there is no need of finding $p$ smoothing parameters. Theory from COmponent Selection and Shrinkage Operator (COSSO), see [15], reduces the problem of finding these $p$ smoothing parameters to finding only one smoothing parameter without a loss in model's flexibility. In fact, empirical studies indicate superior performance of COSSO than that for the traditional additive model see [28], for example.

Chapter 4 presents a general summary and provides a detailed outline of several possible areas for future research and development.

# Chapter 2

# Reproducing Kernel Hilbert Spaces (RKHS)

## 2.1 Introduction

Penalized regression procedures have become a popular approach to estimating complex functions [31, 6, 13]. These procedures use an estimator that is defined as the solution to a minimization problem. In any minimization problem, there are the following questions: Does the solution exist? If yes, is the solution unique? And how can we find it? If the problem is posed in reproducing kernel Hilbert spaces (RKHS), then the solution is guaranteed to exist, it is unique and it takes a particularly simple form.

Reproducing kernel Hilbert spaces and reproducing kernels play a central role in penalized regression. In the first section of Chapter 2 we present and solve simple problems while gently introducing key concepts. Section 2 takes the reader from a basic understanding of fields through Banach Spaces and Hilbert Spaces. In Section 3, we provide elementary theory for RKHS along with some examples. Section 4 discusses Penalized Regression with RKHS. Two specific examples involving ridge regression and the cubic smoothing spline are given with R codes to solidify the concepts. Some methods for smoothing parameter selection are briefly mentioned at the end. Section 5 contains some closing remarks.

### 2.1.1 Why Reproducing Kernel Hilbert Spaces?

Before introducing new concepts, we present some simple problems that illustrate the need and utility of the RKHS framework. Consider solving the system of linear equations

$$x_1 + x_3 = 0 \tag{2.1}$$

$$x_2 = 1. \tag{2.2}$$

Clearly the real-valued solutions to this system are the vectors $\boldsymbol{x}_*^t = (-\alpha, 1, \alpha)$ for $\alpha \in \Re$. Suppose we want to find the "smallest" solution. Under the usual squared norm $\|\boldsymbol{x}\|^2 = x_1^2 + x_2^2 + x_3^2$, the smallest solution is $\boldsymbol{x}_s^t = (0, 1, 0)$.

Consider a more general problem. For a given $p \times n$ matrix $\boldsymbol{R}$ and $n \times 1$ matrix $\boldsymbol{\eta}$, solve

$$\boldsymbol{R}^t\boldsymbol{x} = \boldsymbol{\eta}, \tag{2.3}$$

where $\boldsymbol{R}^t$ is the transpose of $\boldsymbol{R}$, $\boldsymbol{x}$ and the columns of $\boldsymbol{R}$, say $\boldsymbol{R}_k$, $k = 1, 2, ..., n$, are all in $\Re^p$, and $\boldsymbol{\eta} \in \Re^n$. We wish to find the solution $\boldsymbol{x}_s$ that minimizes the norm $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^t\boldsymbol{x}}$. We solve the problem using concepts that extend to RKHS.

A solution $\boldsymbol{x}_*$ (not necessarily a minimum norm solution) exists whenever $\boldsymbol{\eta} \in C(\boldsymbol{R}^t)$. Here $C(\boldsymbol{R}^t)$ denotes the column space of $\boldsymbol{R}^t$. Given one solution $\boldsymbol{x}_*$, all solutions $\boldsymbol{x}$ must satisfy

$$\boldsymbol{R}^t\boldsymbol{x} = \boldsymbol{R}^t\boldsymbol{x}_*$$

or

$$\boldsymbol{R}^t(\boldsymbol{x}_* - \boldsymbol{x}) = \boldsymbol{0}.$$

The vector $\boldsymbol{x}_*$ can be written uniquely as $\boldsymbol{x}_* = \boldsymbol{x}_0 + \boldsymbol{x}_1$ with $\boldsymbol{x}_0 \in C(\boldsymbol{R})$ and $\boldsymbol{x}_1 \in C(\boldsymbol{R})^\perp$, where $C(\boldsymbol{R})^\perp$ is the orthogonal complement of $C(\boldsymbol{R})$. Clearly, $\boldsymbol{x}_0$ is a solution because $\boldsymbol{R}^t(\boldsymbol{x}_* - \boldsymbol{x}_0) = \boldsymbol{R}^t\boldsymbol{x}_1 = \boldsymbol{0}$

In fact, $\boldsymbol{x}_0$ is both the unique solution in $C(\boldsymbol{R})$ and the minimum norm solution. If $\boldsymbol{x}$ is any other solution in $C(\boldsymbol{R})$ then $\boldsymbol{R}^t(\boldsymbol{x} - \boldsymbol{x}_0) = \boldsymbol{0}$ so we have both $(\boldsymbol{x} - \boldsymbol{x}_0) \in C(\boldsymbol{R})^\perp$ and $(\boldsymbol{x} - \boldsymbol{x}_0) \in C(\boldsymbol{R})$, two sets whose intersection is only the $\boldsymbol{0}$ vector. Thus $\boldsymbol{x} - \boldsymbol{x}_0 = \boldsymbol{0}$ and $\boldsymbol{x} = \boldsymbol{x}_0$. In other words, every solution $\boldsymbol{x}_*$ has the same $\boldsymbol{x}_0$ vector. Finally, $\boldsymbol{x}_0$ is also the minimum norm solution because the arbitrary solution $\boldsymbol{x}_*$ has

$$\boldsymbol{x}_0^t\boldsymbol{x}_0 \leq \boldsymbol{x}_0^t\boldsymbol{x}_0 + \boldsymbol{x}_1^t\boldsymbol{x}_1 \leq \boldsymbol{x}_*^t\boldsymbol{x}_*$$

We have established the existence of a unique, minimum norm solution in $C(\boldsymbol{R})$ that can be written as

$$\boldsymbol{x}_s \equiv \boldsymbol{x}_0 = \boldsymbol{R}\xi = \sum_{k=1}^{n} \xi_k \boldsymbol{R}_k, \tag{2.4}$$

for some $\xi_k$, $k = 1, \ldots, n$. To find $\boldsymbol{x}_s$ explicitly, write $\boldsymbol{x}_s = \boldsymbol{R}\xi$ and the defining equation (2.3) becomes

$$\boldsymbol{R}^t\boldsymbol{R}\boldsymbol{\xi} = \boldsymbol{\eta}, \tag{2.5}$$

which is just a system of linear equations. Even if there exist multiple solutions $\boldsymbol{\xi}$, $\boldsymbol{R}\boldsymbol{\xi}$ is unique.

Now we use this framework to find the “smallest” solution to the system of equations (1) and (2). In the general framework we have

$$\begin{aligned}
\boldsymbol{x}^t &= (x_1, x_2, x_3), \\
\boldsymbol{\eta}^t &= (0, 1), \\
\boldsymbol{R}_1^t &= (1, 0, 1), \\
\boldsymbol{R}_2^t &= (0, 1, 0).
\end{aligned}$$

We know that the solution has the form (2.4) and we also know that we have to solve a system of equations given by (2.5). In this case, the system of equations is

$$\begin{aligned}
2\xi_1 + 0\xi_2 &= 0, \\
0\xi_1 + 1\xi_2 &= 1.
\end{aligned}$$

The solution to the above system is $(\xi_1, \xi_2) = (0, 1)$ which implies our solution to the original problem is $\boldsymbol{x}_s^t = 0\boldsymbol{R}_1 + 1\boldsymbol{R}_2 = (0, 1, 0)$ as expected.

Virtually the same methods can be used to solve a similar problem in any inner-product space $\Omega$. As discussed later, an inner product $\langle \cdot, \cdot \rangle$ assigns real numbers to pairs of "vectors." For given vectors $\boldsymbol{R}_k \in \Omega$ and numbers $\eta_k \in \Re$, find $\boldsymbol{x} \in \Omega$ such that

$$\langle \boldsymbol{R}_k, \boldsymbol{x} \rangle = \eta_k,\ k = 1, 2, ..., n \tag{2.6}$$

for which the norm of $\|\boldsymbol{x}\| \equiv \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$ is minimal. The solution has the form

$$\boldsymbol{x}_s = \sum_{k=1}^{n} \xi_k \boldsymbol{R}_k, \tag{2.7}$$

with $\xi_k$ satisfying the linear equations

$$\sum_{k=1}^{n} \langle \boldsymbol{R}_i, \boldsymbol{R}_k \rangle \xi_i = \eta_i, \qquad i = 1, \ldots, n.$$

For a formal proof of this result, see [17]. In RKHS vectors are taken to be functions.

**Interpolating Spline Problem.** We now apply these ideas to finding a function $f(t)$ that interpolates between the pairs of numbers $(t_k, \eta_k)$, $k = 0, 1, 2, ..., n$ where $(t_0, \eta_0) \equiv (0, 0)$. We restrict our attention to functions $f \in F$ where $F$={$f$: $f$ is absolutely continuous on [0,1], $f(0) = 0$, $f' \in L^2[0, 1]$ }. Throughout $f^{(m)}$ denotes the $m$-th derivative of $f$ with $f' \equiv f^{(1)}$ and $f'' \equiv f^{(2)}$. The restriction that $f(0) = 0$ is not really necessary, but simplifies the presentation.

In particular, we want to find the smoothest function $f(t)$ that satisfies $f(t_k) = \eta_k$, $k = 1, \ldots, n$. Defining the inner product

$$\langle f, g\rangle = \int_0^1 f'(x)g'(x)dx$$

implies a norm over the space $F$ that is small for "smooth" functions. To address the interpolation problem, note that the functions $R_k(s) \equiv \min(s, t_k)$, $k = 1, 2, ..., n$ have $R_k(0) = 0$ and the property $\langle R_k, f\rangle = f(t_k)$ which we verify as follows

$$\begin{aligned} \langle f, R_k\rangle &= \int_0^1 f'(s)R_k'(s)ds \\ &= \int_0^{t_k} f'(s)1ds + \int_{t_k}^1 f'(s)0ds \\ &= \int_0^{t_k} f'(s)ds = f(t_k) - f(0) = f(t_k). \end{aligned}$$

Thus, an interpolator $f$ satisfies a system of equations like (6), $f(t_k) = \langle R_k, f\rangle = \eta_k$, $k = 1, \ldots, n$, and by (2.7), the smoothest function $f$ (minimum norm) that satisfies the requirements has the form

$$\hat{f}(t) = \sum_{k=1}^{n} \xi_k R_k(t)$$

where the $\xi_j s$ are the solutions to the system of real linear equations

$$\hat{f}(t_k) = \langle R_k, \hat{f}\rangle = \sum_{j=1}^{n} \langle R_k, R_j\rangle \xi_j = \eta_k, \qquad k = 1, 2, ..., n.$$

Note that

$$\langle R_k, R_j \rangle = R_j(t_k) = R_k(t_j) = \min(t_k, t_j)$$

and define the function

$$R(s,t) = \min(s,t)$$

which turns out to be a reproducing kernel.

**Numerical example.** Given points $f(t_i) = \eta_i$, say, $f(0) = 0$, $f(0.1) = 0.1$, $f(0.25) = 1$, $f(0.5) = 2$, $f(0.75) = 1.5$, and $f(1) = 1.75$, we find

$$\text{arg min}_{f \in F} \|f\|^2 = \int_0^1 f'(x)^2 dx.$$

Recall from (2.7) that $\hat{f}(t) = \sum_{k=1}^n \xi_k R_k(t)$, where the $\xi's$ are the solution to $\sum_{k=1}^n \xi_k R_k(t_i) = \eta_i$ with $R_k(t_i) = \min(t_i, t_k)$. The resulting system of equations is

$$\begin{aligned}
0.1\xi_1 + 0.1\xi_2 + 0.1\xi_3 + 0.1\xi_4 + 0.1\xi_5 &= 0.1 \\
0.1\xi_1 + 0.25\xi_2 + 0.25\xi_3 + 0.25\xi_4 + 0.25\xi_5 &= 1 \\
0.1\xi_1 + 0.25\xi_2 + 0.5\xi_3 + 0.5\xi_4 + 0.5\xi_5 &= 2 \\
0.1\xi_1 + 0.25\xi_2 + 0.5\xi_3 + 0.75\xi_4 + 0.75\xi_5 &= 1.5 \\
0.1\xi_1 + 0.25\xi_2 + 0.5\xi_3 + 0.75\xi_4 + \xi_5 &= 1.75.
\end{aligned}$$

The solution to this system of equations is $\xi^t = (-5, 2, 6, -3, 1)$, which implies that our function is

$$\begin{aligned}
\hat{f}(t) &= -5R_1(t) + 2R_2(t) + 6R_3(t) - 3R_4(t) + 1R_5(t) \qquad (2.8) \\
&= -5R(t,t_1) + 2R(t,t_2) + 6R(t,t_3) - 3R(t,t_4) + 1R(t,t_5)
\end{aligned}$$

or, adding the slopes for $t > t_i$ and finding the intercepts,

$$\hat{f}(t) = \begin{cases} t & 0 \le t \le 0.1 \\ 6t - 0.5 & 0.1 \le t \le 0.25 \\ 4t & 0.25 \le t \le 0.5 \\ -2t + 3 & 0.5 \le t \le 0.75 \\ t + 0.75 & 0.75 \le t \le 1 \end{cases}$$

The solution is the linear interpolating spline as can be seen graphically in Figure 1. For this illustrative example we restricted $f$ so that $f(0) = 0$. This was only for convenience of presentation. It can be shown that the form of the solution remains the same with any shift to the function, so that in general the solution takes the form $\hat{f}(t) = \xi_0 + \sum_{k=1}^{n} \xi_k R_k(t)$.

The key points are (i) the elements $R_i$ that allow us to express a function evaluated at a point as an inner-product constraint, and (ii) the restriction to functions in $F$. $F$ is a very special function space, a reproducing kernel Hilbert space, and $R_i$ is determined by a reproducing kernel $R$.

Ultimately, our goal is to address more complicated regression problems like the

**Linear Smoothing Spline Problem.** Consider simple regression data $(x_i, y_i)$, $i = 1, \ldots, n$ and finding the function that minimizes

$$\frac{1}{n} \sum_{i=1}^{n} \{y_i - f(x_i)\}^2 + \lambda \int_0^1 f'(x)^2 dx. \tag{2.9}$$

If $f(x)$ is restricted to be in some class of functions $F$, minimizing only the first term gives least squares estimation within $F$. If $F$ contains functions with $f(x_i) = y_i$ for all $i$, such functions minimize the first term but are typically very “unsmooth,” i.e., has large second term. The second “penalty” term is minimized by having a horizontal line, but that rarely has a small first term. As we will see in Section 2.4, for suitable $F$ the minimizer takes the form

$$\hat{f}(x) = \xi_0 + \sum_{i=k}^{n} \xi_k R_k(x),$$

where the $R_k$'s are known functions and the $\xi_k$'s are coefficients found by solving a system of linear equations. This produces a linear smoothing spline.

If our goal is only to derive the solution to the linear smoothing spline problem with one predictor variable, RKHS theory is an overkill. The value of RKHS theory lies in its generality. The linear spline penalty can be replaced by any other penalty with an associated inner product, and the $x_i$'s can be vectors in $\Re^p$. Using RKHS results, we can solve the general problem of finding the minimizer of $\frac{1}{n}\sum_{i=1}^{n}\left(y_i - f(x_i)\right)^2 + \lambda J(f)$ for a general functional $J$ that corresponds to a squared norm in a subspace. See [31] or [10] for a full treatment of this approach. We now present an introduction to this theory.

## 2.2 Vector, Banach and Hilbert Spaces

This section summarizes background material required for the formal development of the RKHS framework.

### 2.2.1 Vector Spaces

A vector space is a set that contains elements called "vectors" and supports two kinds of operations: addition of vectors and multiplication by scalars. The scalars are drawn from some field (which are the real numbers in the rest of this article) and the vector space is said to be a vector space over that field. Formally, a set $V$ is a vector space over a field $F$ if there exists a structure of the form $\langle V, F, +, \times, 0_v \rangle$

consisting of $V$, $F$, a vector addition operation $+$, a scalar multiplication $\times$, and an identity element $0_v \in V$. This structure must obey the following axioms for any $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in V$ and $a, b \in F$:

- Associative Law: $(\boldsymbol{u}+\boldsymbol{v})+\boldsymbol{w} = \boldsymbol{u}+(\boldsymbol{v}+\boldsymbol{w})$.
- Commutative Law: $\boldsymbol{u}+\boldsymbol{v} = \boldsymbol{v}+\boldsymbol{u}$.
- Inverse Law: $\exists \boldsymbol{s} \in V$ *s.t.* $\boldsymbol{u}+\boldsymbol{s} = 0_v$. (Write $-\boldsymbol{u} \equiv \boldsymbol{s}$.)
- Identity Laws: $0_v + \boldsymbol{u} = \boldsymbol{u}$.
- $1 \times \boldsymbol{u} = \boldsymbol{u}$.
- Distributive Laws: $a \times (b \times \boldsymbol{u}) = (a \times b) \times \boldsymbol{u}$.
- $(a+b) \times \boldsymbol{u} = a \times \boldsymbol{u} + b \times \boldsymbol{u}$.
- $a \times (\boldsymbol{u}+\boldsymbol{v}) = a \times \boldsymbol{u} + a \times \boldsymbol{v}$.

We will write $\mathbf{0}$ for $0_v \in V$ and $\boldsymbol{u}+(-\boldsymbol{v})$ as $\boldsymbol{u}-\boldsymbol{v}$. Any subset of a vector space that is closed under vector addition and scalar multiplication is call a subspace.

The simplest example of a linear vector space is just $\Re$ itself, which is a linear vector space over $\Re$. Vector addition and scalar multiplication are just addition and multiplication on $\Re$. For more on vector spaces and the other topics to follow in this section, see, for example, [2], [18], [17], or [23].

### 2.2.2 Banach Spaces

**Definition.** A norm of a vector space $V$, denoted by $||\cdot||$, is a nonnegative real valued function satisfying the following properties for all $\boldsymbol{u}, \boldsymbol{v} \in V$ and all $a \in \Re$,

1. Non-negative: $||\boldsymbol{u}|| \geq 0$.
2. Strictly positive: $||\boldsymbol{u}|| = 0$ implies $\boldsymbol{u} = 0$.
3. Homogeneous: $||a\boldsymbol{u}|| = |a| \; ||\boldsymbol{u}||$.
4. Triangle inequality: $||\boldsymbol{u} + \boldsymbol{v}|| \leq ||\boldsymbol{u}|| + ||\boldsymbol{v}||$.

**Definition.** A vector space is called a normed vector space when a norm has been defined on the space.

**Definition.** A sequence $\{\boldsymbol{v}_n\}$ in a normed vector space $V$ is said to converge to $\boldsymbol{v}_0 \in V$ if

$$\lim_{n \to \infty} ||\boldsymbol{v}_n - \boldsymbol{v}_0|| = 0.$$

**Definition.** A sequence $\{\boldsymbol{v}_n\} \subset V$ is called a Cauchy sequence if any given $\epsilon > 0$, there exists an integer $N$ such that $||\boldsymbol{v}_m - \boldsymbol{v}_n|| < \epsilon$ for any $m, n > N$.

Convergence of sequences in normed vector spaces follows the same general idea as sequences of real numbers except that the distance between two elements of the space is measured by the norm of the difference between the two elements.

**Definition (Banach Space).** A normed vector space $V$ is called complete if every Cauchy sequence in $V$ converges to an element of $V$. A complete normed vector space is called a Banach Space.

**Example 2.1.** $\Re$ with the absolute value norm $\|x\| \equiv |x|$ is a complete, normed vector space over $\Re$, and is thus a Banach space.

**Example 2.2.** Let $\boldsymbol{x} = (x_1, ..., x_n)^t$ be a point in $\Re^n$. For $1 \leq p < \infty$, the $l_p$ norm on $\Re^n$ is defined by

$$||\boldsymbol{x}||_p = \left[ \sum_{i=1}^{n} |x_i|^p \right]^{1/p} .$$

One can verify properties 1-4 at the beginning of this section for each $p$, validating that $||\boldsymbol{x}||_p$ is a norm on $\Re^n$.

### 2.2.3 Hilbert Spaces

A Hilbert Space is a Banach space in which the norm is defined by an inner-product (also called dot-product) which we define below. We typically denote Hilbert spaces by $H$. For elements $\boldsymbol{u}, \boldsymbol{v} \in H$, write the inner product of $\boldsymbol{u}$ and $\boldsymbol{v}$ either as $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_H$ or, when it is clear by context that the inner product is taking place in $H$, as $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$. If $H$ is a vector space over $F$, the result of the inner product is an element in $F$. We have $F = \Re$, so the result of an inner product will be a real number. The inner product operation must satisfy four properties for all $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in H$ and all $a \in F$.

1. Associative: $\langle a\boldsymbol{u}, \boldsymbol{v} \rangle = a\langle \boldsymbol{u}, \boldsymbol{v} \rangle$.

2. Commutative: $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \langle \boldsymbol{v}, \boldsymbol{u} \rangle$.

3. Distributive: $\langle \boldsymbol{u}, \boldsymbol{v} + \boldsymbol{w} \rangle = \langle \boldsymbol{u}, \boldsymbol{v} \rangle + \langle \boldsymbol{u}, \boldsymbol{w} \rangle$.

4. Positive Definite: $\langle \boldsymbol{u}, \boldsymbol{u} \rangle \geq 0$ with equality holding only if $\boldsymbol{u} = \boldsymbol{0}$.

**Definition.** A vector space with an inner product defined on it is called an inner-product space. The norm of an element $u$ in an inner-product space is taken as $||\boldsymbol{u}|| = \langle \boldsymbol{u}, \boldsymbol{u} \rangle^{1/2}$. Two vectors are said to be orthogonal if their inner product is 0 and two sets of vectors are said to be orthogonal if every vector in one is orthogonal to every vector in the other. The set of all vectors orthogonal to a subspace is called the orthogonal complement of the subspace. A complete inner-product space is called a Hilbert space.

**Example 2.3.** $\Re^n$ with inner product defined by

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle \equiv \boldsymbol{u}^t \boldsymbol{v} = \sum_{i=1}^{n} u_i v_i$$

is a Hilbert space. For any positive definite matrix $\boldsymbol{A}$, $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\boldsymbol{A}} \equiv \boldsymbol{u}^t \boldsymbol{A} \boldsymbol{v}$ also defines a valid inner product.

**Example 2.4.** Let $L_2(a, b)$ be the vector space of all functions defined on the interval $(a, b)$ that are square integrable and define the inner product

$$\langle f, g \rangle \equiv \int_a^b f(x) g(x) dx.$$

The inner-product space $L_2(a, b)$ is well-known to be complete; see [5], thus $L_2(a, b)$ is a Hilbert space.

## 2.3 Reproducing Kernel Hilbert Space (RKHS)

Hilbert spaces that display certain properties on certain linear operators are reproducing kernel Hilbert spaces.

**Definition.** A function $T$ mapping a vector space $X$ into another vector space $Y$ is called a linear operator if $T(\lambda_1 \boldsymbol{x}_1 + \lambda_2 \boldsymbol{x}_2) = \lambda_1 T(\boldsymbol{x}_1) + \lambda_2 T(\boldsymbol{x}_2)$ for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in X$ and any $\lambda_1, \lambda_2 \in \Re$.

Any $m \times n$ matrix $\boldsymbol{A}$ maps vectors in $\Re^n$ into vectors in $\Re^m$ via $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$ and is linear.

**Definition.** The operator $T : X \to Y$ is continuous at $\boldsymbol{x}_0 \in X$ if and only if for every $\epsilon > 0$ we have $\delta = \delta(\epsilon) > 0$ such that for every $\boldsymbol{x}$ with $||\boldsymbol{x} - \boldsymbol{x}_0|| < \delta$ implies $||T\boldsymbol{x} - T\boldsymbol{x}_0|| < \epsilon$.

Linear operators are continuous everywhere if they are continuous at 0.

**Definition.** A real valued function defined on a vector space is called a functional (i.e., a function from $V$ to $\Re$).

A $1 \times n$ matrix defines a linear functional on $\Re^n$.

**Example 3.1.** Let $S$ be the set of bounded real valued continuous functions $\{f(x)\}$ defined on the real line. Then $S$ is a vector space with the usual $+$ and $\times$ operations for functions. Some functionals on $S$ are $\phi(f) = \int_a^b f(x)dx$ and $\phi_a(f) = f'(a)$ for some fixed $a$. A functional of particular importance is the evaluation functional.

**Definition.** Let $H$ be a Hilbert space of functions defined from $E$ into $\Re$. For any $t \in E$, denote by $e_t$ the evaluation functional at the point $t$, i.e., for $g \in H$, the mapping is $e_t(g) = g(t)$.

Clearly, evaluation functionals are linear operators.

In a Hilbert space (or any normed vector space) of functions, the notion of pointwise convergence is related to the continuity of the evaluation functionals. The following are equivalent for a normed vector space $H$ of real valued functions.

(i) The evaluation functionals are continuous for all $t \in E$.

(ii) If $f_n$, $f \in H$ and $||f_n - f|| \to 0$ then $f_n(t) \to f(t)$ for every $t \in E$.

(iii) For every $t \in E$ there exists $K_t > 0$ such that $|f(t)| \leq K_t||f||$ for all $f \in H$.

Here (ii) is the definition of (i). See [17] for a proof of (iii).

To define a reproducing kernel, we need the famous *Riesz Representation Theorem.*

**Theorem.** Let $H$ be a Hilbert space and let $\phi$ be a continuous linear functional on

$H$. Then there is one and only one vector $g \in H$ such that

$$\phi(f) = \langle f, g \rangle, \qquad \text{for all } f \in H.$$

The vector $g$ is sometimes called the *representation* of $\phi$. However, $\phi$ and $g$ are different objects: $\phi$ is a linear functional on $H$ and $g$ is a point in $H$. For a proof of this theorem see [2] or [17].

For $H = \Re^p$, vectors can be viewed as functions from the set $E = \{1, 2, \ldots, p\}$ into $\Re$. An evaluation functional is $e_i(\boldsymbol{x}) = x_i$. The representation of this linear functional is the indicator vector $\boldsymbol{e}_i$ that is 0 everywhere except has a 1 in the $i$th place. Then

$$x_i = e_i(\boldsymbol{x}) = \boldsymbol{x}^t \boldsymbol{e}_i.$$

In fact, the entire representation theorem is well known in $\Re^p$ because for $\phi(\boldsymbol{x})$ to be a linear functional there must exist a vector $\boldsymbol{\phi}$ such that

$$\phi(\boldsymbol{x}) = \boldsymbol{\phi}^t \boldsymbol{x}.$$

An element of a set of functions, say $f$, is sometimes denoted $f(\cdot)$ to be explicit that the elements are functions, whereas $f(t)$ is the value of $f(\cdot)$ evaluated at $t \in E$. Applying the Riesz Representation Theorem to a Hilbert space $H$ of real valued functions in which evaluations functionals are continuous, for every $t \in E$ there is a unique symmetric function $R : E \times E \longrightarrow \Re$ with $R(\cdot, t) \in H$ the representation of $e_t$, so that

$$f(t) = e_t(f) = \langle f(\cdot), R(\cdot, t) \rangle_H, \qquad f \in H.$$

The function $R$ is called a *reproducing kernel* (r.k.) and $f(t) = \langle f(\cdot), R(\cdot, t) \rangle$ is called the *reproducing property* of $R$. In particular, by the reproducing property

$$R(s, t) = \langle R(\cdot, t), R(\cdot, s) \rangle.$$

In Section 1 we found the r.k. for an interpolating spline problem. Other examples follow shortly.

**Definition** A Hilbert space $H$ of functions defined on $E$ is called a reproducing kernel Hilbert space if all evaluation functionals are continuous.

**The projection principle in RKHS.**
We now consider the connection between the reproducing kernel $R$ of the RKHS $H$ and the reproducing kernel $R_0$ for a subspace $H_0 \subset H$. Suppose $H$ in an RKHS with r.k. $R$ and $H_0$ is a closed subset of $H$ and let $H_0^\perp$ be the orthogonal complement of $H_0$. Then any vector $f \in H$ can be written uniquely as $f = f_0 + f_1$ with $f_0 \in H_0$ and $f_1 \in H_0^\perp$. More particularly, $R(\cdot, t) = R_0(\cdot, t) + R_1(\cdot, t)$ with $R_0(\cdot, t) \in H_0$ and $R_1(\cdot, t) \in H_0^\perp$ if and only if $R_0$ is the r.k. of $H_0$ and $R_1$ is the r.k. of $H_0^\perp$. For a proof see [10].

## 2.3.1 Examples of Reproducing Kernel Hilbert Spaces

**Example 3.2.** Consider the space of all constant functionals over $\boldsymbol{x} = (x_1, \ldots, x_p)^t \in \Re^p$

$$H = \{f_\theta : f_\theta(\boldsymbol{x}) = \theta, \theta \in \Re\},$$

with $\langle f_\theta, f_\lambda \rangle = \theta\lambda$. (For simplicity, think of $p = 1$.) Since $\Re^p$ is a Hilbert Space, so is $H$. $H$ has continuous evaluation functionals, so it is an RKHS and has a unique reproducing kernel. To find the r.k., observe that $R(\cdot, \boldsymbol{x}) \in H$, so it is a constant for any $\boldsymbol{x}$. Write $R(\boldsymbol{x}) \equiv R(\cdot, \boldsymbol{x})$. By the representation theorem and the defined inner product

$$\theta = f_\theta(\boldsymbol{x}) = \langle f_\theta(\cdot), R(\cdot, \boldsymbol{x}) \rangle = \theta R(\boldsymbol{x})$$

for any $\boldsymbol{x}$ and $\theta$. This implies that $R(\boldsymbol{x}) \equiv 1$ so that $R(\cdot, \boldsymbol{x}) = R(\boldsymbol{x}) \equiv 1$ and $R(\cdot, \cdot) \equiv 1$.

**Example 3.3.** Consider all linear functionals over $\boldsymbol{x} \in \Re^p$ passing through the origin

$$H = \{f_{\boldsymbol{\theta}} : f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^t \boldsymbol{x},\ \boldsymbol{\theta} \in \Re^p\}.$$

Define $\langle f_{\boldsymbol{\theta}}, f_{\boldsymbol{\lambda}} \rangle = \boldsymbol{\theta}^t \boldsymbol{\lambda} = \theta_1\lambda_1 + \theta_2\lambda_2 + \ldots + \theta_p\lambda_p$. The kernel $R$ must satisfy

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \langle f_{\boldsymbol{\theta}}(\cdot), R(\cdot, \boldsymbol{x}) \rangle$$

for all $\boldsymbol{\theta}$ and any $\boldsymbol{x}$. Since $R(\cdot, \boldsymbol{x}) \in H$, $R(\boldsymbol{v}, \boldsymbol{x}) = \boldsymbol{u}^t \boldsymbol{v}$ for some $\boldsymbol{u}$ that depends on $\boldsymbol{x}$, i.e., $R(\cdot, \boldsymbol{x}) = f_{\boldsymbol{u}(\boldsymbol{x})}(\cdot)$, so $R(\boldsymbol{v}, \boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x})^t \boldsymbol{v}$. By our definition of $H$ we have

$$\boldsymbol{\theta}^t \boldsymbol{x} = f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \langle f_{\boldsymbol{\theta}}(\cdot), R(\cdot, \boldsymbol{x}) \rangle = \langle f_{\boldsymbol{\theta}}(\cdot), f_{\boldsymbol{u}(\boldsymbol{x})}(\cdot) \rangle = \boldsymbol{\theta}^t \boldsymbol{u}(\boldsymbol{x}),$$

so we need $\boldsymbol{u}(\boldsymbol{x})$ such that for any $\boldsymbol{\theta}$ and $\boldsymbol{x}$ we have

$$\boldsymbol{\theta}^t \boldsymbol{x} = \boldsymbol{\theta}^t \boldsymbol{u}(\boldsymbol{x}).$$

It follows that $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{x}$. For example, taking $\boldsymbol{\theta}$ to be indicator vector $\boldsymbol{e}_i$ implies that $u_i(\boldsymbol{x}) = x_i$ for every $i = 1, \ldots, p$. We now have $R(\cdot, \boldsymbol{x}) = f_{\boldsymbol{x}}(\cdot)$ so that

$$R(\tilde{\boldsymbol{x}}, \boldsymbol{x}) = \boldsymbol{x}^t \tilde{\boldsymbol{x}} = x_1\tilde{x}_1 + x_2\tilde{x}_2 + \ldots + x_p\tilde{x}_p.$$

**Example 3.4.** Now consider all affine functionals in $\Re^p$

$$H = \{f_{\boldsymbol{\theta}} : f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \theta_0 + \theta_1 x_1 + \ldots + \theta_p x_p,\ \boldsymbol{\theta} \in \Re^{p+1}\}$$

with $\langle f_{\boldsymbol{\theta}}, f_{\boldsymbol{\lambda}} \rangle = \theta_0\lambda_0 + \theta_1\lambda_1 + \ldots + \theta_p\lambda_p$. The subspace $H_0 = \{f_{\boldsymbol{\theta}} \in H : \theta_0 \in \Re, 0 = \theta_1 = \cdots = \theta_p\}$ has the orthogonal complement $H_0^\perp = \{f_{\boldsymbol{\theta}} \in H : 0 = \theta_0\}$. For practical purposes, $H_0$ is the space of constant functionals from Example 3.2 and $H_0^\perp$ is the space of linear functionals from Example 3.3. Note that the inner product on $H$ when applied to vectors in $H_0$ and $H_0^\perp$, respectively, reduces to the inner products used in Examples 3.2 and 3.3.

Write $H$ as $H = H_0 \oplus H_0^\perp$ where $\oplus$ denotes the *direct sum* of two vector spaces. For two subspaces $A$ and $B$ contained in a vector space $C$, the direct sum is the space $D = \{a + b : a \in A, b \in B\}$. Any elements $d_1, d_2 \in D$ can be written as $a_1 + b_1$ and $a_2 + b_2$, respectively for some $a_1, a_2 \in A$ and $b_1, b_2 \in B$. When the two subspaces are orthogonal, as in our example, those decompositions are unique and the inner product between $d_1$ and $d_2$ is $\langle d_1, d_2\rangle = \langle a_1, a_2\rangle + \langle b_1, b_2\rangle$. For more information about direct sum decomposition, see [10], for example.

We have already derived the r.k.'s for $H_0$ and $H_0^\perp$ (call them $R_0$ and $R_1$, respectively) in Examples 3.2 and 3.3. Applying the projection principle, the r.k. for $H$ is the sum of $R_0$ and $R_1$, i.e.,

$$R(\tilde{\boldsymbol{x}}, \boldsymbol{x}) = 1 + \boldsymbol{x}^t\tilde{\boldsymbol{x}}.$$

**Example 3.5.** Denote by $H$ the collection of functions $f$ with $f'' \in L^2[0,1]$ and consider the subspace

$$W_2^0 = \{f(x) \in H : f, f' \text{ absolutely continuous and } f(0) = f'(0) = 0\}.$$

Define the inner product on $H$ as

$$\langle f, g\rangle = \int_0^1 f''(t)g''(t)dt. \tag{2.10}$$

Below we demonstrate that for $f \in W_2^0$ and any $s$, $f(s)$ can be written as

$$f(s) = \int_0^1 (s-u)_+ f''(u)du\,, \tag{2.11}$$

where $(a)_+$ is $a$ for $a > 0$ and 0 for $a \le 0$. Given any arbitrary and fixed $s \in [0,1]$,

$$\int_0^1 (s-u)_+ f''(u)du = \int_0^s (s-u)f''(u)du\,.$$

Integrating by parts

$$\int_0^s (s-u)f''(u)du = (s-s)f'(s) - (s-0)f'(0) + \int_0^s f'(u)du = \int_0^s f'(u)du$$

or, by the Fundamental Theorem of Calculus,

$$\int_0^s (s-u)f''(u)du = f(s) - f(0) = f(s)\,.$$

Since the r.k. of the space $W_2^0$ must satisfy $f(s) = \langle f(\cdot), R(\cdot, s)\rangle$ from (2.10) and (2.11) we see that $R(\cdot, s)$ is a function such that

$$\frac{d^2 R(u,s)}{d^2 u} = (s-u)_+.$$

We also know that $R(\cdot, s) \in W_2^0$, so using $R(s,t) = \langle R(\cdot, t), R(\cdot, s)\rangle$

$$R(s,t) = g_s(t) = \int_0^1 (t-u)_+(s-u)_+ du = \frac{\max(s,t)\min^2(s,t)}{2} - \frac{\min^3(s,t)}{6}.$$

For further examples of RKHSs with various inner products, see [3].

## 2.4 Penalized Regression with RKHSs

We start this section with two common examples of penalized regression: ridge regression and smoothing splines.

**Ridge Regression.** In the classical linear regression setting the ridge regression estimator $\hat{\boldsymbol{\beta}}_R$ proposed by [14] minimizes

$$\sum_{i=1}^{n} \left( y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{2.12}$$

where $x_{ij}$ is the $i$-th observation of the $j$-th predictor. The resulting estimate is biased but it reduces the variance over the traditional least squares estimate. The tuning parameter $\lambda \geq 0$ is a constant that controls the trade-off between bias and variance in $\hat{\boldsymbol{\beta}}_R$, and is often selected by some form of cross validation; see Section 4.4.

**Smoothing Splines.** Nonparametric regression is a powerful approach for solving many modern problems such as computer modeling, image processing, and environmental monitoring. The nonparametric regression model is

$$y_i = f(x_i) + \epsilon_i, \quad i = 1, 2, \ldots, n,$$

where $f$ is an unknown regression function and the $\epsilon_i$ are independent error terms.

Smoothing splines are among the most popular methods for the estimation of $f$, due to their good empirical performance and sound theoretical support. It is usually assumed, without loss of generality, that the domain of $f$ is $[0, 1]$. With $f^{(m)}$ the $m$-th derivative of $f$, a smoothing spline estimate $\hat{f}$ is the unique minimizer of

$$\sum_{i=1}^{n} \{y_i - f(x_i)\}^2 + \lambda \int f^{(m)}(x)^2 dx. \tag{2.13}$$

The minimizer depends on both $m$ and $\lambda$.

The minimization of (3.2) is implicitly over functions with square integrable $m$-th derivatives. The first term of (3.2) encourages the fitted $f$ to be close to the data, while the second term penalizes the roughness of $f$. The smoothing parameter $\lambda$, usually pre-specified, controls the trade-off between the two conflicting goals. The special case of $m = 1$ reduces to the linear smoothing spline problem from (2.9). In practice, it is common to choose $m = 2$ in which case the minimizer $\hat{f}_\lambda$ of (3.2) is called a cubic smoothing spline. As $\lambda \to \infty$, $\hat{f}_\lambda$ approaches the least squares simple linear regression line, while as $\lambda \to 0$, $\hat{f}_\lambda$ approaches the minimum curvature interpolant.

### 2.4.1 Solving the General Penalized Regression Problem

We now review a general framework to minimize (2.12), (3.2) and many other similar problems, cf. [29, 28]. The data model associated with the general spline smoothing problem is

$$y_i = f(x_i) + \epsilon_i, \quad i = 1, 2, \ldots, n, \tag{2.14}$$

where $f \in H_R$, a given RKHS of functions on a set $E$, and $\epsilon_i$ are error terms. Suppose $H_R$ is the direct sum of two orthogonal subspaces

$$H_R = H_0 \bigoplus H_1,$$

and $H_0$ is a subspace of functions that can be represented by $M \leq n$ basis functions, $\phi_1, \ldots, \phi_M$. Any $f \in H_R$ can be written uniquely as $f = f_0 + f_1$ with $f_0 = \sum_{j=1}^{M} d_j \phi_j(x)$ and $f_1 \in H_1$. The function $f_1$ is called the orthogonal projection of $f$ onto $H_1$ and is written $P_1 f$.

An estimate of $f$ is obtained by minimizing

$$\frac{1}{n}\sum_{i=1}^{n}\{y_i - f(x_i)\}^2 + \lambda ||P_1 f||_R^2, \tag{2.15}$$

The penalty is placed on the norm of the $H_1$ component of the function $f$, while no penalty is placed on the $H_0$ component. In other words, $H_0$ represents the null space of the penalty.

The key result is that the minimizer is a linear combination of known functions involving the reproducing kernel on $H_1$.

**Representation Theorem.** The minimizer $\hat{f}_\lambda$ of equation (2.15) has the orthogonal decomposition

$$\hat{f}_\lambda(x) = \sum_{j=1}^{M} d_j \phi_j(x) + \sum_{i=1}^{n} c_i R(x_i, x), \tag{2.16}$$

where $R(s,t)$ is the r.k. for $H_1$. See [31] or [10] for a proof.

Once you know that the minimizer takes this form, finding the coefficients reduces to a quadratic minimization problem similar to whose in standard linear models. This occurs because we can write $||P_1 \hat{f}_\lambda||_R^2$ as a quadratic form in $\boldsymbol{c} = (c_1, \ldots, c_n)^t$. Define $\boldsymbol{\Sigma}$ as the $n \times n$ matrix where the $i,j$ entry is $\boldsymbol{\Sigma}_{ij} = R(x_i, x_j)$. Now, using the reproducing property of $R$, write

$$\begin{aligned}\left\|P_1 \hat{f}_\lambda\right\|_R^2 = \left\|\sum_{i=1}^{n} c_i R(x_i, \cdot)\right\|_R^2 &= \left\langle \sum_{i=1}^{n} c_i R(x_i, \cdot), \sum_{j=1}^{n} c_j R(x_j, \cdot)\right\rangle \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j R(x_i, x_j) = \boldsymbol{c}^t \boldsymbol{\Sigma} \boldsymbol{c}.\end{aligned}$$

Define the $n \times M$ matrix $\mathbf{T}$ with $ij$-th entry $t_{ij} = \{\phi_j(x_i)\}$. This typically has full column rank. The vector of predicted values $\hat{\boldsymbol{y}} = (\hat{f}_\lambda(x_1), ..., \hat{f}_\lambda(x_n))^t$ can now be written as

$$\hat{\boldsymbol{y}} = \mathbf{T}\boldsymbol{d} + \boldsymbol{\Sigma}\boldsymbol{c}$$

and the minimization problem associated with (2.15) now takes the form

$$\min_{(\boldsymbol{d},\boldsymbol{c})} \frac{1}{n} \left[\boldsymbol{y} - (\mathbf{T}\boldsymbol{d} + \boldsymbol{\Sigma}\boldsymbol{c})\right]^t \left[\boldsymbol{y} - (\mathbf{T}\boldsymbol{d} + \boldsymbol{\Sigma}\boldsymbol{c})\right] + \lambda \boldsymbol{c}^t \boldsymbol{\Sigma}\boldsymbol{c}. \tag{2.17}$$

In particular, the solution is the generalized least squares estimate from fitting the linear model

$$\begin{bmatrix} \boldsymbol{y} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{T} & \boldsymbol{\Sigma} \\ \mathbf{0}_{n\times M} & \boldsymbol{I}_{n\times n} \end{bmatrix} \begin{bmatrix} \boldsymbol{d} \\ \boldsymbol{c} \end{bmatrix} + e, \quad \mathrm{Cov}(e) \propto \begin{bmatrix} \boldsymbol{I}_{n\times n} & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & \boldsymbol{\Sigma}^{-1}_{n\times n} \end{bmatrix}.$$

Alternatively, define the partitioned matrices,

$$\mathbf{Q}_{n\times(n+M)} = \begin{bmatrix} \mathbf{T}_{n\times M} & \boldsymbol{\Sigma}_{n\times n} \end{bmatrix}, \qquad \boldsymbol{\gamma}_{(n+M)\times 1} = \begin{bmatrix} \mathbf{d}_{M\times 1} \\ \mathbf{c}_{n\times 1} \end{bmatrix},$$

and

$$\mathbf{S}_{(n+M)\times(n+M)} = \begin{bmatrix} \mathbf{0}_{M\times M} & \mathbf{0}_{M\times n} \\ \mathbf{0}_{n\times 1} & \boldsymbol{\Sigma}_{n\times n} \end{bmatrix}$$

so that (2.17) becomes

$$\min_{\boldsymbol{\gamma}} \frac{1}{n} ||\mathbf{y} - Q\boldsymbol{\gamma}||^2 + \lambda \boldsymbol{\gamma}^t S \boldsymbol{\gamma}.$$

Taking derivatives with respect to $\boldsymbol{\gamma}$ we have

$$(\mathbf{Q}^t\mathbf{Q} + \lambda S)\hat{\gamma} = \mathbf{Q}^t\mathbf{y},$$

which requires solving a system of $n+M$ equations to find $\hat{\boldsymbol{\gamma}}$. For analytical purposes, we can write $\hat{\boldsymbol{\gamma}}$ as

$$\hat{\gamma} = (\mathbf{Q}^t\mathbf{Q} + \lambda\mathbf{S})^{-1}\mathbf{Q}^t\mathbf{y}. \tag{2.18}$$

For clarity, we have restricted attention to the usual case of squared error loss between the observations and the unknown function evaluations. The Representation Theorem also generalizes to other convex loss functions and to more general observation equations than that in (2.14); see [31] and [10].

### 2.4.2 General Solution Applied to Ridge Regression

Here we will solve the linear ridge regression problem in (2.12) with the RKHS approach detailed above. The RKHS framework is not necessary to solve the problem for practical purposes, but this simple problem serves as a good illustration of the RKHS machinery. Consider the following space of functions

$$F = \{f(\boldsymbol{x}) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j\}. \tag{2.19}$$

We define the following inner product for the elements in $F$

$$\langle f, g \rangle = \sum_{j=1}^{p} \beta_{f_j} \beta_{g_j}.$$

Notice we can write $H = H_0 \bigoplus H_1$ where $H_0$ is from Example 3.2, $H_1$ is from Example 3.3 and $H = F$. Now the general smoothing spline problem of (2.15) becomes

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \sum_{j=1}^{p} \beta_j^2.$$

In this case, the dimension of $H_0$ is $M = 1$ and $\phi_1(x) = 1$. So from (2.16) the solution takes the form

$$\hat{f}_\lambda(\boldsymbol{x}) = \hat{d}_1 + \sum_{i=1}^{n} \hat{c}_i R(x_i, x) \tag{2.20}$$

with

$$(\mathbf{Q}^t\mathbf{Q} + \lambda\mathbf{S})^{-1}\mathbf{Q}^t\mathbf{y} = \begin{bmatrix} \hat{d}_1 \\ \hat{\mathbf{c}} \end{bmatrix} \tag{2.21}$$

as given by (2.18). In this case it is more informative to write the solution in form

$$\hat{f}(\boldsymbol{x}) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots + \hat{\beta}_p x_p.$$

This can be done as follows, the solution in (2.20) is

$$\hat{f}(\boldsymbol{x}) = \hat{d}_1 + \hat{c}_1 R(x_1, x) + \hat{c}_2 R(x_2, x) + \ldots + \hat{c}_n R(x_n, x).$$

Recall from Example 3.3 that in this case

$$R(\boldsymbol{x}_i, \boldsymbol{x}) = x_{i1}x_1 + x_{i2}x_2 + \ldots + x_{ip}x_p,$$

$$\hat{f}(\boldsymbol{x}) = \hat{d}_1 + \sum_{i=1}^{n} \hat{c}_i x_{i1} x_1 + \sum_{i=1}^{n} \hat{c}_i x_{i2} x_2 + \ldots + \sum_{i=1}^{n} \hat{c}_i x_{ip} x_p,$$

which implies in this case that

$$\hat{\beta}_0 = \hat{d}_1 \quad \text{and} \quad \hat{\beta}_j = \sum_{i=1}^{n} \hat{c}_i x_{ij}$$

for $j = 1, 2, \ldots, p$. In Appendix A, we provide a demonstration of this solution on some actual data using the statistical software R.

### 2.4.3 General Solution Applied to Cubic Smoothing Spline

Consider again the regression problem $y_i = f(x_i) + \epsilon_i, i = 1, 2, \ldots, n$ where $x_i \in [0, 1]$ and $\epsilon_i \sim N(0, \sigma^2)$. We shall focus on the cubic smoothing spline solution to this problem. That is, we find the function that minimizes

$$\sum_{i=1}^{n} \{y_i - f(x_i)\}^2 + \lambda \int f''(x)^2 dx.$$

In this case $H_0$ is the space of linear functions from (2.19) with $p = 1$ and $H_1$ is given in Example 3.5. We know that the reproducing kernel for this space is

$$R(s, t) = \int_0^1 (t - u)_+ (s - u)_+ du = \frac{\max(s, t) \min^2(s, t)}{2} - \frac{\min^3(s, t)}{6}.$$

Here a basis for $H_0$ is $\phi_1(x) = 1$, $\phi_2(x) = x$ and from the Representation Theorem, we know that the solution has the form

$$\hat{f}(x) = \hat{d}_0(1) + \hat{d}_1 x_i + \sum_{i=i}^{n} \hat{c}_i R(x_i, x).$$

From (2.18) we have

$$(\mathbf{Q}^t\mathbf{Q} + \lambda\mathbf{S})^{-1}\mathbf{Q}^t\mathbf{y} = \begin{bmatrix} \hat{\mathbf{d}} \\ \hat{\mathbf{c}} \end{bmatrix} \tag{2.22}$$

Appendix A provides a demonstration with R code of fitting the cubic smoothing spline using the solution in (2.22) on some actual data. The demonstration also includes searching for the best value of the tuning parameter $\lambda$ which is briefly discussed in the next subsection.

### 2.4.4 Choosing the Degree of Smoothness

With the penalized regression procedures described earlier, choosing the smoothing parameter $\lambda$ is an important issue. There are many methods available including visual inspection of the fit, cross-validation, generalized maximum likelihood estimation, and generalized cross-validation (GCV).

For the examples given in the appendix, we use GCV. Given a closed-form solution as in equation (2.18), for example (2.21) or (2.22), the GCV choice of $\lambda$ minimizes

$$V(\lambda) = \frac{1}{n}||(\mathbf{I} - \mathbf{A}(\lambda)\mathbf{y}||^2/[\frac{1}{n}\text{Trace}\{\mathbf{I} - \mathbf{A}(\lambda)\}]^2,$$

where

$$\mathbf{A}(\lambda) = \mathbf{Q}(\mathbf{Q}^t\mathbf{Q} + \lambda\mathbf{S})^{-1}\mathbf{Q}^t.$$

The goal of GCV is to find $\lambda$ so that the resulting estimate has the smallest mean squared error. For more details about GCV and other methods of finding $\lambda$ see [9], [1], [33] and [31].

## 2.5 Concluding Remarks

In this chapter we have given several examples motivating the utility of the RKHS approach to penalized regression problems. We reviewed the building blocks necessary to define an RKHS and presented several key results about these spaces. Finally, we used these results to perform illustrative estimation for ridge regression and the cubic smoothing spline problems, and presented transparent R code to enhance understanding of the examples.

# Chapter 3

# Spatially Adaptive Smoothing Spline

## 3.1 Introduction

## 3.2 Introduction

Nonparametric Regression has proven to be a very useful methodology, with applications to a large list of modern problems such as computer models, image data, environmental processes, to name a few. The nonparametric regression model is given by

$$y_i = f_0(x_i) + \epsilon_i,\ i = 1, 2, ..., n \tag{3.1}$$

where $f_0$ is an unknown regression function and $\epsilon_i$ are independent error terms.

Smoothing splines are among the most popular methods for estimation of $f_0$ due to their good empirical performance and sound theoretical support [4, 26, 6, 30]. It is usually assumed without loss of generality that the domain of $f_0$ is [0,1]. Let

$f^{(m)}$ denote the $m^{th}$ derivative of $f$. The smoothing spline estimate $\hat{f}$ is the unique minimizer of

$$\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int_0^1 \left(f^{(m)}(x)\right)^2 dx \tag{3.2}$$

over all functions, $f$, in $m^{th}$ order Sobolev space,

$$S^m = \{f : f^{(j)} \text{ is absolutely continuous for } j = 1, ..., m-1 \text{ and } f^{(m)} \in L_2\}$$

The minimizer of (3.2) trades off fidelity to the data (in terms of residual sum of squares) against smoothness of the reconstructed curve (in terms of the integrated squared derivative of order $m$, where $m$ is typically taken to be two). The smoothing spline solution uses a global smoothing parameter $\lambda$ which implies that the true underlying mean process has a constant degree of smoothness. In this chapter we suggest a more general, "spatially adaptive", framework that accommodates varying degrees of roughness by seeking solutions where the smoothness penalty depends on the region of the domain being fitted. We derive the solution within a Reproducing Kernel Hilbert Space framework [31, 10].

There are many approaches to surface fitting using spatially adaptive knot placement (basis function selection) with regression splines; see [8], [27], [16], and [11]. However, the properties of these estimators are difficult to study analytically since they are the result of an algorithm and not an explicit solution to an optimization problem. [22] use a piecewise constant function for $\lambda$ in (3.2). However, this form of $\lambda(x)$ requires specifying the number of knots, the knot locations, and the values of $\lambda(x)$ in between knot locations. This was accomplished by selecting one of several candidate knot location options and $\lambda$ values between the knots via GCV. Unfortunately this leads to a smoothing method with large number of smoothing parameters whose values need to be selected. The Loco-Spline procedure of [28] uses a spatially varying penalty based on an initial estimate. The final estimate is penalized less where the initial estimate indicates more curvature is needed. However, this procedure can be unstable for small sample sizes and is computationally expensive for larger samples.

We propose a method which breaks down the interval $[0, 1]$ into $p$ disjoint sub-intervals. Then we define $p$ functional components in $[0, 1]$, which have two important features. First, the purpose of each of these $p$ components is to estimate the true function locally, i.e., in only one of the sub-intervals. Second, even though all components are defined on the entire domain, i.e. $[0, 1]$, a component has curvature only in one of the afore mentioned intervals. The $p$ local estimates are then added together to produce a function estimate over the entire $[0, 1]$ interval. This is similar in spirit to the method of [22]. However, in the proposed method, the additional flexibility that comes from finding these $p$ local functional estimates does not come at any additional computational cost. In spite of having $p$ components there is no need to specify (e.g., choose via cross validation) $p$ smoothing parameters. Theory from COmponent Selection and Shrinkage Operator (COSSO) [15], reduces the problem of specifying these $p$ smoothing parameters to specifying only one smoothing parameter without a loss in flexibility. In fact, empirical studies indicate superior performance of COSSO in the additive model framework over that for the traditional additive model [12], see [29], for example.

In Section 3.3 of this chapter we review the COSSO framework that we will use to solve for our proposed estimator. In Section 3.4, we present the Locally Adaptive COmponent Selection and Shrinkage Operator (LACOSSO), a new method for spatially adaptive nonparametric regression. We discuss the main differences between COSSO and LACOSSO and also the advantages of LACOSSO over its competitors. Section 3.5 is devoted to computational details, e.g., finding the reproducing kernel of the functional spaces needed to solve the proposed optimization problem. In Section 3.6 we state a theorem that establishes the optimal MSE convergence rate of LACOSSO. The proof of this result can be found in the appendix. Section 3.7 presents results from a simulation study and an example dataset to compare LACOSSO to other existing methods. In all the examples presented, LACOSSO's performance is better, or comparable, to the performance shown by its competitors. Section 3.8 concludes the chapter with some closing remarks and mentions areas which we would like to explore in the future.

## 3.3 Smoothing Spline ANOVA and COSSO

In this section we review only the necessary concepts of Smoothing Spline (SS)-ANOVA needed for the development of LACOSSO. For a more detailed overview of Smoothing Splines and SS-ANOVA see [31], [32], [24], [10], and [3]. For a gentle introduction to RKHS and penalized regression, see [19].

In the smoothing spline literature it is typically assumed that $f \in F$ where $F$ is a reproducing kernel Hilbert space (RKHS). Denote the reproducing kernel (r.k.), inner product, and norm of $F$ as $K_F$, $\langle \cdot, \cdot \rangle_F$, and $|| \cdot ||_F$ respectively. Often $F$ is chosen to contain only functions with a certain degree of smoothness. For example, functions on one variable are often assumed to belong to the second order Sobolev space, $S^2 = \{f : f, f' \text{ are absolutely continuous and } f'' \in L^2[0,1]\}$.

A RKHS $F$ can always be written as

$$F = \{1\} \oplus \{\bigoplus_{j=1}^{p} F_j\}, \tag{3.3}$$

where $\oplus$ represents the direct sum operation, $F_1, ..., F_p$ is some orthogonal decomposition of the space, and each of the $F_j$ is itself a RKHS. A familiar example of such a decomposition is the additive model $f(\boldsymbol{x}) = b_0 + \sum_{j=i}^{p} f_j(x_j)$ when there is more than one predictor.

A traditional smoothing spline type method finds $\hat{f} \in F$ to minimize

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda\sum_{j=1}^{p}\theta_j^{-1}||P^j f||^2 \tag{3.4}$$

where $P^j f$ is the orthogonal projection of $f$ onto $F_j$ and $\theta \geq 0$. If $\theta_j = 0$, then the minimizer is taken to satisfy $||P^j f||^2 = 0$. We use the convention $0/0 = 0$ throughout this paper. The smoothing parameter $\lambda$ is confounded with the $\theta' s$, but is usually included in the setup for computational purposes.

The COSSO [15] penalizes on the sum of the norms instead of the squared norms as in the traditional smoothing spline and hence achieves sparse solutions (e.g. some of the functional components are estimated to be exactly zero). Specifically, the COSSO estimate, $\hat{f}$, is given by the function $\hat{f} \in F$ that minimizes

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda\sum_{j=1}^{p}||P^j f||_F \tag{3.5}$$

where $\lambda$ is a smoothing parameter.

The Adaptive COSSO (ACOSSO) improves upon COSSO by using individually weighted norms to smooth each of the components. Specifically, ACOSSO selects as the estimate the function $f \in F$ that minimizes

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda\sum_{j=1}^{p}w_j\|P^j f\|_F \tag{3.6}$$

where $0 < w_j < \infty$ are weights that can depend on an initial estimate of $f$ which we denote $\tilde{f}$. The $w_j$s are not tuning parameters in the sense that they would need to be chosen by cross validation. For more details about ACOSSO see [29].

Finally, it is possible to give an equivalent form of (3.6) that is useful for computational purposes. Consider the problem of finding $[\theta_1, ..., \theta_p] \in \Re^p$ and $f \in F$ to minimize

$$\min_{f \in F}\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda_0\sum_{j=1}^{p}\frac{w_j^2}{\theta_j}\|P^j f\|_F^2 + \lambda_1\sum_{j=1}^{p}\theta_j \tag{3.7}$$

subject to $\theta_j \geq 0$, $j = 1, ..., p$, $\lambda_0$ is a constant that can be fixed at any positive value, and $\lambda_1$ is a smoothing parameter. For a given $\lambda$ in (3.6), there is a value of $\lambda_1$ in (3.7) that will result in the same minimizing function $\hat{f}$. See [29] for a proof of this equivalence. We don't typically care which value of $\lambda_1$ corresponds to which value of $\lambda$, since the smoothing parameter is usually not prespecified, rather it is chosen based on some goodness of fit measure anyhow. Notice, that the minimization in (3.7) has the same flexibility as a minimization with $p$ smoothing parameters $\theta_1, \ldots, \theta_p$. However, the $\theta_j$ are treated as if they are additional model parameters, then they are also penalized (in the last term). This is similar to modeling the $\theta_j$ with a hyper-prior in a hierarchical Bayesian framework.

## 3.4 A locally Adaptive Estimator

Notice that the penalty term on the right of (3.2) is an overall measure of the roughness of the function over the domain. The tuning parameter $\lambda$ controls the trade-off in the resulting estimate between smoothness and fidelity to the data; large values of $\lambda$ will result in smoother functions while smaller values of $\lambda$ result in rougher functions but with better agreement to the data. In many cases the underlying function changes more abruptly in some regions than in others. In situations like this the global penalty will cause the smoothing spline estimator to either over-smooth in some regions and/or under-smooth in others [28].

Here we consider spatially adaptive estimators which are defined by the explicit function minimization problem,

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \sum_{j=1}^{p} w_j \left\{ \int_{\tau_{j-1}}^{\tau_j} [f''(x)]^2 dx \right\}^{1/2} \tag{3.8}$$

over all functions, $f \in S^2$, for given knots $0 = \tau_0 < \tau_1 < \cdots < \tau_p = 1$. The knots need to be prespecified (they could be chosen to be equally spaced on the quantiles of $x$, for example).

An equivalent minimization to (3.8) which is more convenient for computational purposes is

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - b_0 - b_1 x_i - \sum_{j=1}^{p} f_j(x_i)\right)^2 + \lambda \sum_{j=1}^{p} w_j \left\{\int_{\tau_{j-1}}^{\tau_j} [f_j''(x)]^2 dx\right\}^{1/2} \tag{3.9}$$

over $b_0, b_1 \in \Re$, and all functions $f_1 \in S_1^*, \ldots, f_p \in S_p^*$, where

$S_j^* = \{f : f$ and $f'$ absolutely continuous, $f'' \in L_2$ , with $f(x) = 0$ if $x \in [0, \tau_{j-1})$, $f$ is linear for $x \in (\tau_j, 1]\}$ $j = 1, \ldots, p$.

The proposed estimator in (3.8) has several important properties. First, this formulation allows for the functional estimate to vary adaptively with $x$ allowing for more/less penalty in regions of the domain where it is beneficial. This is accomplished by breaking the function down into the $p$ functional components. Second, we are not penalizing the squared norm, rather the norm of each of these $p$ variables, as in the COSSO framework. In doing so our estimator also inherits the computational advantages from COSSO as discussed further in Section 3.5. Third, we have $p$ new elements in our minimization problem, $w_1, w_2, ..., w_p$. However, these are not smoothing parameters (i.e., we do not need to estimate them), rather they are weights that can depend on an initial estimate of $f$ which we denote $\tilde{f}$. For example, we could initially estimate $f$ via the traditional smoothing spline (we will discuss a particular way of finding these quantities in the next subsection). Finally, we only have one smoothing parameter to choose via cross validation or similar means which keeps computation more feasible.

### 3.4.1 Specifying $w_j$.

Given an initial estimate $\tilde{f}$, we wish to construct $w_j$'s so that the prominent functional components enjoy the benefit of a smaller penalty relative to less important functional components. In contrast to the adaptive LASSO procedure for linear models ([34]), there is no single coefficient, or set of coefficients, to measure importance of a variable. One possible scheme would be to make use of an estimate of the RKHS norm used in the COSSO-like penalty and set

$$w_j = \|\tilde{f}_j\|_F^{-\gamma}. \tag{3.10}$$

We suggest the following procedure to specify the $w_j$s:

1. Set $w_j = 1$ for $j = 1, 2, ..., p$ in (3.8). Note that by doing this we are placing the same importance on each of the functional components. With this choice of $w_j$'s (3.8) becomes

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \sum_{j=1}^{p} \|P^j f\|_F. \tag{3.11}$$

   The solution to the above minimization problem gives us $\tilde{f}$.

2. Set $w_j = \|P^j \tilde{f}\|_F^{-\gamma}$, for some parameter $\gamma$. We have found that setting $\gamma = 1$ or $\gamma = 2$ provides good results in practice.

## 3.5 Computation

### 3.5.1 Solving LACOSSO with the RKHS framework

If we endow each of the $S_j^*$ with the inner product

$$\langle f, g \rangle = \int_0^1 f''(x)g''(x)dx$$

then each of the $S_j^*$ are orthogonal in the space $F = \bigoplus_j S_j^*$. It then becomes clear that (3.9) is a special case of (3.6). Thus, using the equivalence of (3.6) and (3.7), we can write the minimization in (3.9) as the minimizer of

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - b_0 - b_1 x_i - \sum_{j=1}^{p} f_j(x_i)\right)^2 + \lambda_0 \sum_{j=1}^{p} \frac{w_j^2}{\theta_j}\left\{\int_{\tau_{j-1}}^{\tau_j} [f_j''(x)]^2 dx\right\} + \lambda_1 \sum_{j=1}^{p} \theta_j \quad (3.12)$$

over $b_0, b_1 \in \Re$, and all functions $f_1 \in S_1^*, \ldots, f_p \in S_p^*$.

The algorithm used to solve (3.7) is detailed in [29] and can be used here to solve (3.12) as well. It hinges on the representation theorem of [31] to write the solution to (3.12) in the form

$$\hat{f}(x) = \hat{b}_0 + \hat{b}_1 + \sum_{i=1}^{n} \hat{c}_i \sum_{j=1}^{p} R_j(x_i, x) \quad (3.13)$$

The main ingredient needed in the ACOSSO algorithm is thus the reproducing kernel $R_j$ for each orthogonal subspace $S_j^*$. These reproducing kernels are presented in the next subsection. First, we present a simple form for the $w_j$ in (3.10) based on the initial estimate $\tilde{f}$. Write the initial estimate in the form $\tilde{f} = \tilde{b}_0 + \tilde{b}_1 + \sum_{i=1}^{n} \tilde{c}_i R_j(x_i, x)$ as given in (3.13). We know that $\tilde{f}_j$ is given by the orthogonal projection of $\tilde{f}$ onto $H_j$, which is $P^j \tilde{f} = \sum_{i=1}^{n} \tilde{c}_i R_j(x_i, \cdot)$. Hence,

$$\|P^j \tilde{f}\|_F^2 = \langle \sum_{i=1}^{n} \tilde{c}_i R_j(x_i, \cdot), \sum_{i=1}^{n} \tilde{c}_i R_j(x_i, \cdot) \rangle = \tilde{\boldsymbol{c}}' \boldsymbol{\Sigma} \tilde{\boldsymbol{c}}, \quad (3.14)$$

or

$$w_j = (\tilde{\boldsymbol{c}}' \boldsymbol{\Sigma} \tilde{\boldsymbol{c}})^{-\gamma/2} \quad (3.15)$$

### 3.5.2 Finding the Reproducing Kernels

Finding the reproducing kernel (r.k.) directly for the $S_j^*$ would be difficult. Hence, we instead make use of the connection between a RKHS $H$ with reproducing kernel $R(s,t)$ and a Gaussian Process (GP) with covariance $K(s,t) = R(s,t)$. The connection is based on the following result, let $\{X(t), t \in T\}$ be a real Gaussian

Stochastic Process defined on a probability space, with mean function $E[X(t)] = 0$ and covariance $K(s,t) = E[X(t)X(s)]$. It is well known, see [20], that $K$ determines a Hilbert space $H(K)$, called the RKHS of $K$, which has the following properties: $K(\cdot,t) \in H(K)$ and $\langle f, K(\cdot,t)\rangle = f(t)$ for every $t \in T$. We say that $\{f(t), t \in T\}$ is a representation of the process $\{X(t), t \in T\}$. Before finding the r.k. for the space of functions we are interested in, we present an example that will guide our intuition through our search process.

**Example** (Integrated Brownian Motion). Denote by $H$ the collection of functions $f$ with $f'' \in L^2[0,1]$ and consider the subspace $W_2 = \{f(x) \in H : f, f'$ absolutely continuous and $f(0) = f'(0) = 0\}$. Define the inner product on $H$ as

$$\langle f, g\rangle = \int_0^1 f''(t)g''(t)dt \tag{3.16}$$

It can be shown, see [19], that

$$R(s,t) = \frac{\max(s,t)\min^2(s,t)}{2} - \frac{\min^3(s,t)}{6} \tag{3.17}$$

Now, we also present a stochastic process with $K(s,t) = R(s,t)$. Let $\{X(t), t \in [0,1]\}$ be the Wiener process. Define a new stochastic process $\{Z(t), t \in [0,1]\}$ by

$$Z(t) = \int_0^t X(s)ds \tag{3.18}$$

We call $\{Z(t), t \in [0,1]\}$ the integrated Wiener process or integrated Brownian process. It can be shown, see [21], that $E[Z(t)] = 0$ and

$$E[Z(s)Z(t)] = \frac{\max(s,t)\min^2(s,t)}{2} - \frac{\min^3(s,t)}{6}. \tag{3.19}$$

Thus $W_2$ is a representation of $\{X(t), t \in [0,1]\}$.

Using intuition gained from this example, we will find the reproducing kernels for the $S_j^*$. The steps we take to find the r.k. are the following:

(i). Use intuition to guess at the G.P. $\{X(t), t \in [0,1]\}$ that corresponds to the RKHS $H$ for which we want to know the r.k. $R$.
(ii). Find the covariance function $K$ for $X(t)$.
(iii). Demonstrate that $R = K$ is such that $R(\cdot, t) \in H$ and $\langle R(\cdot, t), f\rangle = f(t)$ for $f \in H$, so that $R$ is the unique r.k. for $H$.

For ease of presentation, we first consider the simplest case: two subintervals. Let $\tau_1 \in [0,1]$, given $\tau_1$ we break down the $[0,1]$ interval into two subintervals. The basic idea is to express our function $f(x)$ as

$$f(x) = \alpha + \beta x + f_1(x) + f_2(x) \tag{3.20}$$

where $f_1 \in S_1^*$ and $f_2 \in S_2^*$. Note that (3.20) expresses $f(x)$ as a function in the space $F = \{1\} \oplus \{x\} \oplus S_1^* \oplus S_2^*$. To apply the RKHS framework and computational solution of ACOSSO to this problem we need to define RKHS's $H_1 \subset S_1^*$ and $H_2 \subset S_2^*$ and find the corresponding r.k.'s $R_1$ and $R_2$, respectively. It is not necessary to define a RKHS for the constant or linear term since they lie in the null space of the penalty in (3.9).

We define $H_1 = S_1^*$ with inner product $\langle f_1, g_1\rangle_{H_1} = \int_0^1 f_1''(t)g_1''(t)dt$. Similarly $H_2 = S_2^*$ with inner product $\langle f_2, g_2\rangle_{H_2} = \int_0^1 f_2''(t)g_2''(t)dt$. As previously mentioned, the locally adaptive estimator in (3.9) now becomes a special case of ACOSSO in (3.6).

First, we find the r.k. for $H_1$. Note that, by the definition of $H_1$, $f_1 \in H_1$ implies $f_1$ has curvature only in $[0, \tau_1]$, $f_1 \in S^2$ (where $S^2$ represents 2nd order Sobolev Space) and the inner product for $H_1$ is the same as that in the previous example involving $W_2$ and integrated Brownian motion.

In an effort to find the GP representation of $H_1$ we construct a Gaussian process as follows

$$Z_1(t) = \begin{cases} \int_0^t X(s)ds & 0 \leq t \leq \tau_1 \\ \int_0^t X(s)ds + X(\tau_1)(t - \tau_1) & \tau_1 \leq t \leq 1 \end{cases}$$

where $X(t)$ is a Wiener Process or Brownian motion.

The covariance function for $Z_1$, $K_1$, is a function whose domain is $\Re \bigotimes \Re$. Note that given $\tau_1$, a couple $(s, t)$ can fall into one of four regions: (i) $s \in [0, \tau_1]$ and $t \in [0, \tau_1]$, (ii) $s \in (\tau_1, 1]$ and $t \in (\tau_1, 1]$, (iii) $s \in [0, \tau_1]$ and $t \in (\tau_1, 1]$ and (iv) $t \in [0, \tau_1]$ and $s \in (\tau_1, 1]$.

We define $K_1(s, t)$ for each of these cases. However, knowing that $K_1(s, t)$ must be a symmetric function cases (iii) and (iv) are the same so we end up with only three cases. The calculations for each case are carried out in the appendix, but the results are given here for convenience. $K_1(s, t)$ is defined as follows

$$= \begin{cases} \frac{\max(s,t)\min^2(s,t)}{2} - \frac{\min^3(s,t)}{6} & \text{for } s, t \in [0, \tau_1] \\ \frac{\tau_1^3}{3} + \frac{(\max(s,t)-\tau_1)\tau_1^2}{2} + \frac{(\min(s,t)-\tau_1)\tau_1^2}{2} + \frac{2[\min(s,t)-\tau_1][\max(s,t)-\tau_1]\tau_1}{2} & \text{for } s, t \in (\tau_1, 1] \\ \frac{\max(s,t)\min^2(s,t)}{2} - \frac{\min^3(s,t)}{6} & \text{otherwise} \end{cases} \tag{3.21}$$

We demonstrate, in the appendix, that $K_1(s, t) = R_1(s, t)$ has the reproducing property, and hence, is the r.k. for $H_1$. Note that $K_1(s, t)$ also depends on $\tau_1$. For ease of notation, we make this dependence explicit by writing $K^*(s, t, \tau_1) = K_1(s, t)$.

Now, by the definition of $S_2^*$, the functions in $S_2^*$ are functions equal to zero in $[0, \tau_1]$ and with curvature in $(\tau_1, 1]$. Parallel to that above, we define the stochastic process $Z_2(t)$ as follows

$$Z_2(t) = \begin{cases} 0 & 0 \leq t \leq \tau_1 \\ \int_{\tau_1}^t X(s - \tau_1)ds & \tau_1 \leq t \leq 1 \end{cases}$$

where $X(t)$ is a Wiener Process or Brownian motion. From the above definition, it should be clear that $Z_2(t)$ is a shifted version of $Z_1(t)$, taking on a value of exactly 0 in the region where $Z_1$ is nonlinear, and providing nonlinearity in the region where $Z_1$ is linear.

The covariance function of $Z_2$ is

$$K_2(s,t) = \begin{cases} K^*(\frac{s-\tau_1}{1-\tau_1}, \frac{t-\tau_1}{1-\tau_1}, \frac{\tau_2-\tau_1}{1-\tau_1}) & \text{for } s,t \in (\tau_1, 1] \\ 0 & \text{otherwise} \end{cases} \tag{3.22}$$

Again, it is demonstrated in the appendix that $K_2(s,t) = R_2(s,t)$ has the reproducing property, and hence, is the r.k. for $H_2$.

One can do something analogous to derive the reproducing kernels in the case of multiple knots $0 = \tau_0 < \tau_1 < \cdots < \tau_{p-1} < \tau_p = 1$. In the general case that $H_j = S_j^*$ with inner product $\langle f_j, g_j \rangle_{H_j} = \int_0^1 f_j''(t) g_j''(t)dt$, the r.k. for $H_j$ is

$$K_j(s,t) = \begin{cases} K^*(\frac{s-\tau_{j-1}}{1-\tau_{j-1}}, \frac{t-\tau_{j-1}}{1-\tau_{j-1}}, \frac{\tau_j-\tau_{j-1}}{1-\tau_{j-1}}) & \text{for } s,t \in (\tau_{j-1}, \tau_j] \\ K^*(\frac{s-\tau_j}{1-\tau_j}, \frac{t-\tau_j}{1-\tau_j}, \frac{1-\tau_{j-1}}{1-\tau_{j-1}}) & \text{for } s,t \in (\tau_j, 1] \\ 0 & \text{otherwise} \end{cases} \tag{3.23}$$

## 3.6 Asymptotic properties of LACOSSO.

Let the $L_2$ norm of a function evaluated at the data points be denoted

$$\|f\|_n^2 = \frac{1}{n} \sum_{i=1}^{n} f^2(x_i)$$

The following theorem states that LACOSSO attains the optimal convergence rate for nonparametric regression estimators. The proof is deferred to the appendix.

**Theorem**. Consider the regression model $y_i = f_0(x_i) + \epsilon_i$, $i = 1, 2, ..., n$, where $x_i's$ are given values of a covariate in $[0,1]$, and $\epsilon_i's$ are independent $N(0, \sigma^2)$ errors. Assume $f_0$ lies in $S^2$ with $S^2$ being the second order Sobolev space.

Let $\hat{f}$ be defined as in (3.8) with $w_j = 1$ for all $j$ and let $I(f) = \int_0^1 [f''(x)]^2 dx$. Then (i) if $f_0$ is a nonlinear function, and $\lambda_n^{-1} = O_p(n^{2/5}) I^{3/10}(f_0)$, we have $\|\hat{f} - f_0\|_n = O_p(\lambda_n) I^{1/2}(f_0)$; (ii) if $f_0$ is a linear function, we have $\|\hat{f} - f_0\|_n = O_p(\max(n\lambda_n)^{-2/3}, n^{-1/2})$.

*Remark 1.* if $\lambda_n \sim n^{-2/5}$ then $\|\hat{f} - f_0\|_n = O_p(n^{-2/5})$ which is optimal for nonparametric regression estimators.

*Remark 2.* Here we have assumed $w_j = 1$, but this could be relaxed. All we really need is for $w_j = O_p(1)$ and $w_j^{-1} = O_p(1)$ in order for the proof to go through.

## 3.7 Example Results

In this section we evaluate the performance of LACOSSO on several simulated data sets. We compare the results to those from several other competing methods. The methods included in these simulations are:

LOCO - The Loco-Spline procedure with tuning parameter selection via 5-fold CV as described in [28].

SAS(5)- the version of the spatially adaptive smoothing spline suggested in [22] which uses piecewise constant (with 5 bins since this had the best performance in their paper) for $\lambda(x)$.

TRAD - the traditional smoothing spline (TRAD) with tuning parameter chosen via GCV.

LOKERN - local kernel regression with plug-in local bandwidth as provided by the R package lokern. This procedure uses a second order kernel with a plug-in estimate of the asymptotically optimal local bandwidth.

MARS - Multivariate Adaptive Regression Splines [7] as provided by the R package polymars. This procedure uses regression splines with spatially adaptive knot placement.

LACOSSO(0,5) - Locally Adaptive COSSO procedure with tuning parameter selection via GCV with $\gamma = 0$ and $p = 5$ bins ($\tau_j' s$ placed at evenly spaced quantiles of $x$).

LACOSSO(1,5) - Locally Adaptive COSSO procedure with tuning parameter selection via GCV with $\gamma = 1$ and $p = 5$ bins.

LACOSSO(0,10) - Locally Adaptive COSSO procedure with tuning parameter selection via GCV with $\gamma = 0$ and $p = 10$ bins.

LACOSSO(1,10) - Locally Adaptive COSSO procedure with tuning parameter selection via GCV with $\gamma = 1$ and $p = 10$ bins.

LACOSSO(0,20) - Locally Adaptive COSSO procedure with tuning parameter selection via GCV with $\gamma = 0$ and $p = 20$ bins.

LACOSSO(1,20) - Locally Adaptive COSSO procedure with tuning parameter selection via GCV with $\gamma = 1$ and $p = 20$ bins.

### 3.7.1 Mexican Hat Function

The first test problem which we call the Mexican hat function is a quadratic function with a sharp Gaussian bump in the middle of the domain. Specifically the function is given by

$$f(x) = -1 + 1.5x + 0.02\phi_{0.02}(x - 0.5) \tag{3.24}$$

where $\phi_\sigma(x-\mu)$ is the $N(\mu, \sigma^2)$ density evaluated at $x$. We generate a simple random sample of size $n$ from $x_i \sim \text{Unif}(0,1)$, $i = 1, 2, ..., n$. We then generate $Y_i = f(x_i)+\epsilon_i$, where $\epsilon_i \sim N(0, 0.25)$.

Figure 1 displays the data along with the corresponding fits from LACOSSO and traditional smoothing spline for a typical realization with $n = 100$. Here we see that LACOSSO-spline is able to both better capture the peak and stay smooth where the function is flat. On the other hand, see how the traditional smoothing spline "chases" data points in areas where the true function is flat.

In the top of Table 1 we can compare the performance on the Mexican hat example for these methods as sample size increases. The reported summary statistics are the average mean squared error (AMSE) and the percent best. The AMSE is the average of the MSE over 100 realizations at the respective sample sizes. Here we are using the definition of MSE which averages squared errors at the data points, i.e. $MSE = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - \hat{f}(x_i))^2$. The percent best is the percentage of the 100 realizations that a given method had the smallest MSE among the competing methods.

In the Mexican hat section of the table it is clear that LACOSSO has a very competitive performance for all sample sizes on this example. LACOSSO (1,20) had the smallest MSE for approximately 40% of the realizations for each sample size.

### 3.7.2 Dampened Harmonic Motion

The next problem is a dampened harmonic motion also known as the spring equation. Functions with this type of behavior are common to just about any structural engineering problem. The spring equation is given by

$$f(x) = a \ \exp\{-b(1-x)\} \cos\{w(1-x)\} \tag{3.25}$$

We have chosen the parameter values of $a = 1, b = 30, w = 30\pi$ to produce the data for this simulation. We again consider $x_i \sim$ Unif $(0, 1)$, $i = 1, 2, ..., n$ with

$Y_i = f(x_i) + \epsilon_i$, but here $\epsilon \sim N(0, 0.05)$.

Figure 2 displays the data and the corresponding fits from LACOSSO and traditional smoothing spline for a typical realization with $n = 100$. Here we see that the LACOSSO-spline captures better the behavior of this function. Note how the traditional smoothing estimate does not capture the higher amplitude oscillation as well as LACOSSO does and, again, allows for the undesirable behavior of "chasing" points in areas where the true function is flat.

The second tier of Table 1 summarizes the performance on the dampened harmonic example for sample sizes $n = 100, 200,$ and 300. In this example, our method has a performance as good as the one shown by LOCO and SAS(5). However, LACOSSO (1,10) has smaller MSE in, roughly, 30% of the realizations for all sample sizes, almost twice as much as SAS(5), its closest competitor.

### 3.7.3 Rapid Change Function

The rapid change function is defined as

$$f(x) = \frac{0.8}{1 + \exp[-75(x - 0.8)]} \tag{3.26}$$

We once again consider $x_i \sim \text{Unif}\,(0, 1)$ with $Y_i = f(x_i) + \epsilon_i$ and $\epsilon_i \sim N(0, 0.05)$.

Figure 3 displays the data and the corresponding fits from the traditional smoothing spline for a typical realization with $n = 100$. Notice how rough the smoothing spline is overall whereas the LACOSSO-spline is able to fit the true function just as well in the rapid change region as in the other regions.

Tier 3 of Table 1 summarizes the results of the simulations from this example. In this example, LACOSSO (1, 10) and (1, 20) have a lower AMSE than the other methods at all sample sizes. The only method that compares to these two is LOCO-Spline.

### 3.7.4 Motorcycle Crash Dataset

Here we take a look at a real data set that benefits from our local approach to smoothing. This data comes from a computer simulation of motorcycle accidents. The response is a series of measurements of head acceleration over time in a simulated motorcycle accident used to test crash helmets. This benchmark example data set made popular by ([25]) is available as *mcycle* in the R library MASS.

Figure 4 shows the estimated curves from LACOSSO and TRAD respectively. Notice how the LACOSSO estimate appears to have better agreement with the data, in general, especially in the last part of the domain (time $> 0.6$). See how the traditional smoothing spline estimate bounces around some in this region while LACOSSO remains very smooth which seems to give a much more visually appealing fit to the data.

## 3.8 Conclusions

Our new estimator, LACOSSO, is obtained via solving a regularization problem with a novel adaptive penalty on the sum of functional norms which allows for a locally varying smoothness of the resulting estimate. We demonstrated the effectiveness of this approach as a scatterplot smoother when compared to the traditional smoothing spline. LACOSSO machinery can be effectively transferred into higher dimensional problems and non-continuous responses (Bernoulli data, Poisson data, etc.) and its performance is not heavily affected when allowed for more flexibility (more bins) unlike other methods that have a tendency to overfit. In fact the performance of LACOSSO seems to improve with the addition of bins, which is a contrast to the method of [22], for example, with 10 and 20 bins. We attribute this behavior to the formulation of the minimization in the COSSO like framework, which involves only one tuning parameter, instead of one tuning parameter per bin. The MSE asymptotic optimality of this method has also been established.

# Chapter 4

# Conclusions and Future Work.

## 4.1 Conclusions

We have given several examples motivating the utility of the RKHS approach to penalized regression problems. We reviewed the building blocks necessary to define an RKHS and presented several key results about these spaces. We also used these results to perform illustrative estimation for ridge regression and the cubic smoothing spline problems, and presented transparent R code to enhance understanding of the examples.

We have introduced the LACOSSO, a new method for scatterplot smoothing which allows for locally varying smoothness of the resulting estimate. We have demonstrated the MSE asymptotic optimality of this method, and we have also shown the benefit that the local flexibility on smoothness can provide on many simulated examples. The new method compares favorably to existing methods for both speed and estimation accuracy as determined by extensive empirical testing.

## 4.2 Future Work.

**Multiple Predictor Case.**

Before we generalize the ideas we presented in the previous chapter, we recall what we did in the univariate case. Our function estimate is given by the minimizer of

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \sum_{j=1}^{p} w_j \left[\int_{\tau_{j-1}}^{\tau_j} [f''(x)]^2 dx\right]^{1/2} \tag{4.1}$$

The $w_j^{\prime}s$ will help us to penalize the different regions of the domain differently. Then we define $S_j^*$, where $S_j^* = \{f : f \text{ and } f' \text{ a.c. with} f'' \in L_2 \text{ for } x \in [\tau_{j-1}, \tau_j], f(x) = 0 \text{ if } x \in [0, \tau - j - 1),\ f \text{ is linear for } x \in (\tau_j, 1]\}$, $j = 1, ..., p$, for knots $0 = \tau_0 < \tau_1 < ... < \tau_p = 1$. These functional spaces allow us to decompose $f(x)$ as

$$f(x) = f_1(x) + f_2(x) + ... + f_p(x) \tag{4.2}$$

with $f_j \in S_j^*$. Expressing $f(x)$ in this way and using theory from RKHS, the ACOSSO framework and the Smothing Spline ANOVA, we can find the minimizer of (4.1) by solving an equivalent problem that is easier to solve, namely

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \sum_{j=1}^{p} w_j \left[\int_{\tau_{j-1}}^{\tau_j} [f_j''(x)]^2 dx\right]^{1/2} \tag{4.3}$$

A natural extension of our method is considering functions with multiple predictors. To reduce the level of abstraction, we discuss the two-predictor case. Suppose that two explanatory variables, $x_1$ and $x_2$, are available for a response variable, $y$, and that a simple additive model structure

$$y_i = f_1(x_{1i}) + f_2(x_{2i}) + \epsilon_i \tag{4.4}$$

is appropriate. The $f_j$ are smooth functions and the $\epsilon_i$ are iid $N(0, \sigma^2)$. Again, for simplicity, assume that $x_1$ and $x_2$ lie in $[0, 1]$.

The fact that the model now contains more than one function introduces an identifiability problem: $f_1$ and $f_2$ are each only estimable to within an additive constant. To see this, note that any constant could be simultaneously added to $f_1$ and subtracted from $f_2$, without changing the model predictions. Hence identifiability constraints have to be imposed on the model before fitting. One of the simplest ways to deal with this is to constrain one of the intercepts to zero.

Now let's try to generalize LACOSSO to estimate (4.4). Doing something similar to what we did in the univariate case, we define our estimate as the minimizer of

$$\frac{1}{n}\sum_{i=1}^{n}(y_i-f(\boldsymbol{x}))^2+\lambda\left[\sum_{j=1}^{p}w_{j1}\int_{\tau_{j-1}}^{\tau_j}(\frac{\partial^2 f}{\partial^2 x_1})dx_1^2+\sum_{j=1}^{p}w_{j2}\int_{\tau_{j-1}}^{\tau_j}(\frac{\partial^2 f}{\partial^2 x_2})^2dx_2\right]^{1/2} \tag{4.5}$$

which in this case is equivalent to

$$\frac{1}{n}\sum_{i=1}^{n}(y_i-f(\boldsymbol{x}))^2+\lambda\left[\sum_{j=1}^{p}w_{j1}\int_{\tau_{j-1}}^{\tau_j}(f_1''(x_1))^2dx_1+\sum_{j=1}^{p}w_{j2}\int_{\tau_{j-1}}^{\tau_j}(f_2''(x_2))^2dx_2\right]^{1/2} \tag{4.6}$$

Note that we have a common smoothing parameter $\lambda$ for both predictors but the $w_j's$ will provide us with the flexibility we need to estimate our function.

And parallel to what we did in the univariate case, decomposing $f_1(x_1)=\sum_{j=1}^{p}f_{j1}(x_1)$ and $f_2=\sum_{j=1}^{p}f_{j2}(x_2)$ with $f_{jk}\in S_j^*$, the solution to (4.5) is equivalent to find the minimizer of

$$\frac{1}{n}\sum_{i=1}^{n}(y_i-f(\boldsymbol{x}))^2+\lambda\left[\sum_{j=1}^{p}w_{j1}\int_{\tau_{j-1}}^{\tau_j}(f_{j1}''(x_1))^2dx_1+\sum_{j=1}^{p}w_{j2}\int_{\tau_{j-1}}^{\tau_j}(f_{j2}''(x_2))^2dx_2\right]^{1/2} \tag{4.7}$$

From the last equation it should be clear that the additive model can be represented and estimated in the same way as for the univariate model. The asymptotic properties of the estimator for the multiple predictor model are the same as those for the univariate model. The proof would be almost identical.

### Non-Gaussian Responses.

To have a better understanding of the main issues when extending penalized regression methods to non-normal responses we will revisit the cubic smoothing spline. Consider a "regression" problem $y_i = \eta + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$ and $\eta$ is an unknown constant. A sensible approach to this problem would be to find $\eta$, unknown "regression" function, using maximum likelihood. In this case, it is easy to see that the log-likelihood that we want to maximize is proportional to

$$-\sum_{i=1}^{n}(y_i - \eta)^2. \tag{4.8}$$

Maximizing this last expression would be equivalent to minimizing the negative log-likelihood

$$\sum_{i=1}^{n}(y_i - \eta)^2 \tag{4.9}$$

Now, consider another regression problem $y_i = \eta(x_i) + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$ and $\eta(x)$ is an unknown function. If we attempt to maximize the log-likelihood

$$-\sum_{i=1}^{n}(y_i - \eta(x_i))^2 \tag{4.10}$$

over all smooth functions $\eta$, the result is useless. It is always possible to choose $\eta$ sufficiently complicated that it interpolates the data. Hence, instead of maximizing the log-likelihood alone, we choose $\hat{\eta}$ in second order Sobolev space to maximize the penalized log-likelihood

$$-\sum_{i=1}^{n}(y_i - \eta(x_i))^2 - \frac{\lambda}{2}\int_0^1 (\eta''(x))^2 \tag{4.11}$$

which is equivalent to minimizing the negative log-likelihood

$$\sum_{i=1}^{n}(y_i - \eta(x_i))^2 + \frac{\lambda}{2}\int_0^1 (\eta''(x))^2 \tag{4.12}$$

where, as we already know, the first term discourages the lack of fit of $\eta$ to the data, the second penalizes the roughness of $\eta$, and the smoothing parameter $\lambda$ controls the trade-off between the two conflicting goals. We also know, from chapter 2, that the minimizer $\eta_\lambda$ is called a cubic smoothing spline.

Now consider exponential family distributions with densities of the form

$$f(y|x) = \exp\left([y\eta(x) - b(\eta(x))]/a(\phi) + c(y, \phi)\right) \tag{4.13}$$

where $a > 0$, $b$, and $c$ are known functions, $\eta(x)$ is the parameter of interest dependent on a covariate $x$ and $\phi$ is either known or considered as a nuisance parameter that is independent of $x$. Observing $y_i|x_i \sim f(y|x_i)$, $i = 1, 2, ..., n$, one is to estimate the regression function $\eta(x)$. Parallel to what happened in the normal case, we could minimize the negative of the penalized log-likelihood functional

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i\eta(x_i) - b(\eta(x_i))\right)^2 + \frac{\lambda}{2}J(\eta) \tag{4.14}$$

for some penalty $J(\eta)$ such as a penalty on smoothness as in (4.11). It can be shown (see [10]) that the minimizer $\eta_\lambda$ of (4.14) takes the form

$$\eta(x) = \sum_{\nu=1}^{m} d_\nu \phi_\nu(x) + \sum_{i=1}^{n} c_i R(x_i, x) \tag{4.15}$$

where $\{\phi_\nu\}_{\nu=1}^m$ is a basis of the null space of the penalty $J(\eta)$ and $R(x_i, x)$ is an appropriate reproducing kernel. With a non-normal log likelihood, one needs iterations to compute $\eta_\lambda$, even for fixed smoothing parameters, which adds to the complexity of the problem.

The implementation of efficient and effective algorithms to locate good estimates from among the $\eta_\lambda$'s with varying smoothing parameters would allow us to fit models with non-normal responses using our method. Such algorithms do exist, see chapter five of [10].

**Variable Selection.**

Variable selection for multivariate nonparametric regression is a challenging problem due to the infinite dimensionality of the function space. We say a nonparametric regression estimator has the nonparametric oracle property if it selects the correct subset of predictors with probability tending to one and estimates the regression surface $f$ at the optimal nonparametric rate; see ([28]). It would be interesting to investigate if our method, LACOSSO, in the additive model case could be a nonparametric oracle and, if so, under which conditions.

# Appendices

# Appendix A

## A.1 Appendix

### A.1.1 R Codes

**Ridge Regression.** We consider a sample of size $n = 20$, $(y_1, y_2, y_3, ..., y_{20})$, from the model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i$$

where $\beta_0 = 2$, $\beta_1 = 3$, $\beta_2 = 0$ and $\epsilon_i$ has a $N(0, 0.25^2)$ . The distribution of the covariates, $x_1$ and $x_2$, is uniform in $[0, 1]^2$ so that $x_2$ is uninformative. The following lines of code generate the data.

```
###### Data  ########

set.seed(3)

n<-20

x1<-runif(n)

x2<-runif(n)

X<-matrix(c(x1,x2),ncol=2)   # design matrix

y<-2+3*x1+rnorm(n,sd=0.25)
```

*Appendix A.*

Below we give a function “ridge regression” to solve the ridge regression problem using the RKHS framework we discussed in Section 4.2.

```
##### function to find the inverse of a matrix ####

my.inv<-function(X,eps=1e-12){ eig.X<-eigen(X,symmetric=T)

P<-eig.X[[2]] lambda<-eig.X[[1]] ind<-lambda>eps

lambda[ind]<-1/lambda[ind] lambda[!ind]<-0

ans<-P%*%diag(lambda,nrow=length(lambda))%*%t(P)

return(ans) }


###### Reproducing Kernel #########

rk<-function(s,t){

    p<-length(s)

    rk<-0

    for (i in 1:p){

        rk<-s[i]*t[i]+rk

                    }

    return( (rk) )

} ##### Gram matrix  ####### get.gramm<-function(X){ n<-dim(X)[1]

Gramm<-matrix(0,n,n) #initializes Gramm array #i=index for rows

#j=index for columns Gramm<-as.matrix(Gramm)   # Gramm matrix

for(i in 1:n){

      for (j in 1:n){

    Gramm[i,j]<-rk(X[i,],X[j,])

         }
} return(Gramm) }


ridge.regression<-function(X,y,lambda){
    Gramm<-get.gramm(X)   #Gramm matrix (nxn)
    n<-dim(X)[1]          # n=length of y
    J<-matrix(1,n,1)      # vector of ones dim=n
    Q<-cbind(J,Gramm)     # design matrix
    m<-1                  # dimension of the null space
                          # of the penalty
    S<-matrix(0,n+m,n+m)  #initialize S
    S[(m+1):(n+m),(m+1):(n+m)]<-Gramm #non-zero part of S
    M<-(t(Q)%*%Q+lambda*S)
    M.inv<-my.inv(M)       # inverse of M
    gamma.hat<-crossprod(M.inv,crossprod(Q,y))
    f.hat<-Q%*%gamma.hat
    A<-Q%*%M.inv%*%t(Q)
    tr.A<-sum(diag(A))              #trace of hat matrix
    rss<-t(y-f.hat)%*%(y-f.hat)     #residual sum of squares
    gcv<-n*rss/(n-tr.A)^2           #obtain GCV score
    return(list(f.hat=f.hat,gamma.hat=gamma.hat,gcv=gcv))
}
```

A simple direct search for the GCV optimal smoothing parameter can be made as follows:

```
# Plot of GCV
```

```
lambda<-1e-8 V<-rep(0,40) for (i in 1:40){

    V[i]<-ridge.regression(X,y,lambda)$gcv   #obtain GCV score

    lambda<-lambda*1.5         #increase lambda

} index<-(1:40) plot(1.5^(index-1)*1e-8,V,type="l",main="GCV

score",lwd=2,xlab="lambda",ylab="GCV")  # plot score
```

The GCV plot produced by this code is displayed in Figure 2.

Now, by following Section 4.2, $\hat{\beta}_0, \hat{\beta}_1$ and $\hat{\beta}_2$ can be obtained as follows.

```
i<-(1:60)[V==min(V)]                    # extract index of min(V)

opt.mod<-ridge.regression(X,y,1.5^(i-1)*1e-8)  #fit optimal model

### finding beta.0, beta.1 and beta.2  ##########

gamma.hat<-opt.mod$gamma.hat

beta.hat.0<-opt.mod$gamma.hat[1]#intercept

beta.hat<-gamma.hat[2:21, ]%*%X   #slope and noise term coefficients
```

The resulting estimates are: $\hat{\beta}_0 = 2.1253$, $\hat{\beta}_1 = 2.6566$ and $\hat{\beta}_2 = 0.1597$.

**Cubic Smoothing Spline.** We consider a sample of size $n = 50$, $(y_1, y_2, y_3, ..., y_{20})$, from the model

$$y_i = \sin(2\pi x_i) + \epsilon_i$$

where $\epsilon_i$ has a $N(0, 0.2^2)$ . The following code generates $x$ and $y$

```
###### Data  ########
```

```
set.seed(3)
n<-50
x<-matrix(runif(n),nrow=n,ncol=1)
x.star<-matrix(sort(x),nrow=n,ncol=1) # sorted x, used by plot
y<-sin(2*pi*x.star)+rnorm(n,sd=0.2)
```

Below we give a function to find the cubic smoothing spline using the RKHS framework we discussed in Section 4.3. We also provide a graph with our estimation along with the true function and data.

```
#### Reproducing Kernel for <f,g>=int_0^1 f''(x)g''(x)dx  #####
rk.1<-function(s,t){
    return( (1/2)*min(s,t)^2)*(max(s,t)+ (1/6)*(min(s,t))^3  )
}


get.gramm.1<-function(X){
    n<-dim(X)[1]
    Gramm<-matrix(0,n,n)  #initializes Gramm array
    #i=index for rows
    #j=index for columns
    Gramm<-as.matrix(Gramm)   # Gramm matrix
for (i in 1:n){
      for (j in 1:n){
    Gramm[i,j]<-rk.1(X[i,],X[j,])
            }
              }
return(Gramm) }


smoothing.spline<-function(X,y,lambda){
    Gramm<-get.gramm.1(X)   #Gramm matrix (nxn)
    n<-dim(X)[1]          # n=length of y
    J<-matrix(1,n,1)      # vector of ones dim=n
    T<-cbind(J,X)         # matrix with a basis for the null
                          # space of the penalty
    Q<-cbind(T,Gramm)     # design matrix
    m<-dim(T)[2]          # dimension of the null space of
                          # the penalty
    S<-matrix(0,n+m,n+m)  #initialize S
    S[(m+1):(n+m),(m+1):(n+m)]<-Gramm #non-zero part of S
    M<-(t(Q)%*%Q+lambda*S)
    M.inv<-my.inv(M)       # inverse of M
    gamma.hat<-crossprod(M.inv,crossprod(Q,y))
    f.hat<-Q%*%gamma.hat
    A<-Q%*%M.inv%*%t(Q)
    tr.A<-sum(diag(A))            #trace of hat matrix
    rss<-t(y-f.hat)%*%(y-f.hat)  #residual sum of squares
    gcv<-n*rss/(n-tr.A)^2        #obtain GCV score
    return(list(f.hat=f.hat,gamma.hat=gamma.hat,gcv=gcv))
}
```

A simple direct search for the GCV optimal smoothing parameter can be made as follows:

```
### Now we have to find an optimal lambda using GCV...


### Plot of GCV


lambda<-1e-8 V<-rep(0,60) for (i in 1:60){
    V[i]<-smoothing.spline(x.star,y,lambda)$gcv   #obtain GCV score
    lambda<-lambda*1.5         #increase lambda
} plot(1:60,V,type="l",main="GCV score",xlab="i")  # plot score
i<-(1:60)[V==min(V)]                   # extract index of min(V)
opt.mod.2<-smoothing.spline(x.star,y,1.5^(i-1)*1e-8)  #fit optimal
model


#Graph (Cubic Spline)
plot(x.star,opt.mod.2$f.hat,type="l",lty=2,lwd=2,col="blue",xlab="x",
ylim=c(-2.5,1.5),xlim=c(-0.1,1.1),ylab="response",main="CubicSpline")
#predictions
lines(x.star,sin(2*pi*x.star),lty=1,lwd=2) #true
legend(-0.1,-1.5,c("predictions","true"),lty=c(2,1),bty="n",
lwd=c(2,2),col=c("blue","black")) points(x.star,y) #data
```

This graph is plotted in Figure 3.

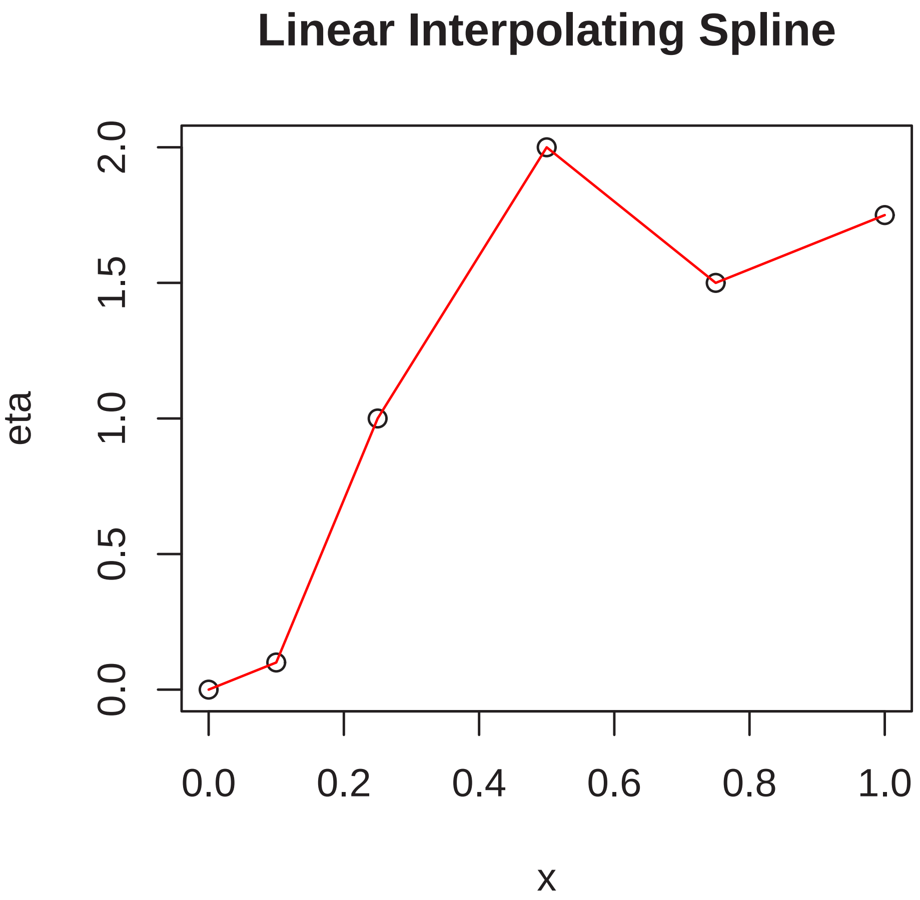

Figure A.1: Linear Interpolating Spline

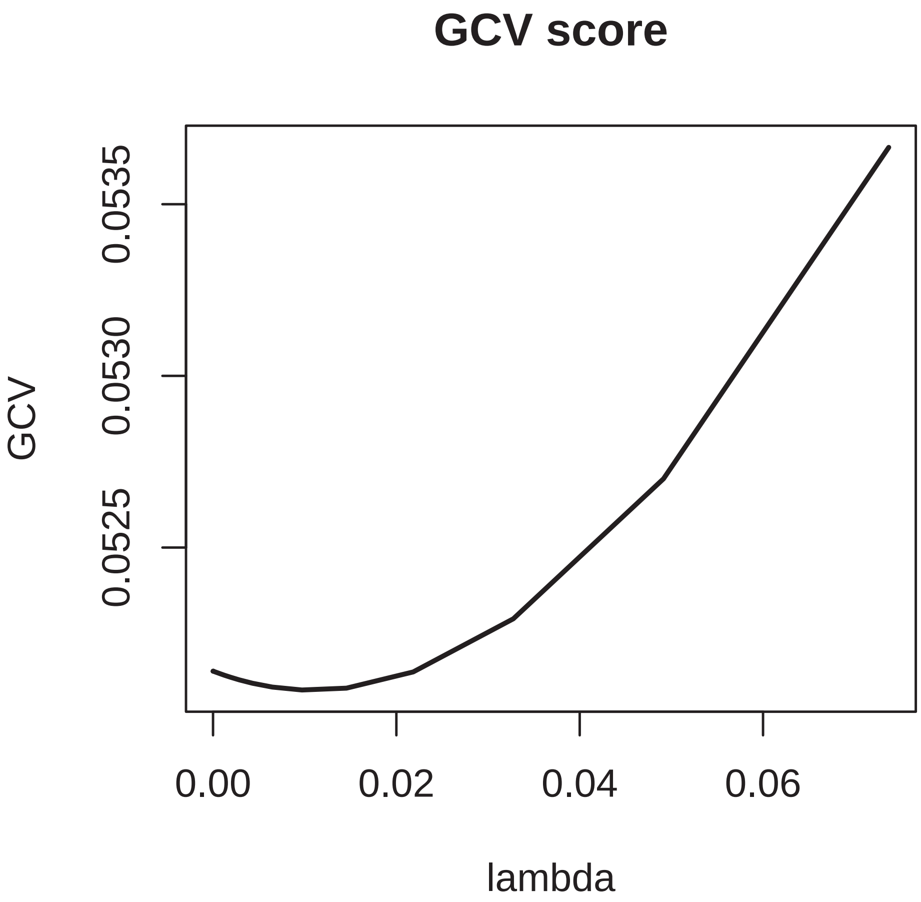

Figure A.2: GCV Score

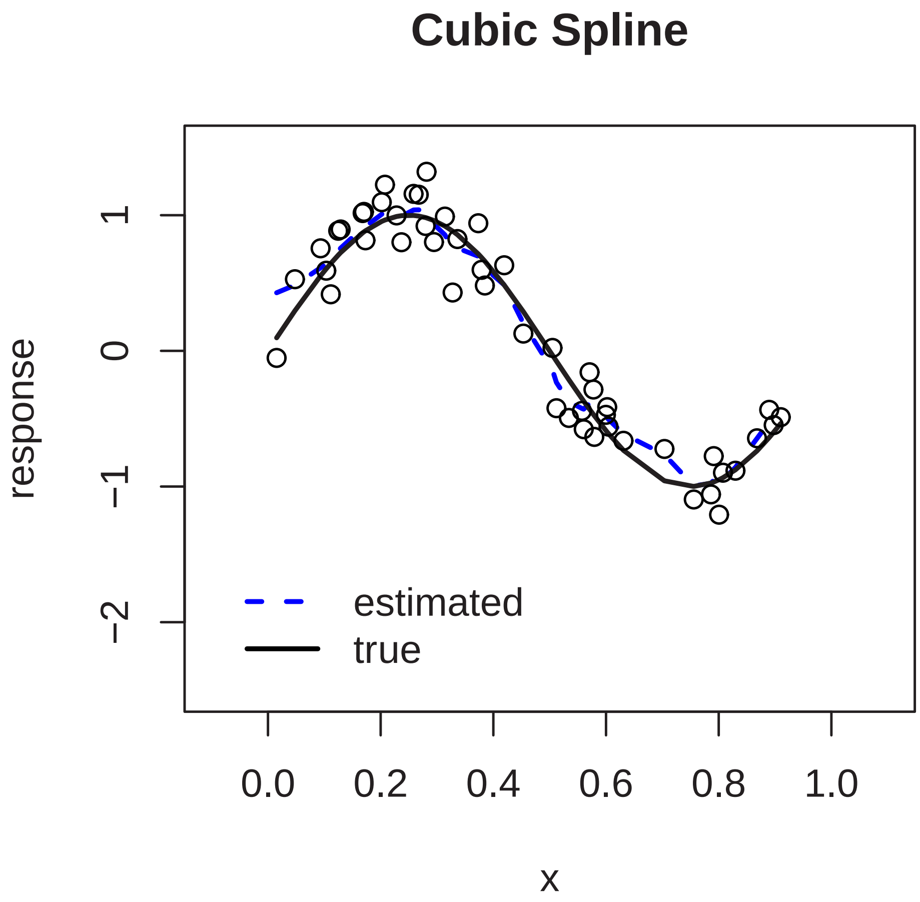

Figure A.3: Cubic Spline

# Appendix B

## B.1 Appendix

### B.1.1 Calculations to find the r.k.

Here we give the specific calculations of the covariance functions $K_1$ and $K_2$ for section 3.5.2. We then demonstrate that $K_1$ and $K_2$ are the r.k.'s for $S_1^*$ and $S_2^*$, respectively.

**Case 1.** $0 \leq s \leq t \leq \tau_1$

First, we need to recall that the mean value and covariance of a Wiener Process are given by

$$m(t) = E[X(t)] = 0 \tag{B.1}$$

$$Cov[X(s), X(t)] = \min(s, t) \tag{B.2}$$

Now, note that the expectation of $Z_1(t)$ is equal to zero.

$$E\left[\int_0^t X(\nu)d\nu\right] = \int_0^t E[X(\nu)]d\nu = 0 \tag{B.3}$$

This implies that the covariance between $Z_1(s)$ and $Z_1(t)$ is given by

$$\begin{aligned}
Cov[Z_1(s)Z_1(t)] &= E\left[Z_1(s)Z_1(t)\right] \\
&= E\left[\int_0^s X(y)dy \int_0^t X(u)du\right] \\
&= E\left[\int_0^s \int_0^t X(y)X(u)dydu\right] \\
&= \int_0^s \int_0^t E\left[X(y)X(u)\right] dydu \\
&= \int_0^s \int_0^t \min(y,u)dydu \\
&= \int_0^s \left(\int_0^u ydy + \int_u^t udy\right) du \\
&= s^2\left(\frac{t}{2} - \frac{s}{6}\right) = \frac{s^2t}{2} - \frac{s^3}{6}
\end{aligned}$$

or in general for $0 < s < \tau_1$, $0 < t < \tau_1$,

$$E\left[Z_1(s)Z_1(t)\right] = \frac{\min^2(s,t)\max(s,t)}{2} - \frac{\min^3(s,t)}{6} \tag{B.4}$$

**Case 2.** If $t$ and $s$ are in $[\tau_1, 1]$ and $s < t$.

First, note that $E(Z_1(t)) = 0$.

To determine the covariance between $Z_1(s)$ and $Z_1(t)$ we need to find its product

$$\begin{aligned}
Z_1(s)Z_1(t) &= [Z_1(\tau_1) + (s-\tau_1)X(\tau_1)][Z_1(\tau_1) + (t-\tau_1)X(\tau_1)] \\
&= Z_1^2(\tau_1) + Z_1(\tau_1)X(\tau_1)(t-\tau_1) + Z_1(\tau_1)X(\tau_1)(s-\tau_1) + (s-\tau_1)(t-\tau_1)X^2(\tau_1) \\
&= Z_1^2(\tau_1) + Z_1(\tau_1)X(\tau_1)[(t-\tau_1) + (s-\tau_1)] + (s-\tau_1)(t-\tau_1)X^2(\tau_1)
\end{aligned}$$

Now,

$$E[Z_1(s)Z_1(t)] = E[Z_1^2(\tau_1)] + [(t-\tau_1)+(s-\tau_1)]E[Z_1(\tau_1)X(\tau_1)] + (s-\tau_1)(t-\tau_1)E[X^2(\tau_1)] \tag{B.5}$$

We know, from case 1, that

$$E[Z_1^2(\tau_1)] = \frac{\tau_1^3}{2} - \frac{\tau_1^3}{6} = \frac{\tau_1^3}{3} \tag{B.6}$$

$$E[X(\tau_1)Z_1(\tau_1)] = E\left[\int_0^{\tau_1} X(y)X(\tau_1)dy\right] \tag{B.7}$$

$$= \int_0^{\tau_1} \min(y, \tau_1)dy = \frac{\tau_1^2}{2} \tag{B.8}$$

$$E[X^2(\tau_1)] = \min(\tau_1, \tau_1) = \tau_1 \tag{B.9}$$

Then, substituting (B.6), (B.8) and (B.9) into (B.5) gives

$$E[Z_1(s)Z_1(t)] = \frac{\tau_1^3}{3} + [(t-\tau_1) + (s-\tau_1)]\left[\frac{\tau_1^2}{2}\right] + (s-\tau_1)(t-\tau_1)\tau_1 \tag{B.10}$$

**Case 3.** $0 \le s \le \tau_1 \le t \le 1$.

From the two cases discussed above, we have that $E(Z_1(t)) = 0$. To find the covariance of $Z_1(t)$, first we need the product $Z_1(s)Z_1(t)$

$$\begin{aligned} Z_1(s)Z_1(t) &= \left[\int_0^s X(\nu)d\nu\right]\left[\int_0^{\tau_1} X(\nu)d\nu + (t-\tau_1)X(\tau_1)\right] \\ &= \left[\int_0^s X(\nu)d\nu\right]\left[\int_0^s X(\nu)d\nu + \int_s^{\tau_1} X(\nu)d\nu + (t-\tau_1)X(\tau_1)\right] \\ &= \left[\int_0^s X(\nu)d\nu\right]^2 + \left[\int_0^s X(\nu)d\nu\right]\left[\int_s^{\tau_1} X(\nu)d\nu\right] + (t-\tau_1)X(\tau_1)\left[\int_0^s X(\nu)d\nu\right] \\ &= [Z_1(s)]^2 + [Z_1(s)]\,[Z_1(\tau_1) - Z_1(s)] + (t-\tau_1)X(\tau_1)\,[Z_1(s)] \end{aligned}$$

Now,

$$E[Z_1(s)Z_1(t)] = E\,[Z_1(s)]^2 + E\,[Z_1(s)]\,[Z_1(\tau_1) - Z_1(s)] + (t-\tau_1)E\,[X(\tau_1)Z_1(s)] \tag{B.11}$$

From calculations in cases 1 and 2, we know that

$$E[Z_1^2(s)] = Var[Z_1(s)] = \frac{s^3}{3} \tag{B.12}$$

$$E[X(\tau_1)Z_1(s)] = \frac{s^2}{2} \tag{B.13}$$

We also need to find the following expectation

$$E\left[Z_1(s)\right]\left[Z_1(\tau_1) - Z_1(s)\right] = Cov\left[Z_1(s)Z_1(\tau_1)\right] - Var\left[Z_1(s)\right] \tag{B.14}$$

$$= \left[\frac{s^2\tau_1}{2} - \frac{s^3}{6} - \frac{s^3}{3}\right] = \left[\frac{s^2\tau_1 - s^3}{2}\right] \tag{B.15}$$

Finally, substituting (B.12),(B.13) and (B.15) into (B.11) we have that

$$E[Z_1(s)Z_1(t)] = \frac{s^3}{3} + \left[\frac{\tau_1 s^2 - s^3}{2}\right] + \frac{(t-\tau_1)s^2}{2} = \frac{ts^2}{2} - \frac{s^3}{6} \tag{B.16}$$

or for general case $0 \le s \le \tau_1 \le t \le 1$ or $0 \le t \le \tau_1 \le s \le 1$

$$E[Z_1(s)Z_1(t)] = \frac{\max(s,t)\min^2(s,t)}{2} - \frac{\min^3(s,t)}{6} \tag{B.17}$$

### B.1.2 Proving that $R_1 = K_1$ is the r.k. of $S_1^*$.

We now prove that $R_1^* = K_1$ has the "reproducing property" and hence is the r.k. of the space $S_1^*$. We split this into two cases: (1) $t \in [0, \tau_1]$ and (2) $t \in [\tau_1, 1]$.

**Case 1**. Assume that $t \in [0, \tau_1]$, by definition the inner product between $f(\cdot)$ and $K(\cdot, t)$ is given by

$$\langle f(t), K(\cdot, t)\rangle = \int_0^1 \frac{\partial^2}{\partial s^2} K(s,t) f''(s) ds$$

$$= \int_0^t \frac{\partial^2}{\partial s^2} K(s,t) f''(s) ds + \int_t^{\tau_1} \frac{\partial^2}{\partial s^2} K(s,t) f''(s) ds + \int_{\tau_1}^1 \frac{\partial^2}{\partial s^2} K(s,t) f''(s) ds$$

Note that the second term and the third term on the right hand side of the last equation are equal to zero. One can see this by using our definition of $K(s,t)$, case 1 in (B.4) and case 3 in (B.17) respectively, and finding the second partial derivative of $K(s,t)$ for each case with respect to $s$ when $s > t$.

Doing this we have that

$$\langle f(t), K(s,t)\rangle = \int_0^t \frac{\partial^2}{\partial s^2} K(s,t) f''(s) ds = \int_0^t (t-s) f''(s) ds. \tag{B.18}$$

Integrating by parts this last expression

$$\langle f(t), K(s,t)\rangle = (t-t)f'(t) - (t-0)f'(0) + f(t) - f(0). \tag{B.19}$$

Finally, recall that $f'(0) = f(0) = 0$ which implies that

$$\langle f(t), K(s,t)\rangle = f(t). \tag{B.20}$$

**Case 2**. Assume that $t \in [\tau_1, 1]$, by definition the inner product between $f(\cdot)$ and $K(\cdot, t)$ is given by

$$\langle f(t), K(s,t)\rangle = \int_0^1 \frac{\partial^2}{\partial s^2} K(s,t) f''(s) ds$$

$$= \int_0^{\tau_1} \frac{\partial^2}{\partial s^2} K(s,t) f''(s) ds + \int_{\tau_1}^t \frac{\partial^2}{\partial s^2} K(s,t) f''(s) ds + \int_t^1 \frac{\partial^2}{\partial s^2} K(s,t) f''(s) ds$$

Note that the second term and third terms on the right hand side of the equation shown above are equal to zero, by our definition of $K(s,t)$, so that

$$\langle f(t), K(s,t)\rangle = \int_0^{\tau_1} \frac{\partial^2}{\partial s^2} K(s,t) f''(s) ds = \int_0^{\tau_1} (t-s) f''(s) ds \tag{B.21}$$

$$= \int_0^{\tau_1} (t-s) f''(s) ds = (t-\tau_1) f'(\tau_1) - t f'(0) + f(\tau_1) - f(0) \tag{B.22}$$

Recalling that $f'(0) = f(0) = 0$ we have that

$$\langle f(t), K(s,t)\rangle = f(\tau_1) + (t-\tau_1) f'(\tau_1) = f(t) \tag{B.23}$$

Applying the same arguments to the shifted and rescaled versions of $s$ and $t$, one can prove that $R_2^* = K_2$ is the r.k. for $S_2^*$ as well.

### B.1.3 Proof of the Convergence Theorem.

Before presenting a proof for the result regarding optimal MSE convergence, we state a definition and a lemma necessary for the proof.

**Definition**. (Entropy for the supremum norm) For a function space $G$, let $N_\infty(\delta, G)$ be the smallest value of $N$ such that there exists $\{g_j\}_{j=1}^N$ with

$$\sup_{g \in G} \min_{j=1,..,N} |g - g_j|_\infty \leq \delta.$$

Then $H_\infty(\delta, G) = \log N_\infty(\delta, G)$ is called the $\delta$-entropy of $G$ for the supremum norm. Where $|g|_\infty = \sup_{x \in X} |g(x)|$.

**Lemma 1**. Consider a regression model $y_i = g_0(x_i) + \epsilon_i$, $i = 1, ..., n$ where $g_0$ is known to lie in a class $G$ of functions, $x_i$ are given covariates in $[0,1]^p$, and $\epsilon_i$ are independent $N(0, \sigma^2)$ errors. Let $I : G \to [0, \infty)$ be a pseudo-norm on $G$. Define

$$\hat{g} = \arg \min_{g \in G} \frac{1}{n} \sum_{i=1}^{n} \{y_i - g(x_i)\}^2 + \lambda_n^2 I(g)$$

Assume

$$H_\infty \left( \delta, \{ \frac{g - g_0}{I(g) + I(g_0)} : g \in G, I(g) + I(g_0) > 0 \} \right) \leq A\delta^{-\alpha} \tag{B.24}$$

for all $\delta > 0$, $n \geq 1$ and some $A > 0$, $0 < \alpha < 2$. Here $H_\infty$ stands for the entropy for the supreme norm. Then i) if $I(g_0) > 0$ and $\lambda_n^{-1} = O_p(n^{\frac{1}{2+\alpha}}) I^{\frac{2-\alpha}{4+2\alpha}}(g_0)$, we have $\|\hat{g} - g_0\|_n = O_p(\lambda_n) I^{1/2}(g_0)$; ii) if $I(g_0) = 0$ we have $\|\hat{g} - g_0\|_n = O_p(n^{\frac{-1}{2-\alpha}}) \lambda_n^{\frac{-2\alpha}{2-\alpha}}$

Before actually proving the theorem we give a brief sketch of the proof to provide more clarity. First, note that if we prove that $G$, the class of functions we are interested in, is bounded in entropy then we can use Lemma 1 and we are done. It turns out that with the supremum norm we have problems with the linear part of our functions. That is, Lemma 1 cannot be used directly since (B.24) is not satisfied in our case. To see this define the following set of functions

$$F = \{ f(x) = \alpha + \beta x, \ x \in [0,1], \ \alpha, \beta \in \Re \} \tag{B.25}$$

Note that given any $\delta > 0$ and any finite set of functions $g_j$'s we can find a function $f \in F$ such that $|f - g_j|_\infty > \delta$. Therefore, we decompose our functional space into two parts: linear part and non-linear part. Then we deal with each of these components separately. After we give a rate of convergence for the linear part, we deal with the non-linear part. We will show that the entropy of the functional space of the nonlinear part has the form $A^*\delta^{1/2}$, for some $A^*$ and the desired result follows from Lemma 1.

**Proof of the Theorem**. Lemma 1 cannot be used directly since (B.24) is not satisfied in our case. Therefore, to apply lemma 1 we have to decompose the space of functions in two parts: linear part and non-linear part. This problem can be dealt with with the following arguments. For any $f \in S^2$, we can write

$$f(x) = b_0 + b_1 x + f_1(x) + ... + f_p(x) = g_1(x) + g_2(x)$$

where $g_1(x) = b_0 + b_1 x$, $g_2(x) = f_1(x) + ... + f_p(x)$, $f_j \in S_j^*$, $\sum_{i=1}^n f_j(x_i) = 0$ and $\sum_{i=1}^n x_i f_j(x_i) = 0$ for $j = 1, 2, ..., p$.

Similarly, for the unknown underlying function $f_0$, write

$$f_0(x) = b_{00} + b_{01} x + f_{01}(x) + ... + f_{0p}(x) = g_{01}(x) + g_{02}(x)$$

where $g_{01}(x) = b_{00} + b_{01} x$, $g_{02}(x) = f_{01}(x) + ... + f_{0p}(x)$, $f_{0j} \in S_j^*$, $\sum_{i=1}^n f_{0j}(x_i) = 0$ and $\sum_{i=1}^n x_i f_{0j}(x_i) = 0$ for $j = 1, 2, ..., p$. Then, by construction $\sum_{i=1}^n \{g_{01}(x_i) - g_1(x_i)\}\{g_{02}(x_i) - g_2(x_i)\} = 0$.

Then we can write $\frac{1}{n}\sum_{i=1}^n [y_i - f(x_i)]^2 + \lambda_n J(f)$ , where $J(f) = \sum_{j=1}^p \left\{ \int_{\tau_{j-1}}^{\tau_j} [f''(x)]^2 dx \right\}^{1/2}$, as

$$\frac{1}{n}\sum_{i=1}^n \{(g_{01}(x_i) - g_1(x_i)) + (g_{02}(x_i) - g_2(x_i) + \epsilon_i)\}^2 + \lambda_n J(g)$$

$\frac{1}{n}\sum_{i=1}^n \{(g_{01}(x_i) - g_1(x_i))\}^2 + \frac{2}{n}\sum_{i=1}^n (g_{01}(x_i) - g_1(x_i))(g_{02}(x_i) - g_2(x_i) + \epsilon_i)$
$+\sum_{i=1}^n (g_{02}(x_i) - g_2(x_i) + \epsilon_i)^2 + \lambda_n J(g)$.

Due to the conditions imposed above (we have those conditions to guarantee that $g_1$ and $g_2$ are orthogonal under the empirical inner product), we have

$$\frac{1}{n}\sum_{i=1}^{n}\{(g_{01}(x_i)-g_1(x_i))\}^2+\frac{2}{n}\sum_{i=1}^{n}(g_{01}(x_i)-g_1(x_i))\epsilon_i+\sum_{i=1}^{n}(g_{02}(x_i)-g_2(x_i)+\epsilon_i)^2$$
$$+\ \lambda_n J(g_2)$$

Therefore the corresponding $g_1$ to the $f$ which minimizes (3.8) must minimize

$$\frac{1}{n}\sum_{i=1}^{n}\{(g_{01}(x_i)-g_1(x_i))\}^2+\frac{2}{n}\sum_{i=1}^{n}(g_{01}(x_i)-g_1(x_i))\epsilon_i.$$

By example 9.3.1 ([30] page 152), we have that $\hat{g}_1$ converges with rate $n^{-1/2}$.

On the other hand, the non-linear part, $\hat{g}_2$ must minimize

$$\frac{1}{n}\sum_{i=1}^{n}\left[g_{02}(x_i)-g_2(x_i)\right]^2+\lambda_n J(g_2)$$

Let $G=\{g\in S^2 : g(x)=f_1(x)+...+f_p(x)$ with $f_j\in S_j^*$, $\sum_{i=1}^{n}f_j(x_i)=0$, and $\sum_{i=1}^{n}x_i f_j(x_i)=0$, $j=1,2,...,p\}$.

We can now apply lemma 1 with $I=J$ and $\alpha=1/2$. All that remains to be shown is that (B.24) is satisfied. The conclusion of the Theorem then follows from the conclusion of lemma 1.

Let $J^*(g)=\int_0^1[f''(x)]^2dx$. From page 168 of [30], note that

$$H_\infty(\delta,\{g\in G : J^*(g)\le 1\})\le A\delta^{-1/2}.$$

Also,

$$J^*(g)=\int_0^1[f''(x)]^2dx\le\left(\sum_{j=1}^{p}\left\{\int_{\tau_{j-1}}^{\tau_j}[f''(x)]^2dx\right\}^{1/2}\right)^2=J^2(g)$$

Thus $J(g)\le 1$ implies that $J^*(g)\le 1$ so that $\{g\in G : J(g)\le 1\}\subset\{g\in G : J^*(g)\le 1\}$. Now if $\{g\in G : J^*(g)\le 1\}$ can be covered by $N$ balls of radius $\delta$, then $\{g\in G : J(g)\le 1\}$ can be covered by the same balls since it is a smaller set. Hence,

$$H_\infty(\delta,\{g\in G : J(g)\le 1\})\le A\delta^{-1/2}.$$

Lastly, noting that $J(g - g_0) \leq J(g) + J(g_0)$ for any $g \in G$, we see that (B.24) is satisfied. The conclusion of the Theorem then follows from the conclusion of lemma 1.

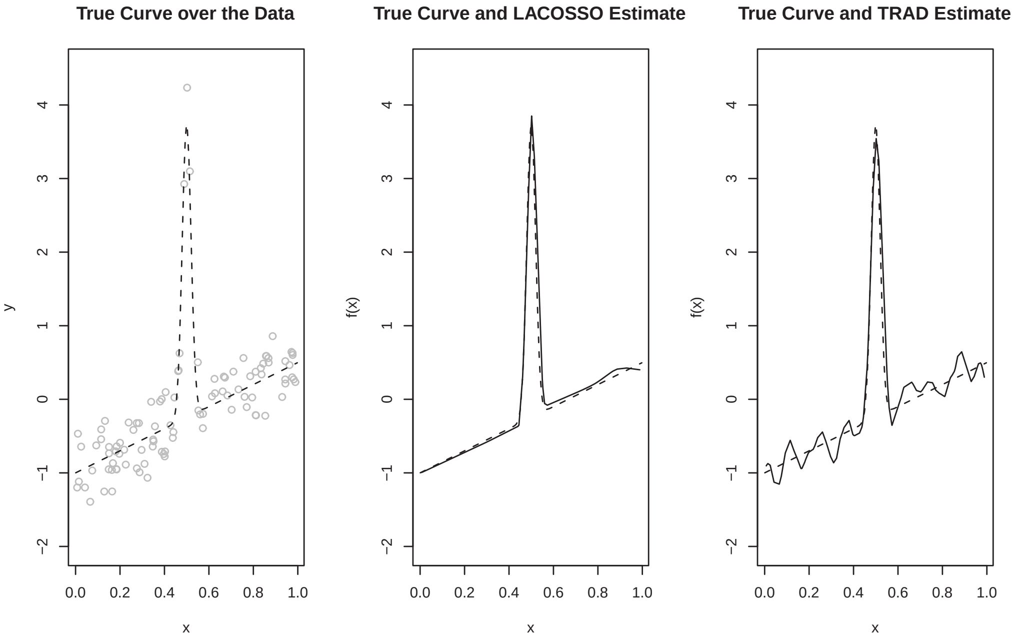

Figure B.1: Mexican hat function. Left: Data generated from the Mexican hat Function with n=100 along with the true function. Middle: The LACOSSO (1,20) estimate (solid) with true function (dashed). Right: The traditional smoothing spline estimate (solid) with the true function (dashed)

.

| Mexihat n=100 | AMSE $\times 10^{-4}$ | % Best |
|---|---|---|
| LOCO | 158.40 (5.82) | 23 |
| SAS(5) | 106.47 (5.40) | 8 |
| TRAD | 205.27 (4.62) | 0 |
| LOKERN | 342.03 (20.81) | 0 |
| MARS | 655.64 (40.3) | 0 |
| LACO (0,5) | 173.38(5.74) | 0 |
| LACO (1,5) | 103.56 (4.39) | 10 |
| LACO (0,10) | 145.07 (5.30) | 1 |
| LACO (1,10) | 88.59 (3.87) | 18 |
| LACO (0,20) | 134.87 (5.40) | 0 |
| LACO (1,20) | 85.08 (4.00) | 40 |
| Mexihat n=200 | AMSE $\times 10^{-4}$ | % Best |
| LOCO | 52.69 (2.71) | 21 |
| SAS(5) | 55.10(3.09) | 10 |
| TRAD | 116.96 (2.62) | 0 |
| LOKERN | 157.31(6.12) | 0 |
| MARS | 645.70 (44.54) | 0 |
| LACO (0,5) | 89.93 (3.07) | 0 |
| LACO (1,5) | 48.66 (1.79) | 12 |
| LACO (0,10) | 77.20 (2.79) | 0 |
| LACO (1,10) | 44.81 (1.78) | 18 |
| LACO (0,20) | 74.61 (2.84) | 0 |
| LACO (1,20) | 43.67 (1.99) | 39 |
| Mexihat n=300 | AMSE $\times 10^{-4}$ | % Best |
| LOCO | 37.32(1.88) | 14 |
| SAS(5) | 35.45(1.59) | 9 |
| TRAD | 81.77(1.57) | 0 |
| LOKERN | 108.64(3.05) | 0 |
| MARS | 493.14(33.11) | 0 |
| LACO (0,5) | 61.78(2.04) | 0 |
| LACO (1,5) | 31.55(1.29) | 16 |
| LACO (0,10) | 52.98(1.80) | 0 |
| LACO (1,10) | 30.62(1.21) | 13 |
| LACO (0,20) | 46.6(1.64) | 2 |
| LACO (1,20) | 27.91(1.16) | 46 |

Table B.1: Table 1: Results of 100 Realizations from Mexican hat. AMSE is the mean square error averaged over the 100 realizations; standard error in parentheses. The percentage of the realizations that a particular method had the smallest MSE among the other methods is given as % Best.

| Harmonic n=100 | AMSE $\times 10^{-4}$ | % Best |
|---|---|---|
| LOCO | 3.75(0.40) | 2 |
| SAS(5) | 3.33(0.17) | 17 |
| TRAD | 9.40(0.26) | 0 |
| LOKERN | 31.50(2.15) | 0 |
| MARS | 44.13(3.17) | 1 |
| LACO (0,5) | 5.12(0.19) | 0 |
| LACO(1,5) | 3.59(0.16) | 8 |
| LACO (0,10) | 3.77(0.16) | 6 |
| LACO (1,10) | 3.17(0.16) | 31 |
| LACO (0,20) | 3.67(0.15) | 6 |
| LACO (1,20) | 3.31(0.17) | 29 |
| Harmonic n=200 | AMSE $\times 10^{-4}$ | % Best |
| LOCO | 2.99(0.29) | 0 |
| SAS(5) | 1.92(0.07) | 17 |
| TRAD | 5.86(0.11) | 0 |
| LOKERN | 22.98(1.50) | 0 |
| MARS | 52.02(2.44) | 0 |
| LACO (0,5) | 2.91(0.09) | 0 |
| LACO(1,5) | 2.06(0.08) | 11 |
| LACO (0,10) | 2.11(0.08) | 6 |
| LACO (1,10) | 1.82(0.09) | 29 |
| LACO (0,20) | 2.03(0.08) | 11 |
| LACO (1,20) | 2.040(0.09) | 24 |
| Harmonic n=300 | AMSE $\times 10^{-4}$ | % Best |
| LOCO | 2.33(0.08) | 0 |
| SAS(5) | 1.3(0.04) | 16 |
| TRAD | 4.09(0.06) | 0 |
| LOKERN | 19.68(1.55) | 0 |
| MARS | 68.37(1.97) | 0 |
| LACO (0,5) | 1.89(0.06) | 0 |
| LACO(1,5) | 1.32(0.04) | 7 |
| LACO (0,10) | 1.37(0.04) | 6 |
| LACO (1,10) | 1.23(0.05) | 39 |
| LACO (0,20) | 1.31(0.04) | 16 |
| LACO (1,20) | 1.37(0.04) | 16 |

Table B.2: Table 2: Results of 100 Realizations from Dampened Harmonic. AMSE is the mean square error averaged over the 100 realizations; standard error in parentheses. The percentage of the realizations that a particular method had the smallest MSE among the other methods is given as % Best.

| Rapid n=100 | AMSE$\times 10^{-4}$ | % Best |
|---|---|---|
| LOCO | 3.54(0.22) | 31 |
| SAS(5) | 4.15(0.25) | 4 |
| TRAD | 5.49(0.16) | 0 |
| LOKERN | 7.41(0.32) | 0 |
| MARS | 5.34(0.34) | 7 |
| LACO (0,5) | 4.26(0.11) | 0 |
| LACO (1,5) | 3.16(0.12) | 3 |
| LACO (0,10) | 4.32(0.12) | 0 |
| LACO (1,10) | 2.69(0.13) | 26 |
| LACO (0,20) | 4.83(0.12) | 0 |
| LACO (1,20) | 2.77(0.14) | 29 |
| Rapid n=200 | AMSE$\times 10^{-4}$ | % Best |
| LOCO | 1.61(0.09) | 33 |
| SAS(5) | 2.05(0.10) | 4 |
| TRAD | 3.05(0.07) | 0 |
| LOKERN | 3.76(0.12) | 0 |
| MARS | 3.37(0.20) | 6 |
| LACO (0,5) | 2.43(0.07) | 0 |
| LACO (1,5) | 1.87(0.08) | 2 |
| LACO (0,10) | 2.46(0.07) | 0 |
| LACO (1,10) | 1.55(0.09) | 17 |
| LACO (0,20) | 2.53(0.07) | 0 |
| LACO (1,20) | 1.45(0.09) | 38 |
| Rapid n=300 | AMSE$\times 10^{-4}$ | % Best |
| LOCO | 1.13(0.05) | 28 |
| SAS(5) | 1.36(0.05) | 11 |
| TRAD | 2.15(0.04) | 0 |
| LOKERN | 2.68(0.07) | 0 |
| MARS | 2.61(0.1) | 7 |
| LACO (0,5) | 1.81(0.06) | 0 |
| LACO (1,5) | 1.33(0.05) | 1 |
| LACO (0,10) | 1.86(0.05) | 0 |
| LACO (1,10) | 1.07(0.04) | 17 |
| LACO (0,20) | 1.87(0.05) | 0 |
| LACO (1,20) | 0.98(0.04) | 36 |

Table B.3: Table 3: Results of 100 Realizations from Rapid Change. AMSE is the mean square error averaged over the 100 realizations; standard error in parentheses. The percentage of the realizations that a particular method had the smallest MSE among the other methods is given as % Best.

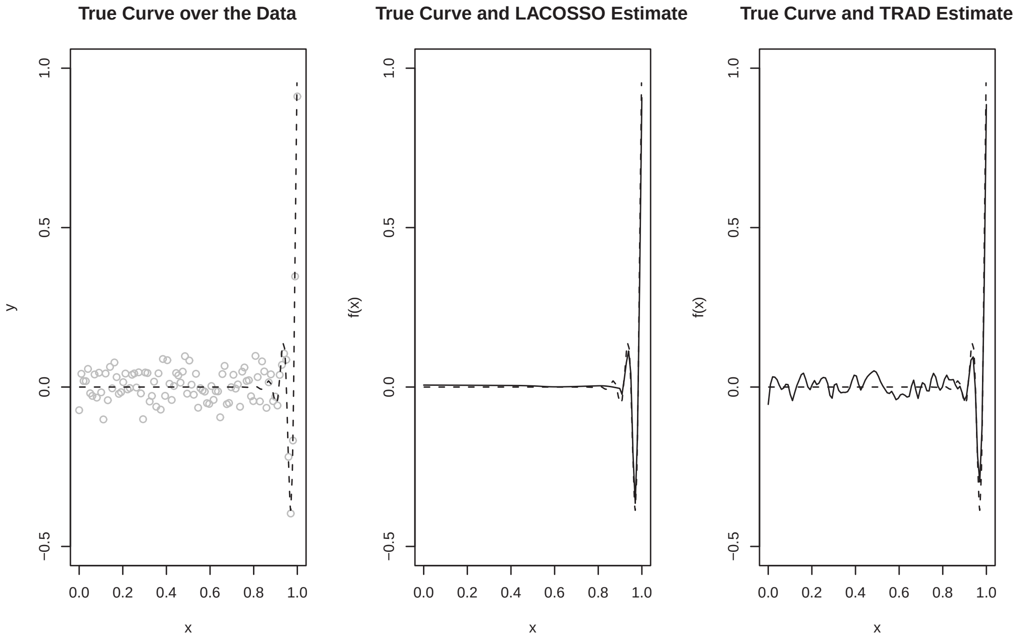

Figure B.2: Dampened harmonic function. Left: Data generated from the dampened harmonic function with n=100 along with the true function. Middle: The LACOSSO (1,10) estimate (solid) with true function (dashed). Right: The traditional smoothing spline estimate (solid) with the true function (dashed)

.

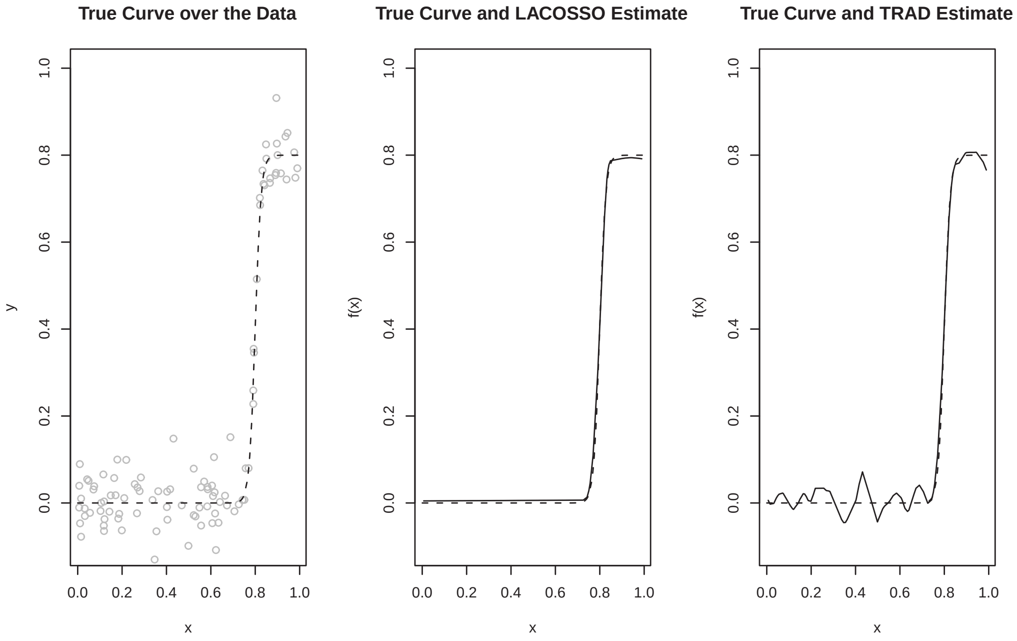

Figure B.3: Rapid change function. Left: Data generated from the rapid change function with n=100 along with the true function. Middle: The LACOSSO (1,20) estimate (solid) with true function (dashed). Right: The traditional smoothing spline estimate (solid) with the true function (dashed)

.

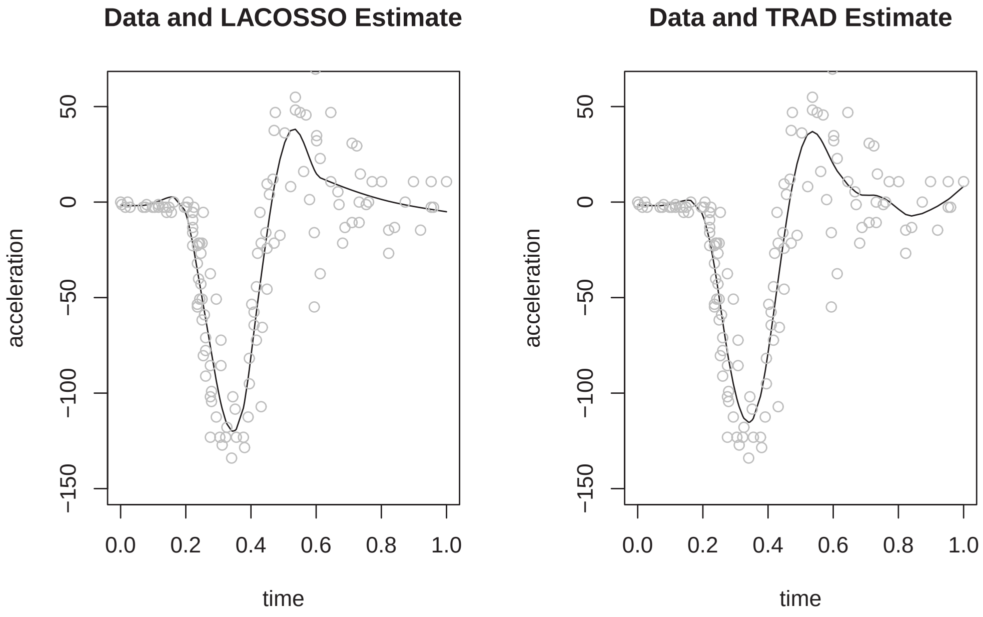

Figure B.4: Left: Motorcycle crash data along with the estimate given by LACOSSO (1,5). Right: Motorcycle crash data along with the estimate given by traditional smoothing spline.

.

# References

[1] D. Allen, *The relationship between variable selection and data augmentation and a method for prediction*, Technometrics **16** (1974), 125–127.

[2] N. Arch and G. Sell, *Linear operator theory in engineering and science*, 1st edition ed., Holt Rinehart and Winston, 1971.

[3] A. Berlinet and C. Thomas-Agnan, *Reproducing kernel hilbert spaces in probability and statistics*, Norwell, MA: Kluwer Academic Publishers, 2004.

[4] D.D. Cox, *Asymptotics for m-type smoothing splines*, Annals of Statistics **11** (1983), no. 2, 530–551.

[5] G. de Barra, *Measure theory and integration*, Ellis Horwood, 1981.

[6] R.L. Eubank, *Nonparametric regression and spline smoothing*, CRC Press, 1999.

[7] J.H. Friedman, *Multivariate adaptive regression splines (with discussion)*, Annals of Statistics **19** (1991), 1–141.

[8] J.H. Friedman and B.W. Silverman, *Flexible parsimonious smoothing and additive modeling (with discussion)*, Technometrics **31** (1989), 3–39.

[9] G. et al Golub, *Generalized cross-validation as a method for choosing a good ridge parameter*, Technometrics **21** (1979), 215–223.

[10] C. Gu, *Smoothing spline ANOVA models*, Springer-Verlag, New York, NY, 2002.

[11] M. Hansen and C. Kooperberg, *Spline adaptation in extended linear models (with discussion)*, Statistical Science **17** (2002), 2–51.

[12] T. Hastie and R.J. Tibshirani, *Generalized additive models*, Chapman & Hall/CRC, 1990.

[13] T. Hastie, R.J. Tibshirani, and J.H. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*, Springer-Verlag, New York, NY, 2001.

[14] A.E. Hoerl and Kennard, *Ridge regression: Biased estimation for nonorthogonal problems*, Technometrics **12** (1970a), 55–67.

[15] Y. Lin and H. Zhang, *Component selection and smoothing in smoothing spline analysis of variance models*, Annals of Statistics **34** (2006), no. 5, 2272–2297.

[16] Z Luo and G Wahba, *Hybrid adaptive splines*, Journal of the American Statistical Association **92** (1997), no. 437, 107–116.

[17] L. Mate, *Hilbert space methods in science and engineering*, 1st edition ed., Adam Hilger, 1989.

[18] Young N., *An introduction to hilbert space*, 1st edition ed., Cambridge University Press, 1988.

[19] A. Nosedal-Sanchez, C. Storlie, T. Lee, and R. Christensen, *Reproducing kernel hilbert spaces for penalized regression: A tutorial*, Statistical Science (2010), under revision.

[20] E. Parzen, *An approach to time series analysis*, Annals of Statistics **32** (1961), 951–989.

[21] ______, *Stochastic processes*, 1st edition ed., Holden-Day, 1962.

[22] A. Pintore, P. Speckman, and C.C. Holmes, *Spatially adaptive smoothing splines*, Biometrika **93** (2006), no. 1, 113–125.

[23] J. Rustagi, *Optimization techniques in statistics*, 1st edition ed., Academic Press, 1994.

[24] M. Schimek (ed.), *Smoothing and regression: Approaches, computation, and application*, John Wiley & Sons, Inc., New York, NY, 2000.

[25] B.W. Silverman, *Some aspects of the spline smoothing approach to non-parametric curve fitting*, Journal of the Royal Statistical Society: Series B **47** (1985), 1–52.

[26] P. Speckman, *Spline smoothing and optimal rates of convergence in nonparametric regression-models*, Annals of Statistics **13** (1985), no. 3, 970–983.

[27] C.J. Stone, M.H. Hansen, C. Kooperberg, and Y.K. Truong, *1994 wald memorial lectures - polynomial splines and their tensor products in extended linear modeling*, Annals of Statistics **25** (1997), no. 4, 1371–1425.

[28] C.B. Storlie, H.D. Bondell, and B.J. Reich, *A locally adaptive penalty for estimation of functions with varying roughness*, Journal of Computational and Graphical Statistics **19** (2010).

[29] C.B. Storlie, H.D. Bondell, B.J. Reich, and H.H. Zhang, *Surface estimation, variable selection, and the nonparametric oracle property*, Statistica Sinica (2010), to appear, URL: www.stat.unm.edu/∼ verb1 ∼ 1storlie/acosso.pdf.

[30] S. van de Geer, *Empirical processes in m-estimation*, Cambridge University Press, 2000.

[31] G. Wahba, *Spline models for observational data*, CBMS-NSF Regional Conference Series in Applied Mathematics, 1990.

[32] G. Wahba, Y. Wang, C. Gu, R. Klein, and B. Klein, *Smoothing spline anova for exponential families, with application to the WESDR*, Annals of Statistics **23** (1995), 1865–1895.

[33] W. et al Wecker, *The signal extraction approach to nonlinear regression and spline smoothing*, JASA (1983), 81–89.

[34] H. Zou, *The adaptive lasso and its oracle properties*, Journal of the American Statistical Association **101** (2006), no. 476, 1418–1429.