A Finite Approximation Framework for Infinite Dimensional Functional Problems

Brayan Ortiz

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

Doctor of Philosophy

University of Washington

2018

Reading Committee:

Noah Simon, Chair

Scott Emerson

Patrick Heagerty

Program Authorized to Offer Degree:
Biostatistics

University of Washington

Abstract

A Finite Approximation Framework for Infinite Dimensional Functional Problems

Brayan Ortiz

Chair of the Supervisory Committee:
Dr. Noah Simon
Department of Biostatistics

It is often of interest to non-parametrically estimate regression functions. Penalized regression (PR) is one effective, well-studied solution to this problem. Unfortunately, in many cases, finding exact solutions to PR problems is computationally intractable. In this manuscript, we propose a *mesh-based approximate solution*, or MBS, for those scenarios. MBS transforms the complicated functional minimization of PR, to a finite parameter, discrete convex minimization allowing us to leverage the tools of modern convex optimization. We show applications of MBS for both univariate and multivariate regression with a number of explicit examples (including isotonic regression and partially linear additive models), and explore how the number of parameters must increase with our sample-size in order for MBS to maintain the rate-optimality of PR. We also give an efficient algorithm to minimize the MBS objective while effectively leveraging the sparsity inherent in MBS.

TABLE OF CONTENTS

Page

List of Figures iii

Chapter 1: Introduction 1

Chapter 2: Mesh Based Solutions to Univariate Penalized Regression 6
2.1 Introduction 6
2.2 Univariate MBS Proposal 8
2.3 Comparisons to Other Univariate Methods 15
2.4 Solving MBS Optimization 17
2.5 Simulation Study 23
2.6 Convergence of MBS for Penalized Regression 26
2.7 Discussion 31

Chapter 3: Mesh Based Solutions to Multivariate Penalized Regression 34
3.1 Introduction 34
3.2 Multivariate MBS 36
3.3 The Multivariate MBS Objective 42
3.4 Simulation Study 44
3.5 Discussion 50

Chapter 4: Extensions of Mesh Based Solutions 52
4.1 Introduction 52
4.2 Partially Linear Additive Models 55
4.3 Penalized Isotonic Regression 57
4.4 The Interaction Problem 59
4.5 Discussion 61

Chapter 5: Discussion 64

Appendix A: Matrix Notation of Univariate/Bivariate MBS 71

Appendix B: Rising Polynomial Basis 74

Appendix C: Proofs of Theoretical Results 75
C.1 Sub-Optimality Lemma 75
C.2 Rate of Convergence for $\Omega\left(\tilde{f}_D\right)$ 77

Appendix D: ADMM Overview 81

Appendix E: Vignette to R Package, **MTV** 84
E.1 Abstract 84
E.2 Introduction 84
E.3 **MTV** for Multivariate Data 88
E.4 Notes and Limitations 93

LIST OF FIGURES

Figure Number — Page

2.1 Comparison of MBS estimates for simulated data. We draw $N = 100$ noisy observations (transparent) from $f(x) = \sin(2\pi x)$ (black). In blue, we draw MBS estimates with $m = 20$ and (r, k) varying, $l = 1$. Vertical lines are drawn at the mesh. 14

2.2 Results for 500 simulations over data generated from an exponential function with noise for $N = 40, 80, 120$. MBS models were fit over varying m, r, and k. Line type denotes k: 0 (····), 1 (- - -), and 2 (—). Top row ranges for all m; bottom row ranges for $m \leq 10$. 24

2.3 Timing results for 500 simulations over a response generated from an exponential function with log-spaced data and noise for $N = 80$. MBS models were fit over varying m, r, and k. Line type denotes k: 0 (····), 1 (- - -), and 2 (—). 26

3.1 Comparison of MBS estimates fit on simulated multivariate data with underlying conditional mean given by bivariate exponential function. We observe $N = 100$ noisy observations (transparent) of a bivariate exponential function shown in (a) and wireframe. We draw MBS fits (blue) using $\{\boldsymbol{r}_1 = (1, 1), \boldsymbol{r}_2 = (1, 0), \boldsymbol{r}_3 = (0, 1)\}$ and $k = 0$. 44

3.2 Comparison of MBS estimates fit on data with underlying conditional mean given by bivariate exponential function. We simulated $N = 1,000, 5,000, 10,000$ noisy observations of a bivariate exponential function shown in Figure 3.1. For each of 500 simulations, we approximated the bivariate fused lasso using MBS over a range of m. 46

3.3 Plotting the Tower and Sombrero functions. We observe $N = 100$ noisy observations (points) from a set of towers (a) and a sombrero (b) in bivariate space. 48

3.4 Comparing MBS and thin plate spline estimates of tower and sombrero functions. We simulated $N = 100$ noisy observations from bivariate functions shown in Figure 3.3. For each of 500 simulations, we approximated the bivariate fused lasso using MBS over a range of m and fit `TPS` using the `fields` package in `R`. We plot the *Relative RMSE* $= \frac{RMSE(MBS)}{RMSE(TPS)}$. 49

3.5 Simulation results of fitting the tower and sombrero functions with a large sample size. We simulated $N = 1,000$ noisy observations from bivariate functions shown in Figure 3.3. For each of 500 simulations, we approximated the bivariate fused lasso using MBS over a range of m. 50

E.1 Total variation solution for piecewise constant function at best cross-validated model. 86

E.2 Total variation solution for piecewise constant function at other solutions. . . 87

E.3 (a) Fused lasso given by MultivarTV. (b) Fused lasso given by genlasso. . . . 88

E.4 Plot of N=100 points drawn from Towers function (drawn as mesh) with noise. 90

E.5 Total variation solution for noisy towers. 91

E.6 Thin plate spline solution for noisy towers. 92

E.7 Total variation solution for noisy towers with many data points. 93

E.8 Residuals plot for perfectly noisy data. 94

ACKNOWLEDGMENTS

First and foremost, I express my deepest appreciation for my advisor and mentor, Noah Simon. He has been my greatest advocate throughout my graduate school career. I thank him for his patient guidance in all of our many endeavors: coding, internships, interview prep, career advice, technical writing, technical speaking, all of it. Noah, you inspire me!

I also thank my fantastic committee: Scott Emerson, Patrick Heagerty, and Andrew Bruce. Each of you has taught me so much about biostatistics and what it takes to be a good human being in science. My conversations with each of you have been insightful and fun, but none more entertaining than when we are all together.

I would also like to thank Jim Hughes and Tim Thornton, for whom I was a research assistant. I owe a big thanks to Gitana Garofalo and the Department of Biostatistics for helping me throughout graduate school. I thank the SLAB-LAB of which I am a proud member: thank you for being ears when talks were not yet polished.

I thank Cindy Perez, for all her love and support and with whom I have built a home here in Seattle. Last but not least, I thank my family, which has grown since I left home. With many months separating my visits, I thank you all for making sure each of my young nieces and nephews knows and asks about their "Tio Brayan."

DEDICATION

For overcoming more than any one person should have to overcome across many lifetimes, I dedicate this thesis to my parents, Armida and Manuel.

Chapter 1

INTRODUCTION

Supervised learning is naturally motivated in hypothesis-driven problems: we may be interested in understanding a possibly nonlinear association between an outcome and a set of predictors. For example, one might be interested in a person's risk of cardiovascular disease (CVD) given information such as their body mass index (BMI) and exercise habits. Risk estimates can help guide preventative measures against outcomes such as CVD. We might hypothesize that an increase in BMI tends to increase risk of CVD, but some levels of BMI might share the same CVD risk. Additionally, this relationship may change with exercise level. We can learn the association between CVD, the predictors of BMI and exercise. Using data from a random sample of N people, we aim to train a statistical model that flexibly estimates a potentially non-linear association between an outcome and predictors, then predict that outcome for new people.

As a statistical problem, we measure a response y_i and p covariates $\boldsymbol{x}_i \in [a, b]^p$, on each of $i = 1, \ldots, N$ observations. We assume the response is generated from a general model with the form

$$y_i = f^* (\boldsymbol{x}_i) + w_i,$$

where f^* is an unknown function from a (sometimes known) function class $\mathcal{F}$, and w_i are iid errors with $\mathrm{E}[w_i|\boldsymbol{x}_i] = 0$ and $\mathrm{var}[w_i|\boldsymbol{x}_i] = \sigma^2 < \infty$. We are interested in finding an estimate $\hat{f}$ of f^* based on the observed data. When $\mathcal{F}$ is known, minimax rate optimal estimators for $f^* \in \mathcal{F}$ can be derived, but may not be easy to calculate.

For finite dimensional families or "parametric classes," minimax optimal estimators are generally easy to calculate, e.g. linear regression, and achieve a fast rate of convergence. However, assuming f^* lives in a specific parametric $\mathcal{F}$ severely constrains the shape of f^*.

When incorrect, this parametric specification will lead to an inconsistent estimator.

Without prior knowledge of f^*, it is generally preferable not to impose strict shape constraints (as assumed by a parametric model). In this case, one might instead assume that f^* lies in a more general family that instead only constrains the overall roughness of f^*. In particular, it is common to choose a roughness measure $P(\cdot)$, and bound $c \in \mathbb{R}$, and consider infinite dimensional classes of the form

$$\mathcal{F}_{P,c} := \{f : [a,b]^p \to \mathbb{R} | P(f) \leq c\}.$$

Generally, $P(f)$ measures the variation of derivatives of f. Examples include total variation, Sobolev norms, and smoothing spline measures of roughness [33, 10]. Often these can be written as $P(f) = \sum_s \|\mathcal{D}_s f\|_\ell^\ell$, $1 \leq \ell \leq \infty$. Here, $\mathcal{D}_s$ is a partial derivative operator, where s is a multi-index of p-many integers denoting the partial derivative order for each feature (see Chapter 3). In this thesis, we will be particularly concerned with measures of the aforementioned type.

For $\mathcal{F} \equiv \mathcal{F}_{P,c}$, one might consider the empirical minimizer:

$$\hat{f} = \underset{f \in \mathcal{F}_{P,c}}{\operatorname{argmin}} \|y - f\|_N^2, \tag{1.1}$$

where $\|z\|_N^2 = \frac{1}{N}\sum_{i=1}^N z_i^2$. For many choices of $P(f)$, $\hat{f}$ is a minimax rate-optimal estimator. In particular, this is the case provided $\mathcal{F}_{P,c}$ is not too "large" (where the size of $\mathcal{F}_{P,c}$ is characterized by its *entropy*, a quantity which we will discuss more formally in Chapter 2). As entropy of a class increases, the optimal rate of convergence slows down, i.e. larger N is needed for small prediction error.

In practice, while we may believe $P(f^*) < \infty$, we rarely know a good bound c. Unfortunately, accidentally using $c < P(f^*)$, would result in an inconsistent estimator. If we alternatively try to solve (1.1) using $c = \infty$, then, for any reasonable $P(\cdot)$, our solution will interpolate the data points (in effect imposing no smoothness) resulting in a useless

estimator.

When we believe $P(f^*) < \infty$, we can use penalized regression to find a solution over

$$\mathcal{F}_{P,\infty} := \{f : [a,b]^p \to \mathbb{R} | P(f) < \infty\}.$$

Under $\mathcal{F}_{P,\infty}$, we can estimate a flexible and rate optimal solution, $\hat{f}$, from the following minimization problem [8, 33]:

$$\hat{f} = \underset{f \in \mathcal{F}_{P,\infty}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - f\left(\boldsymbol{x}_i\right)\right)^2 + \lambda_N P\left(f\right) \tag{1.2}$$

where $\lambda_N \geq 0$ is a tuning parameter and $P(\cdot)$ is treated as a penalty function which penalizes "complexity." We allow the data to inform the complexity of $\hat{f}$ in (1.2) by exchanging goodness of fit measured in the squared loss term for structure governed by the penalty. Solutions to penalized problems have good theoretical properties even for unknown $P(f^*)$: for $\mathcal{F}_{P,\infty}$ not too large (again in terms of entropy) and carefully chosen λ_N, these penalized-regression-based estimates converge at minimax rates ([21], [33]). Examples of penalized regression include smoothing splines, lasso, fused lasso, trend filtering and locally adaptive regression splines [10, 29, 31, 18, 32, 20].

Computationally, when $P(\cdot)$ is convex and $\mathcal{F}$ can be finitely parametrized, e.g. the span of a finite number of basis functions, $\hat{f}$ can be solved efficiently (in polynomial time) in both theory and practice ([35], [10]). Even when $\mathcal{F}$ cannot be finitely parametrized, sometimes the solution for an observed set of data falls in a calculable finite dimensional subfamily, and so we can efficiently solve (1.2). This is the case for smoothing splines, the fused lasso, and in general for reproducing kernel Hilbert spaces ([10], [32], [35]).

However, finite parametrizations of $\mathcal{F}$ can have computational issues. For many variation-based problems that can be finitely parametrized, the solution to (1.2) is known to be a spline, with knots at the observed data-points. Using either natural splines or B-splines to parametrize the solution results in an optimization problem that, in some problems,

can be extremely poorly conditioned [32] (such as in the goodness of fit term, for natural splines, or the penalty term, for B-splines); and in the case of natural splines, does not leverage sparsity. Additionally, in many cases, precisely solving (1.2) may not be required for minimax-optimal statistical accuracy (computationally cheaper alternatives may maintain statistical optimality).

In this dissertation, we propose a computationally tractable framework for approximately solving (1.2), as well as other function-estimation problems, when the true solution may not fall in a simple, easy-to-characterize, finite dimensional subfamily. We approximate the penalized regression problem (1.2) on a mesh, reducing the original infinite dimensional problem to a finite dimensional problem (with asymptotically growing dimension). In this framework, we decouple the approximation of the goodness of fit term and the approximation of the penalty. We exploit this decoupling to develop a computationally tractable approximation to (1.2) that maintains statistical optimality. This allows us to calculate statistically optimal approximate solutions for penalties $P(\cdot)$ that previously had no computationally tractable proposals. For example, there is interest in solving (1.2) with $P(f)$ defined as the total variation (`TV`) semi-norm [28, 33]. For $p > 1$, when f^* lives within the class of functions of bounded total variation, solving (1.2) with this total variation penalty results in an estimator that achieves a faster convergence rate than the contemporary methods used in practice such as thin plate splines (the generalization of smoothing splines) [33, 11, 36]. Unfortunately, for $p > 1$, to date, no computationally efficient solution for the total-variation penalized problem has been proposed. The one proposal we know of for general $p > 1$ by [34] scales poorly and ignores the symmetry of the variation norm. Using our framework, we propose a solution to the `TV` problem.

Chapter 2 focuses on describing our mesh-based approach for univariate non-parametric penalized regression. In our framework, we alter the optimization problem in (1.2) slightly. We partition the sample space into a grid and use the fitted-values on the grid as our optimization parameters. Using the grid fitted values, we replace the penalty function with a finite-difference/Riemann approximation. The fitted values at the data points are ap-

proximated by cleverly interpolating between grid fitted values. We show that approximate solutions constructed this way retain minimax optimality like the exact solutions to (1.2) under conditions dependent on N.

In Chapter 3, we extend this framework to multivariate data, using multivariate interpolators and finite-difference/Riemann approximations to multiple penalties. We will propose a solution to the multivariate total variation problem with a vignette provided in Appendix E. Most notably, we introduce `MTV`, an `R` package with the computational backend written in `C++` for finding solutions with bounded multivariate total variation.

In Chapter 4, we show how the framework extends to other applied problems. We consider three problems. First, we consider the problem of estimating a potentially non-linear and monotonic association between an outcome and predictor, i.e. an isotonic regression problem [2]. Also, we describe the partially linear additive model where we aim to fit a surface over features that may confound a primary linear relationship between an outcome and predictors of interest [15]. Finally, we discuss an interaction problem where we want to estimate the effect of a primary feature on an outcome as it varies according to coefficients on a shared surface over secondary features.

We conclude with a general discussion of the ideas presented in this thesis in Chapter 5.

Chapter 2

MESH BASED SOLUTIONS TO UNIVARIATE PENALIZED REGRESSION

2.1 Introduction

In Chapter 1, we introduced the general regression problem that we aim to solve. Here, in Chapter 2, we describe our proposal for univariate regression, where we measure a response y_i and a covariate $x_i \in [a, b]$, on each of $i = 1, \ldots, N$ observations. With broad generality, we can assume a generative model of the form

$$y_i = f^* (x_i) + w_i,$$

where f^* is an unknown function from a known function class $\mathcal{F}$, and w_i are iid errors with $\mathrm{E}[w_i|x_i] = 0$ and $\mathrm{var}[w_i|x_i] = \sigma^2 < \infty$. We are interested in estimating f^* based on the observed data.

One common approach is to use penalized regression as given in (1.2) ([8], [33]):

$$\hat{f} = \underset{f \in \mathcal{F}_{P,\infty}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda_N P(f),$$

where

$$\mathcal{F}_{P,\infty} := \{f : [a, b] \to \mathbb{R} | P(f) < \infty\},$$

$\lambda_N \geq 0$ is a tuning parameter, and $P(\cdot)$ is a penalty function which penalizes "complexity." With carefully chosen λ_N, we know that $\hat{f}$ converges at a minimax rate that is respective of $P(f)$ and the parametrization of $\mathcal{F}$ [21, 33]. However, for hard problems, computational efficiency depends on the true solution falling in a simple finite dimensional subfamily, which

is not guaranteed.

We propose a computationally tractable framework for approximately solving (1.2), for a broad class of roughness penalties $P(\cdot)$, when the true solution does not fall in a simple finite dimensional subfamily. In our framework, we alter the optimization problem in (1.2) slightly. First, we select a mesh of m points over the domain of x_i and use the fitted-values at those points as our optimization parameters. Next, we change the penalty function to a finite-difference/Riemann approximation over the m mesh points. Finally, the *fitted values at the data points* are approximated by an interpolation scheme between *fitted values at mesh points*. In this way, we approximate $P(f)$ independently of a chosen parametrization of $\mathcal{F}$. We refer to the general approach as MBS, or *mesh based solution*.

More formally, we consider a finite set of points, or mesh, $D \equiv \{d_1, \ldots, d_m\} \subset [a, b]$, such that the observed $x_1, \ldots, x_N$ are within the convex hull of D. Using D, we formulate an approximation to our original problem (1.2):

$$\tilde{f_D} = \text{argmin}_{f_D \in \mathbb{R}^m} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \Omega_{x_i}(f_D)\right)^2 + \lambda_N P_D\left(f_D\right), \tag{2.1}$$

where $P_D(f_D)$ is an approximation to $P(f)$ based on finite-differences/Riemann sums using only our fitted values on the mesh, $f_D \equiv \left(f\left(d_1\right), \ldots, f\left(d_m\right)\right)^\top$; and $\Omega : \mathbb{R}^m \to \mathcal{F}$ is an *interpolator*; where $\Omega_{x_i}(f_D) \equiv \left[\Omega\left(f_D\right)\right](x_i)$ takes in fitted values on our mesh, and an x_i (potentially not on the mesh), and calculates an interpolated fit at a point x_i. For P_D convex and $\Omega_{x_i}(f_D)$ linear in f_D, (2.1) is a convex problem. Our estimate of f^* is given simply by $\Omega(\tilde{f_D})$.

In Section 2.2, we discuss choices of $P_D(f_D)$ (2.2.1) and $\Omega_{x_i}(f_D)$ (2.2.2) and how they impact the closeness of the approximate solution to the exact solution. We briefly discuss similarities to other contemporary nonparametric univariate methods (Section 2.3). We describe an efficient solver for convex objective function (2.1) based on the alternating direction method of multipliers (ADMM) in Section 2.4. In Section 2.5, we run a simulation study highlighting that, for even a modest number of mesh points, the approximation error induced

by replacing (1.2) by (2.1) can be smaller than estimation error of the original problem. In Section 2.6, we provide theoretical results supporting our findings.

2.2 Univariate MBS Proposal

We discuss, in detail, our MBS proposal for approximating (1.2). For now we restrict ourselves to penalties of the form

$$P(f) = \int \left| f^{(r+1)}(d) \right|^{\ell} \partial d$$

for some integer $r \geq 0$ and $\ell > 0$, and where $f^{(r)} = \frac{\partial^r}{\partial d^r} f(d)$. These Sobolev-norm penalties are a fairly broad class, which include smoothing splines and total-variation penalties among others (one can also apply these ideas with $r = \infty$).

A mesh, D, on $[a, b]$, can be written as

$$D : a = d_1 \leq d_2 \leq \ldots \leq d_{m-1} \leq d_m = b.$$

Let $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_{m-1})$ denote the bin widths within the mesh, where $\delta_j = d_{j+1} - d_j$ for $j = 1, \ldots, m-1$. Define $\delta_{max} = \max_j \delta_j$. Often, we specify D as a regular or even mesh with widths δ, where $\delta \equiv \delta_j = d_{j+1} - d_j = \frac{b-a}{m}$ for $j = 1, \ldots, m-1$. Note, we will often want to think of $f_D = (f(d_1), \ldots, f(d_m))$ as a function, so we sometimes slightly abuse notation by using $f(d_j) \equiv (f_D)_j$.

For our discrete, approximate problem (2.1) we need to specify two pieces:

- $P_D(f_D)$: an approximation to $P(f)$ calculated using only $d_1, \ldots, d_m$ and $f_D = (f(d_1), \ldots, f(d_m))^{\top}$.

- $\Omega_{x_i}(f_D)$: an interpolation function, which allows us to map f_D to a function that can be evaluated on the entirety of $[a, b]$; this is important as, generally, $x_i \notin D$ for observed x_i. This interpolation should be a linear function of f_D to retain convexity of (2.1).

The degree to which our solution to the mesh-based problem (2.1), retains the nice properties

of the solution to (1.2) will depend on our choice of D, P_D and $\Omega_{x_i}(f_D)$. We will discuss how to make these choices below.

2.2.1 Choosing $P_D(f_D)$

We use finite-differences/Riemann sums to approximate P. This is chosen in part because of the form we have assumed for P: $P(f) = \int \left|f^{(r+1)}(d)\right|^\ell \partial d$. We approximate the operator $\frac{\partial}{\partial d}$ using discrete differences. We define the normalized first order difference function $\Delta_m^1 : \mathbb{R}^m \to \mathbb{R}^{m-1}$ such that

$$[\Delta_m^1 f_D]_i = \frac{f(d_{i+1}) - f(d_i)}{\delta_i},$$

where $i = 1, \ldots, m-1$. For regular D, we define a normalized rth order difference function $\Delta_m^r : \mathbb{R}^m \to \mathbb{R}^{m-r}$ such that

$$[\Delta_m^r f_D]_i = \left[\Delta_{m-1}^{r-1}\left[\Delta_m^1 f_D\right]\right]_i = \ldots = \left[\Delta_{m-r+1}^1\left[\Delta_{m-r}^1\left[\ldots\left[\Delta_m^1 f_D\right]\ldots\right]\right]\right]_i,$$

where $i = 1, \ldots, m-r$. We now approximate the integral with a Riemann sum (again using our mesh D). Thus, for general r and ℓ, our approximation becomes

$$P_D(f_D) = \sum_{i=1}^{m-r} \left|\delta^{\frac{1}{\ell}}[\Delta_m^{r+1} f_D]_i\right|^\ell. \tag{2.2}$$

The algebraic formulation of $P_D(f_D)$ will be useful in the multivariate case. As with other penalized methods, our penalty $P_D(f_D)$ will not be able to approximate high frequency periodic variation at frequencies smaller than the mesh widths, δ.

In Appendix A, we present matrix representations for $P_D(f_D)$, for both regular and irregular mesh cases, which will be useful for deriving solvers for univariate MBS. Briefly, we show in Appendix A that

$$P_D(f_D) = \left\|\delta^{\frac{1}{\ell}-r-1}\bar{\Delta}_m^{(r+1)} f_D\right\|_\ell^\ell, \tag{2.3}$$

where recursively $\bar{\Delta}_n^{(r)} = \bar{\Delta}_{n-1}^{(r-1)} \cdot \bar{\Delta}_n^{(1)}$ and

$$\bar{\Delta}_n^{(1)} = \begin{pmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & \dots & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(n-1)\times n}.$$

2.2.2 Choosing $\Omega_{x_i}(f_D)$

There is a vast literature on function interpolation/estimation [9]. Popular choices include linear interpolation, higher order piecewise polynomial interpolation and, in particular, splines-based interpolation. In our case, a good interpolator, $\Omega_{x_i}(f_D)$, will be linear in f_D to preserve convexity of (2.1) and permit a computationally inexpensive solution to (2.1). We briefly discuss a general approach to interpolation, then discuss a particular piecewise polynomial interpolator that meets both of our criteria.

Consider the general problem of interpolating b points $\theta_{1:b} = [f(d_1), \dots, f(d_b)]^\top$, via $\tilde{f} \leftarrow \sum_{i=1}^b \alpha_i \psi_i$ a linear combination of pre-specified basis elements $\psi_1, \dots, \psi_b$ and coefficients $\alpha = (\alpha_1, \dots, \alpha_b)^\top$. With design matrix Ψ where $\Psi_{ij} = \psi_j(d_i)$, we find coefficients α by solving the linear system $\Psi\alpha = \theta_{1:b}$, where $\theta_{1:b} = (\theta_1, \dots, \theta_b)^\top$. Thus, $\hat{\alpha} = \Psi^{-1}\theta_{1:b}$. To find $\tilde{f}(x_{new})$ for a new point x_{new}: first evaluate x_{new} over the basis, $\tilde{\psi} = [\psi_1(x_{new}), \dots, \psi_b(x_{new})]^\top$; then compute $\tilde{f}(x_{new}) = \tilde{\psi}^\top\hat{\alpha} = \tilde{\psi}^\top\Psi^{-1}\theta_{1:b}$. Note that this is linear in $\theta_{1:b}$. Spline-based interpolations can be defined using this framework by specifying the basis elements $\psi_1, \dots, \psi_b$ appropriately. For example, interpolating via regression splines is given by defining $\psi_1, \dots, \psi_b$ as the truncated power basis [32].

We now motivate a special local polynomial, then we construct it using the general framework described in the previous paragraph. Suppose we have values $f_D = (f(d_1), \dots, f(d_m))^\top$, where $d_1 < d_2 \dots < d_m$. We can interpolate at a point $x_1 \in [d_1, d_2]$ to approximate $f(x_1)$

using the line connecting $f(d_1)$ and $f(d_2)$,

$$f(x_1) \approx f(d_1) + \frac{f(d_2) - f(d_1)}{d_2 - d_1}(x_1 - d_1).$$

For $x_2 \in [d_2, d_3]$, the approximation uses $f(d_2)$ and $f(d_3)$; and so forth. In general, an observation can be fit from a kth order polynomial that uses only $k+1$ points, since that uniquely defines a kth degree polynomial. We propose to interpolate at points $x_1 \in [d_1, d_{k+1}]$ and $x_2 \in [d_{k+1}, d_{2k+1}]$ using only $f(d_1), \ldots, f(d_{k+1})$ and $f(d_{k+1}), \ldots, f(d_{2k+1})$, respectively. Using only $k+1$ points to fit at each x_i allows for fast computation as we will see in the next paragraph. However, this type of local interpolation is different from spline-based interpolation described before, since it lacks continuous first derivatives globally.

We now formally construct our special local polynomial using the general interpolation framework, which requires determining the points $\theta_{1:b}$ that we interpolate over and the basis elements $\psi_1, \ldots, \psi_b$ that define the interpolation. For local interpolation at a point x_i, we find a neighborhood of $k+1$ local mesh points $\tilde{d} = \left(\tilde{d}_1, \ldots, \tilde{d}_{k+1}\right)^\top$ such that $x_i \in [\tilde{d}_1, \tilde{d}_{k+1}]$. Let $\theta_{1:k+1} = \left(f(\tilde{d}_1), \ldots, f(\tilde{d}_{k+1})\right)^\top = (\theta_1, \ldots, \theta_{k+1})^\top$ be the evaluations of f for the mesh neighborhood about x_i. We define an interpolation of $\theta_{1:k+1}$ using the basis functions:

$$\psi_1(x) = 1, \psi_2(x) = x, \ldots, \text{ and } \psi_{k+1}(x) = x^k.$$

Based on these basis functions, this local polynomial interpolator coincides with the spline interpolator for $k = 0, 1$. As in the general approach, we consider a design matrix $\Psi \in \mathbb{R}^{(k+1)\times(k+1)}$ with $\Psi_{i'j} = \psi_j(\tilde{d}_{i'})$. Thus, we can calculate coefficients $\hat{\alpha} = \Psi^{-1}\theta_{1:k+1}$. To find the value of $\tilde{f}(x_i)$ at a point x_i, we form $\tilde{\boldsymbol{\psi}} = (\psi_1(x_i), \ldots, \psi_{k+1}(x_i))^\top$ and compute $\tilde{f}(x_i) = \tilde{\boldsymbol{\psi}}^\top\hat{\alpha} = \tilde{\boldsymbol{\psi}}^\top\Psi^{-1}\theta_{1:k+1} = \tilde{o}_i^\top\theta_{1:k+1}$. Note that $\tilde{o}_i$ is specific to x_i. At a new point $x_j \notin [\tilde{d}_1, \tilde{d}_{k+1}]$, we move to another set of neighboring mesh points $\tilde{d}'$ which leads to $\tilde{o}_j$. Hence, we call this interpolation scheme the *simple piecewise polynomial interpolator* or

SPPI. As before, for observations $\boldsymbol{x} = (x_1, \ldots, x_n)$, we can write

$$\Omega_{\boldsymbol{x}}(f_D) = O f_D,$$

where $O = (\boldsymbol{o}_1, \ldots, \boldsymbol{o}_n)^\top$, and for $x_i \in [d_j, d_{j+k}]$, we the vector $\boldsymbol{o}_i$ as

$$\boldsymbol{o}_i = \left(0, \ldots, 0, \tilde{\boldsymbol{o}}_i^\top, 0, \ldots, 0\right)^\top,$$

where $\tilde{\boldsymbol{o}}_i$ occupies indices j through $j + k$.

As it was generated from the simple piecewise polynomial, the interpolation matrix, O, is k-banded, which allows for easy storage and manipulation. However, it is unclear what order k interpolation we will want to use. Under our current penalized regression framework, we penalize rth order smoothness by approximating $P(f)$ via rth order differences and Riemann sums, as discussed in Section 2.2.1. By penalizing against rth order smoothness, we set an *a priori* belief that the solution lacks higher (than r) order smoothness. Intuitively then, we should not interpolate at a higher order than the assumed smoothness, or allow $k > r$. For other methods with different spline based interpolations, setting $k = r$ is a requirement. For example, in rth order smoothing splines, rth order splines must be used to evaluate the rth order smoothing spline penalty $P(f)$ and so observations are fit using the rth order spline. However, we approximate $P(f)$ using rth order differences which are independent of whichever kth order interpolator we choose. Thus, in our framework, it is possible to set $k < r$ and in general we allow $k \leq r$. We will show later both via simulation and theory that the closer k is to r, the fewer points we need in our mesh to achieve a close approximation of the exact solution. However, for many problems, $k = 1, 2$ (linear and quadratic interpolation) may do well enough with a modest number of mesh points, (conservatively, no more than $m = \sqrt{N}$ mesh points should be needed, see 2.6.1). This is important since low order interpolators such as linear and quadratic SLPs are not computationally demanding.

2.2.3 Writing the Univariate MBS Problem

We have introduced $P_D(f_D)$ and $\Omega_{x_i}(f_D)$. For response y and data $\boldsymbol{x}$ with related kth-order interpolation matrix O, the MBS objective (2.1) can be written simply as

$$\tilde{f_D} = \underset{f_D \in \mathbb{R}^m}{\operatorname{argmin}} \|y - Of_D\|_N^2 + \lambda_N \left\| \delta^{\frac{1}{\ell}-r-1} \Delta_m^{(r+1)} f_D \right\|_\ell^\ell. \tag{2.4}$$

Often, we may let λ_N absorb the constants related to δ and write (2.4) as

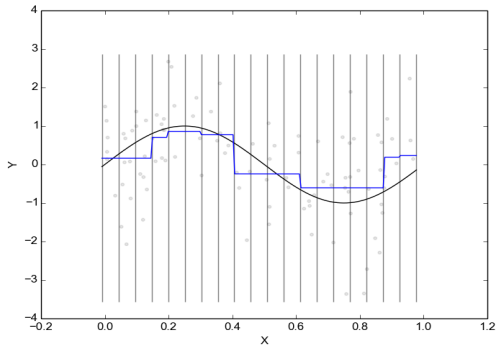
$$\tilde{f_D} = \underset{f_D \in \mathbb{R}^m}{\operatorname{argmin}} \|y - Of_D\|_N^2 + \lambda_N \left\| \Delta_m^{(r+1)} f_D \right\|_\ell^\ell. \tag{2.5}$$

We denote the fitted values of $\tilde{f}$ at the observed x-values as $\tilde{f} = O\tilde{f_D}$.

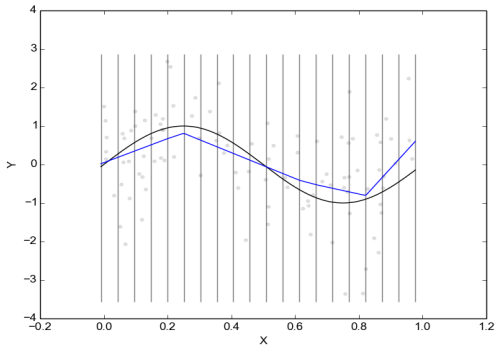
In Figure 2.1, we show a small example using MBS solving (2.5) with $l = 1$ and various k, r. We compare our MBS fits to the true underlying regression curve. The fits were based on observations $y_i = \sin(2\pi x_i) + w_i$, where $x_i \sim \text{Unif}[0, 1]$ and $w_i \sim N(0, 1)$, for $i = 1, \ldots, N$. Where the vertical lines intersect the blue lines denote the $m = 20$ mesh fitted values, the optimization parameters of (2.5). By solving (2.5) with $k = r$, we get rth-order piecewise polynomials with adaptively chosen knots. With piecewise constant structure and many sequential mesh fitted values fit to be the same, the MBS fit in Figure 2.1a looks like a fused lasso or 0th order trend filtering solution. Similarly, MBS fits in Figures 2.1bc look like 1st and 2nd order trend filter solutions. In the next section, we discuss the close relationship between MBS and other univariate methods.

2.2.4 Basis Representation of MBS

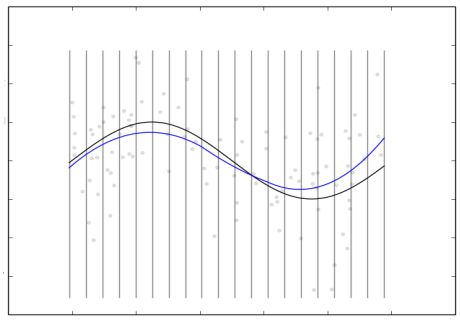
Here we show that one can equivalently write our MBS objective (2.1) using a basis expansion. As an alternative to the route we discussed above, one might consider optimizing (1.2) over a linear class $\mathcal{F} \equiv \text{span}\,(\psi_1, \ldots, \psi_K)$ with basis elements $\psi_1, \ldots, \psi_K$. As we will discuss in Section 2.3, for some bases, the rth order total variation $\int_x \left| \frac{\partial^r}{\partial x^r} \sum \beta_k \psi_k(x) \right| dx$ has a simple representation, however often it does not. In cases where it does not, we could approximate



(a) $r = 0$, $k = 0$



(b) $r = 1$, $k = 1$



(c) $r = 2$, $k = 2$

Figure 2.1: Comparison of MBS estimates for simulated data. We draw $N = 100$ noisy observations (transparent) from $f(x) = \sin(2\pi x)$ (black). In blue, we draw MBS estimates with $m = 20$ and (r, k) varying, $l = 1$. Vertical lines are drawn at the mesh.

it using a similar discretization strategy as before: For a given element of our linear space $f = \sum \beta_k \psi_k$ and a mesh $D = (d_1, \ldots, d_m)$ define $f_D(\theta) \equiv [\sum_k \theta_k \psi_k(d_1), \ldots, \sum_k \theta_k \psi_k(d_m)]$. Now consider the problem:

$$\tilde{\theta} = \underset{\theta \in \mathbb{R}^K}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \sum_k \theta_k \psi_k(x_i) \right)^2 + \lambda P_D(f_D(\theta)) \tag{2.6}$$

This is exactly equivalent to our original formulation for MBS , given by (2.5) if, the matrix Ψ_D, with elements $\Psi_{D(k,j)} = \psi_k(d_j)$, is a basis for $\mathbb{R}^m$. In particular, in this case our

interpolator is precisely $\Omega_x\left(f_D(\theta)\right) = \sum_k \theta_k \psi_k(x)$.

Note that if we define $(\psi_1, \ldots, \psi_K)$ by the "rising polynomial basis" given in Appendix B, then the basis-expansion-based interpolation-scheme is equivalent to that described in Section 2.2.2 using the SPPI. However, the SPPI as defined earlier with sparse Ψ_D leads to more efficient matrix operations than the rising polynomial basis (see Appendix B). For this computational reason, we prefer implementing interpolation using the simple piecewise polynomial over the rising polynomial basis.

2.3 Comparisons to Other Univariate Methods

There are two other methods for approximately solving (1.2) with a total variation penalty: ℓ_1 trend filtering (`TF`) by [32] and locally adaptive regression splines (`LocARS`) by [20]. Like the exact solution to the functional problem (1.2), both `TF` and `LocARS` give minimax rate-optimal solutions. These two methods use the basis expansion framework discussed in Section 2.2.4. They both solve a total variation problem

$$\hat{f}_{TV} \in \underset{f \in \mathcal{F}_N^{restrict}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - f\left(x_i\right)\right)^2 + \lambda \cdot TV\left(f^{(r)}\right), \tag{2.7}$$

over two different linear subspaces, $\mathcal{F}_N^{restrict}$. For `LocARS`, one uses $\mathcal{F}_N^{restrict} \equiv \operatorname{span}\left(\psi_1^{LocARS}, \ldots, \psi_N^{LocARS}\right)$ where the ψ_i^{locARS} are from the r-th order truncated power basis with knots at the observations, x_i. For `TF` one uses $\mathcal{F}_N^{restrict} \equiv \operatorname{span}\left(\psi_1^{TF}, \ldots, \psi_N^{TF}\right)$, where the ψ_i^{TF} are from the r-th order falling factorial basis with knots at the observations, x_i. More details on these bases can be found in [32]. These bases are chosen in part because functions in their span permit a simple, finite dimensional representation of $TV\left(f^{(r)}\right)$. This is in contrast to MBS where the penalty was instead approximated via $P_D(f_D)$. In fact, for each of these bases, (2.7) can be rewritten as a simple lasso problem:

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^N}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \sum_{k=1}^{N} \theta_k \psi_k\left(x_i\right)\right)^2 + \lambda \sum_{k=r+1}^{N} \left|\theta_k\right|, \tag{2.8}$$

with $\hat{f} \leftarrow \sum_{k=1}^{N} \hat{\theta}_k \psi_k$, where $\psi_k \equiv \psi_k^{TF}$ or $\psi_k \equiv \psi_k^{locARS}$ for `TF` and `LocARS`, respectively. While the lasso form of these problems is useful for interpretation, solving either `TF` or `LocARS` by applying a general purpose lasso solver to (2.8) is very inefficient, as the design matrix is poorly conditioned. `TF`, unlike `LocARS`, is more amenable to efficient computation: One can rewrite `TF` as a particular instance of MBS,

$$\hat{f}_D \in \underset{f_D \in \mathbb{R}^N}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - f_D\left(x_i\right)\right)^2 + \lambda P_D\left(f_D\right) \tag{2.9}$$

where we use a mesh with grid points at the observations $D = (x_1, \ldots, x_n)$, and P_D is our discretized TV penalty from before. There are 2 main downsides to using the observations as our mesh: 1) We have N optimization parameters, which can slow down computation, when for statistical accuracy many fewer may be needed; and 2) the uneven spacing of the observations means that (2.9) is still a poorly conditioned problem — this leads to instability for many convex solvers (as noted in [27]). The ability of our method to use a regular mesh, with fewer than N mesh points is a potentially significant asset in problems with large sample size.

MBS has an additional advantage when the features lie in 2 or more dimensions (discussed further in the next chapter). The obvious extensions of `TF` and `LocARS` to higher dimensions use complicated and potentially computationally inefficient basis expansions in 2+ dimensions (eg. thin plate splines). As we will see in Chapter 3, MBS allows us to work with multivariate local polynomial interpolators. These are simple objects, and allow us to maintain a sparse representation of our interpolation matrix. In addition, because we work with a discrete approximation to our penalty of choice, we can simultaneously use sparse representations of our discretized penalty and our interpolator. This allows us to easily extend our method and computation to multiple features and thousands of observations.

2.4 Solving MBS Optimization

In this section, we describe an ADMM solver for the univariate MBS problem when $\ell = 1$. Note that in this case the solver will be similar to the standard ADMM algorithm for the trend filtering problem given by [27]. For an ADMM algorithm as described by [6], we begin by rewriting (2.4) with $\ell = 1$ as

$$\min_{f_D \in \mathbb{R}^m, \alpha \in \mathbb{R}^{m-k-1}} \|y - Of_D\|_2^2 + \lambda_n \|\alpha\|_1 \text{ subject to } \alpha = \Delta_m^{(r+1)} f_D. \tag{2.10}$$

We write the augmented Lagrangian as

$$L(f_D, \alpha, u) = \|y - Of_D\|_2^2 + \lambda\|\alpha\|_1 + \frac{\rho}{2}\|\alpha - \Delta_m^{(r+1)} f_D + u\|_2^2 - \frac{\rho}{2}\|u\|_2^2,$$

where we treat ρ as a step-size variable for the dual update that gets smaller when the primal (α) and dual (u) variables get closer. The ADMM iterates for f_D are found by solving for f_D in

$$\nabla_{f_D} L(f_D, \alpha, u) = 0.$$

The same is done for α, while the dual variable u is updated by the primal residual, $r_{primal}^{j+1} = \alpha^{j+1} - \Delta_m^{(r+1)} f_D^{j+1}$. This results in the following ADMM iterates:

$$f_D^{j+1} \leftarrow \left(O^\top O + \rho(\Delta_m^{(r+1)})^\top \Delta_m^{(r+1)}\right)^{-1} \left(O^\top y + \rho(\Delta_m^{(r+1)})^\top (\alpha^j + u^j)\right), \tag{2.11}$$

$$\alpha^{j+1} \leftarrow S_{\lambda/\rho}\left(\Delta_m^{(r+1)} f_D^{j+1} - u^j\right), \tag{2.12}$$

$$u^{j+1} \leftarrow u^j + \alpha^{j+1} - \Delta_m^{(r+1)} f_D^{j+1}. \tag{2.13}$$

We sequentially calculate f_D^{j+1}, α^{j+1}, then u^{j+1} to complete one ADMM cycle. We can initialize $\rho = \lambda$, but it is useful to let ρ change at each iteration based on how far apart the primal and dual residuals are from each other [6]. Note that the dual residual is given by $r_{dual}^{j+1} = \rho\left(\Delta_m^{(r+1)}\right)^\top (u^{j+1} - u^j)$. Although there is no theoretical justification, there is

empirical evidence to support this adaptive ρ strategy [6]. We can further optimize the calculations by using warm-starts of f_d, α and u.

Our ADMM stopping criteria require that both the primal and dual residuals meet a tolerance: $\|r_{primal}^{j+1}\|_2 < \epsilon$ and $\|r_{dual}^{j+1}\|_2 < \epsilon$ (we choose $\epsilon = 10^{-4}$). However, satisfying these stopping criteria can require a large number of iterations, so it is important for each iteration to be computationally inexpensive. We can determine the complexity of each iteration by analyzing the cost of the matrix operations for each update. When using kth order SPPIs, O becomes banded with bandwidth $k+2$. Meanwhile, $\Delta_m^{(r+1)}$ is also banded with bandwidth $r+2$. Since $k \leq r$, the f_D-update can be implemented with time $O\left(m(r+2)^2 + n(r+2)\right)$ and $O\left(m(r+2)^2\right)$ after the first iteration with caching. Updating α with coordinate-wise soft-thresholding ($S_{\lambda/\rho}$) requires time $O(m-k-1)$, while updating u takes $O\left(m(r+2)\right)$ time. Considering k and r as constants, a full iteration of ADMM updates can be done in linear time.

2.4.1 Parallelized ADMM

Since each iteration runs linearly in m, large values of m will be computationally difficult. One way to reduce computational complexity is to distribute the fitting procedure across multiple processors. Using consensus ADMM, we can split each ADMM iteration across the mesh points [6]. In the ADMM we described in Section 2.4, targets of optimizations were $f_D = (f(d_1), \ldots, f(d_m))$, training observations were fit using some interpolation of f_D which had the form Of_D, and we approximated the penalty using sums of differences, $P_D(f_D) = \|\Delta_m^{(r+1)} f_D\|_1$. For consensus ADMM, we will partition the mesh points D such that $D = (D_1, \ldots, D_M)$, with $D_i \in \mathbb{R}^{m_i}$, where $\sum_{i=1}^{M} m_i = m^*$. For our consensus description here, we will assume $m_1 = m_2 = \ldots = m_M$, which is not unusual since the size of each partition could be determined by the interpolation order, $m_i = k+1$. Using this mesh notation, the mesh fitted values can be partitioned as $f_D^{part} = (f_{D_1}, \ldots, f_{D_M})^\top$. Note that if we interpolate via SPPI, then the last element of f_{D_i} will be the same as the first element of $f_{D_{i+1}}$, or $f_{D_i}[m_i] = f_{D_{i+1}}[1]$; so $m^* = m + M - 1$ and $f_D^{part} \in \mathbb{R}^{m+M-1}$.

Let O_i denote the interpolation matrix for observations that fall in D_i. Using the convention that $[a;b] = \begin{bmatrix} a \\ b \end{bmatrix}$, we define a matrix $\tilde{O}_i = [0;\ldots;0;O_i;0;\ldots;0] \in \mathbb{R}^{N\times m_i}$. For example, $\tilde{O}_1 = [O_1;0] \in \mathbb{R}^{N\times m_1}$. In this way, $Of_D = \sum_{i=1}^{M} \tilde{O}_i f_{D_i}$. Note that each $\tilde{O}_i$ is sparse, but not banded. Finally, we can evaluate our approximation to the penalty using f_D^{part} by $\|\Delta_m^{(r+1)} f_D\|_1 = \sum_{i=1}^{M} \|\Delta_{m_i}^{(r+1)} f_{D_i}\|_1$.

Using f_D^{part}, the objective we aim to minimize is

$$\min \left\| y - \sum_{i=1}^{M} \tilde{O}_i f_{D_i} \right\|_N^2 + \lambda \sum_{i=1}^{M} \|\Delta_{m_i}^{(r+1)} f_{D_i}\|_1 \text{ s.t. } f_{D_i}[m_i] = f_{D_{i+1}}[1], \tag{2.14}$$

for $i = 1,\ldots,M-1$, where the constraint is necessary since we define $\tilde{O}_i$ by the SPPI. Let $b_i \in \mathbb{R}^{m+M-1}$, such that $b_i[im_i] = 1$, $b_i[im_i+1] = -1$, and otherwise, $b_i = 0$, for $i = 1,\ldots,M-1$. We define a matrix

$$B = \begin{bmatrix} b_1 \\ \vdots \\ b_M \end{bmatrix} \in \mathbb{R}^{M-1\times m+M-1}.$$

For example, if $m_i = 3$ for $i = 1,2,3$ (so $M = 3$), then

$$B = \begin{bmatrix} 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \end{bmatrix}.$$

It will be useful to denote the blocks B_i of $B = [B_1,\ldots,B_m]$, which are specific to D_i. In our previous example,

$$B_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, B_2 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, B_3 = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}.$$

Note that $B_i^\top B_{i+1} = 0$. We can rewrite the constraint in (2.14) as $Bf_D^{part} = 0$.

We begin to solve (2.14) by introducing primal variables via some equality constraints: $z_i \equiv \begin{bmatrix} \gamma_i \\ \alpha_i \end{bmatrix} = A_i f_{D_i}$, where $A_i = \begin{bmatrix} \tilde{O}_i \\ \Delta_{m_i}^{(r+1)} \end{bmatrix} \in \mathbb{R}^{N+m_i-r-1 \times m_i}$, for $i = 1, \ldots, M$. Let $A = \begin{bmatrix} A_1 & A_2 & \ldots & A_M \end{bmatrix} \in \mathbb{R}^{N+m_i-r-1 \times m+M-1}$, $z = \begin{bmatrix} z_1 \\ \vdots \\ z_M \end{bmatrix} \in \mathbb{R}^{N+m_i-r-1}$, $C = \begin{bmatrix} A \\ B \end{bmatrix} \in \mathbb{R}^{N+m_i-r+M-2 \times m+M-1}$, and $\theta = \begin{bmatrix} z \\ 0 \end{bmatrix} \in \mathbb{R}^{N+m_i-r+M-2}$. Furthermore, let $1_{\gamma_i}\theta = \gamma_i$ and $1_{\alpha_i}\theta = \alpha_i$. We aim to solve the following minimization problem:

$$\min \left\| y - \sum_{i=1}^{M} 1_{\gamma_i}\theta \right\|_N^2 + \lambda \sum_{i=1}^{M} \|1_{\alpha_i}\theta\|_1 \text{ subject to } C f_D^{part} - \theta = 0. \tag{2.15}$$

We denote the dual variables by $u_i = \begin{bmatrix} u_{\gamma_i} \\ u_{\alpha_i} \end{bmatrix}$ and $\mu = \begin{bmatrix} u_1 \\ \vdots \\ u_{N+m_i-r-1} \\ \nu \end{bmatrix} \in \mathbb{R}^{N+m_i-r+M-2}$. We can write the Lagrangian equation as

$$\mathcal{L}_\rho(f_D^{part}, \theta, \mu) = \left\| y - \sum_{i=1}^{M} 1_{\gamma_i}\theta \right\|_N^2 + \lambda \sum_i \|1_{\alpha_i}\theta\|_1 + \mu^\top (C f_D^{part} - \theta) + \rho \|C f_D^{part} - \theta\|_2^2. \tag{2.16}$$

Note the following equalities:

$$\mu^\top (C f_D^{part} - \theta) = \sum_i^{N+m_i-r-1} u_i^\top (A_i f_{D_i} - z_i) + \sum_i^{M-1} \nu_i^\top (B_i f_{D_i} + B_{i+1} f_{D_{i+1}} - 0) \tag{2.17}$$

$$\|C f_D^{part} - \theta\|_2^2 = \sum_i^{N+m_i-r-1} \|A_i f_{D_i} - z_i\|_2^2 + \sum_i^{M-1} \|B_i f_{D_i} + B_{i+1} f_{D_{i+1}}\|_2^2. \tag{2.18}$$

With the equalities, we are able to separate θ and f_D^{part} into sums of z_i and f_{D_i}. We now

aim to find the iterates for the primal and dual variables:

$$f_{D_i}^{j+1} \leftarrow \underset{f_{D_i}}{\operatorname{argmin}} \frac{\rho}{2}\left(\left\|\Delta_{m_i}^{r+1} f_{D_i} - \alpha_i^j + u_{\alpha i}^j\right\|_2^2 + \left\|\tilde{O}_i f_{D_i} - \gamma_i^j + u_{\gamma i}^j\right\|_2^2 + \left\|B_i f_{D_i} + B_{i+1} f_{D_{i+1}} + \nu_i\right\|_2^2\right) \tag{2.19}$$

$$z_i^{j+1} \leftarrow \underset{z_i}{\operatorname{argmin}} \left\|y - \sum_{i=1}^{M} 1_{\gamma_i} z\right\|_N^2 + \lambda \sum_i \|1_{\alpha_i} z\|_1 + \sum_i \frac{\rho}{2}\left\|z_i - u_i^j - A_i f_{D_i}^{j+1}\right\|_2^2, \tag{2.20}$$

$$u_i^{j+1} \leftarrow u_i^j + A_i f_{D_i}^{j+1} - z_i^{j+1}, \tag{2.21}$$

$$\nu_i^{j+1} \leftarrow \nu_i^j + B_i f_{D_i}^{j+1} + B_{i+1} f_{D_{i+1}}^{j+1}. \tag{2.22}$$

The z_i-update splits into γ_i and α_i updates:

$$\alpha_i^{j+1} \leftarrow \underset{\alpha_i}{\operatorname{argmin}} \|\alpha_i\|_1 + \frac{\rho}{2} \sum_i \left\|\Delta_{m_i}^{(r+1)} f_{D_i}^{j+1} - \alpha_i + u_{\alpha i}^j\right\|_2^2 \text{ and} \tag{2.23}$$

$$\gamma_i^{j+1} \leftarrow \underset{\gamma_i}{\operatorname{argmin}} \left\|y - \sum_{i=1}^{M} \gamma_i\right\|_N^2 + \frac{\rho}{2} \sum_i \left\|\gamma_i - u_{\gamma i}^j - \tilde{O}_i f_{D_i}^{j+1}\right\|_2^2. \tag{2.24}$$

The γ-update can be solved as the following minimization problem:

$$\min_{\gamma_i} \|y - M\bar{\gamma}\|_N^2 + \frac{\rho}{2} \sum_i \|\gamma_i - u_{\gamma i}^j - \tilde{O}_i f_{D_i}^{j+1}\|_2^2 \text{ s.t. } \bar{\gamma} = \frac{1}{M} \sum_i \gamma_i. \tag{2.25}$$

Let $\overline{Of_D} = \frac{1}{M}\sum_i \tilde{O}_i f_{D_i}$. With fixed $\bar{\gamma}$, (2.25) has solution

$$\gamma_i = u_{\gamma i}^j + \tilde{O}_i f_{D_i}^{j+1} + \bar{\gamma} - \bar{u_\gamma}^j - \overline{Of_D}^{j+1}. \tag{2.26}$$

To compute $\bar{\gamma}$, we can solve the unconstrained problem given by

$$\min_{\bar{\gamma}} \frac{1}{2}\|y - M\bar{\gamma}\|_N^2 + \frac{\rho}{2} \sum_i \|\bar{\gamma} - \bar{u_\gamma}^j - \overline{Of_D}^{j+1}\|_2^2. \tag{2.27}$$

Hence,

$$\bar{\gamma}^{j+1} = \frac{1}{M+\rho}(y + \rho(\bar{u_\gamma}^j + \overline{Of_D}^{j+1})). \tag{2.28}$$

Plug γ_i from (2.26) into $u_{\gamma i}^{j+1}$ to get

$$u_{\gamma i}^{j+1} \leftarrow \bar{u_\gamma}^j + \overline{Of_D}^{j+1} - \bar{\gamma}^{j+1}, \tag{2.29}$$

which shows that all γ-dual variables are equal or *in consensus* and can be replaced with a single dual variable $u_\gamma \in \mathbb{R}^N$.

Plugging γ_i^k into the f_{D_i}-update we see that

$$f_{D_i}^{j+1} \leftarrow \underset{f_{D_i}}{\operatorname{argmin}} \frac{\rho}{2}(\left\|\Delta_{m_i}^{r+1} f_{D_i} - \alpha_i^j + u_{\alpha i}^j\right\|_2^2 + \left\|\tilde{O}_i f_{D_i} - \tilde{O}_i f_{D_i}^j + \overline{Of_D}^j - \bar{\gamma}^j + u_\gamma^j\right\|_2^2 \tag{2.30}$$
$$+ \left\|B_i f_{D_i} + B_{i+1} f_{D_{i+1}} + \nu_i\right\|_2^2).$$

Thus, the iterates are given as follows:

$$f_{D_i}^{j+1} \leftarrow \left(\Delta_{m_i}^{(r+1)\top}\Delta_{m_i}^{(r+1)} + \tilde{O}_i^\top \tilde{O}_i\right)^{-1} (\Delta_{m_i}^{(r+1)\top}\left(\alpha_i^j - u_{\alpha i}^j\right) \tag{2.31}$$
$$+ \tilde{O}_i^\top \left(\tilde{O}_i f_{D_i}^j - \overline{Of_D}^j + \bar{\gamma}^j - u_\gamma^j\right) + B_i^\top \left(B_i f_{D_i}^j + \nu_i\right)), \tag{2.32}$$
$$\alpha_i^{j+1} \leftarrow S_{\lambda/\rho}\left(\Delta_{m_i}^{(r+1)} f_{D_i}^{j+1} - u_{\alpha i}^j\right), \tag{2.33}$$
$$u_{\alpha i}^{j+1} \leftarrow u_{\alpha i}^j + \alpha_i^{j+1} - \Delta_{m_i}^{(r+1)} f_{D_i}^{j+1}, \tag{2.34}$$
$$\nu_i^{j+1} \leftarrow \nu_i^j + B f_D^{j+1} + B_{i+1} f_{D_{i+1}}^{j+1} \tag{2.35}$$
$$\bar{\gamma}^{j+1} \leftarrow \frac{1}{M+\rho}\left(y + \rho\left(\bar{u_\gamma}^j + \overline{Of_D}^{j+1}\right)\right), \tag{2.36}$$
$$u_\gamma^{j+1} \leftarrow \bar{u_\gamma}^j + \overline{Of_D}^{j+1} - \bar{\gamma}^{j+1}. \tag{2.37}$$

We distribute each set of $(f_{D_i}^{j+1}, \alpha_i^{j+1}, u_{\gamma i}^{j+1})$ to M processors, then compute ν, $\bar{\gamma}$ and u_γ on a central processor. With enough processors, the M distributed computations could have as

little as $m_i = k + 1$ optimization parameters. In future work, we will be implementing this procedure.

2.5 Simulation Study

In this section, we conduct a simulation study showing how the estimation and approximation error decrease as functions of m, N, r, and k for MBS solutions. We generate a response $y_i = f(x_i)+\epsilon_i$, where $f(z) = e^{\pi z}$, $x_i \in U[0, 1]$ and $\epsilon_i \sim N(0, 1)$ with sample sizes of $N = 40, 80, 120$. For each (x, y)-pair, we solve the ℓ_1 MBS problem using $m = 4, 5, 6, 7, 8, 10, 20, 30, 90$, $r = 0, 1, 2$ and $k = 0, 1, 2$ (via SLP). Note that we maintain $k \leq r$ by solving the following pairings: $r = 0$ and $k = 0$; $r = 1$ and $k = 0, 1$; as well as $r = 2$ and $k = 0, 1, 2$.

When tuning λ, we choose 50 logarithmically spaced values from 10^{-3} to λ_{max}, where

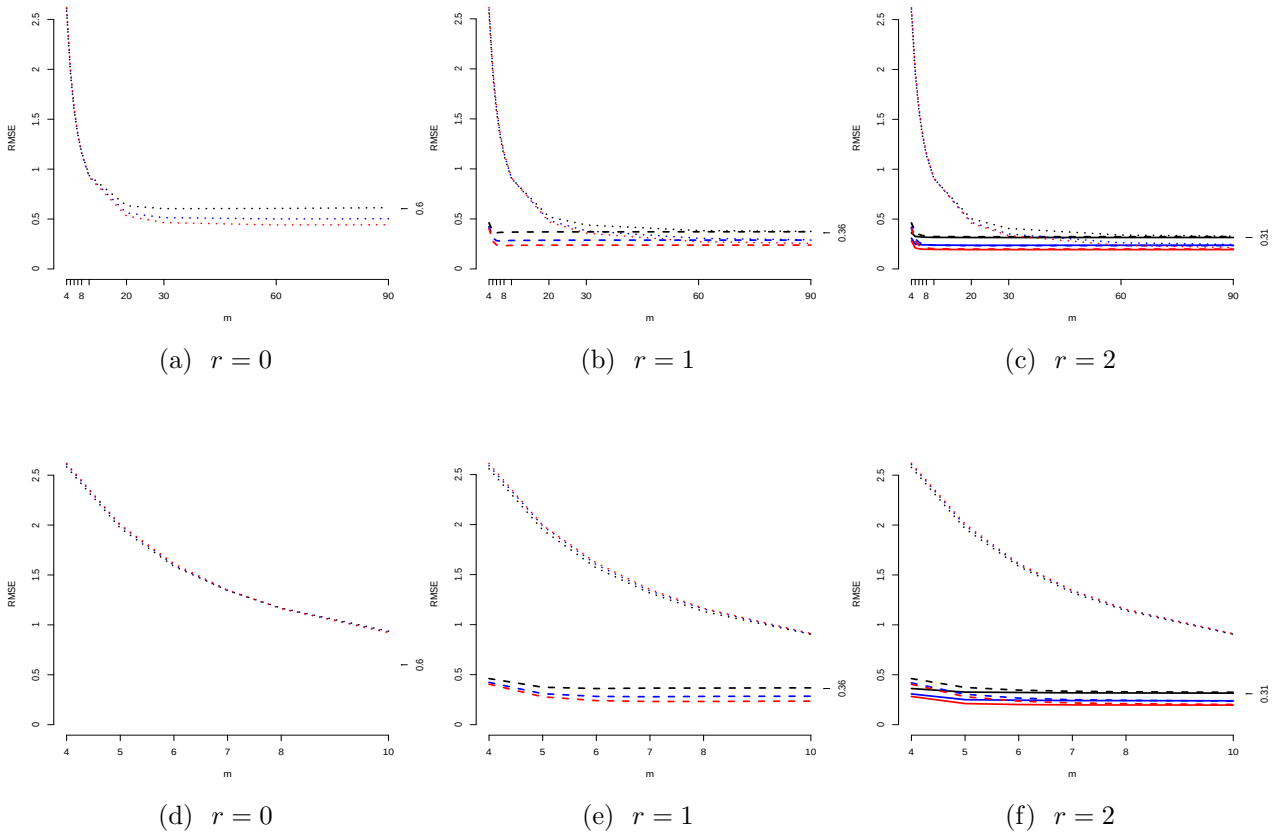
$$\lambda_{max} = \left\| \left(O \left(\Delta_m^{(r+1)} \right)^+ \right)^\top y \right\|_\infty$$

and $A^+ = A^\top (AA^\top)^{-1}$ is the generalized inverse for matrix A. For each (m, r, k)-configuration of MBS, we calculate $RMSE = \left(\sum_{j=1}^{500} MSE_j \right)^{\frac{1}{2}}$ where

$$MSE_j = \sum_{i=1}^{N} \left(\tilde{f}(x_i) - f(x_i) \right)^2$$

and $\tilde{f}$ denotes the MBS estimate.

As expected, when we hold r and k constant, RMSE tends to decrease towards a limit (specific to N) as m increases (Figure 2.2). Holding sample size constant, we see that the limiting RMSE we approach for $r = 0$ (Figure 2.2a) is greater than the limiting RMSE for $r = 1$ (Figure 2.2b), and (though it is difficult to see) the limiting RMSE for $r = 2$ is the smallest of the three values of r. We expect the limiting error to decrease in r since the error should decrease like $N^{-\frac{2r}{2r+1}}$, based on the minimax rate for functions of bounded variation [33]. By choosing to solve (2.5) with $r = 2$, we assume the underlying conditional mean f has bounded derivatives up to $f^{(3)}$. Here, f is an exponential function here with infinitely many



(a) $r = 0$ (b) $r = 1$ (c) $r = 2$

(d) $r = 0$ (e) $r = 1$ (f) $r = 2$

Figure 2.2: Results for 500 simulations over data generated from an exponential function with noise for $N = 40, 80, 120$. MBS models were fit over varying m, r, and k. Line type denotes k: 0 (····), 1 (- - -), and 2 (—). Top row ranges for all m; bottom row ranges for $m \leq 10$.

bounded derivatives, so we could choose a high order r. However, the difference between $N^{-\frac{2r}{2r+1}}$ evaluated at $r = 2$ versus $r = 3$ or $r = 4$ is negligible.

Next, we note that as we interpolate at an order close to our strongest assumption of the smoothness, i.e. $k \to r$, we require smaller m to approach the limiting RMSE. In Figure 2.2c when $r = 2$, the linear ($k = 1$) and quadratic ($k = 2$) interpolator are essentially equally close to the limiting RMSE with modest $m = 6$. This indicates that $k = 1$ with a modest number of mesh points might generally be sufficient for computationally efficient estimation

when $r \geq 1$.

Of course, we have been fitting the MBS solution using uniformly spaced data, which may be giving optimistic results. With non-uniformly spaced data, it is possible we may need many more mesh points to avoid underfitting. We investigate this by repeating the same simulation as above, but for non-uniformly distributed data. The response is generated the same as before, but $y_i = f(\tilde{x}_i) + \epsilon_i$, where $\tilde{x}_i = \log(x_i)$ or $\tilde{x}_i = x_i^2$. We consider only a sample size of $N = 80$. In Table 2.1, we show the results for $r = 2$. Based on Table 2.1, interpolating with $k = 2$ tends to require no more than $m = 10$ to achieve the limiting RMSE using log-spaced data. However, MBS with $k = 1$ tended to require $10 < m < 20$ for a similar limiting RMSE as $k = 2$, but may need $m > 60$ to achieve the exact same RMSE. Using quadratically spaced data ($\tilde{x}_i = x_i^2$), the RMSE for $k = 2$ with $m = 4, 5, 6, 7$ tended to be smaller than $k < 2$, but $k = 1$ hit the same limiting RMSE as $k = 2$ by $m = 8$ (Table 2.1). Overall, with uniformly spaced data (Figure 2.2), we also saw that lower order interpolators required larger m, but the limiting RMSE for $r = 2$ was achieved by $m < 10$ for both $k = 1$ and $k = 2$. With the log-spaced data, we saw that $k < r$ could require potentially much more discretization than $k = r$, although the RMSEs may be close enough in practice. This is evidence that non-uniformly spaced data may require more larger values of m than uniformly spaced data, but still, $m < N$.

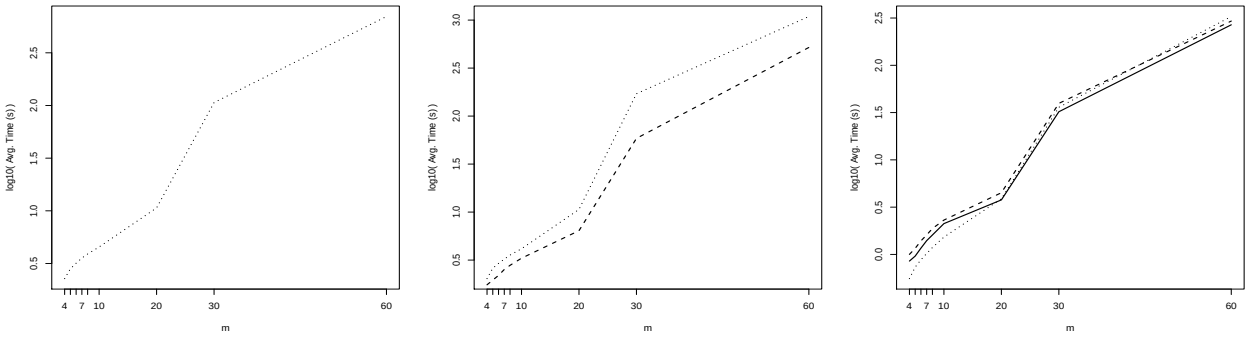
$\tilde{x}_i$	k	$m = 4$	5	6	7	8	10	20	30	60
$log(x_i)$	0	0.318	0.304	0.293	0.280	0.270	0.255	0.230	0.223	0.220
	1	0.243	0.231	0.224	0.220	0.218	0.216	0.214	0.214	0.214
	2	0.227	0.219	0.216	0.213	0.212	0.212	0.212	0.212	0.212
x_i^2	0	2.009	1.531	1.226	1.030	0.888	0.698	0.384	0.308	0.252
	1	0.346	0.271	0.247	0.238	0.234	0.231	0.229	0.229	0.230
	2	0.272	0.240	0.236	0.235	0.234	0.232	0.231	0.231	0.232

Table 2.1: Prediction results for fitting MBS for $r = 2$ with non-uniformly spaced data.

We minimize MBS using an ADMM solver. Unfortunately this algorithm can require

many iterations; however each iteration is quick, running linearly in m. In Figure 2.3, we show how the average time per MBS solution changes as a function of m and k. The average solution time tends to be smaller for greater values of k with fixed r. In particular, for $m \leq 20$, computation time is generally no more than $10^{0.5} = 3.6$ seconds. However that required time increases substantially for $m > 20$. It is important for computational tractability that a good fit can be found without a large number of mesh points.

In the next section, we give theoretical results supporting our observation that MBS can achieve an optimal solution with enough mesh points. We also provide some theory to support our finding that higher order interpolators can reduce the required number of mesh points.



(a) $r = 0$ (b) $r = 1$ (c) $r = 2$

Figure 2.3: Timing results for 500 simulations over a response generated from an exponential function with log-spaced data and noise for $N = 80$. MBS models were fit over varying m, r, and k. Line type denotes k: 0 (····), 1 (- - -), and 2 (—).

2.6 Convergence of MBS for Penalized Regression

In this section, we discuss theoretical properties of the MBS estimate. First, we introduce some notation. We follow this with an outline of the results of this section. We now would like to discuss the MBS estimate as a function (as opposed to just values on a mesh). In

Section 2.2.2, we defined interpolation at a point by $\Omega_{x_i}(f_D)$, which used mesh fitted values f_D to estimate $f(x_i)$. Now, we generalize the interpolator function by $\Omega : \mathbb{R}^m \to \mathcal{F}$ such that for $f_D \in \mathbb{R}^m$, $\Omega(f_D) \in \mathcal{F}$, with $[\Omega(f_D)](x_i) \equiv \Omega_{x_i}(f_D)$. As a reminder, the MBS estimate of f^* is given by $\Omega\left(\tilde{f}_D\right)$ where $\tilde{f}_D$ minimizes the following problem:

$$\tilde{f}_D = \underset{f_D \in \mathbb{R}^m}{\operatorname{argmin}} L_D(f_D), \tag{2.38}$$

where

$$L_D(f_D) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \Omega_{x_i}(f_D)\right)^2 + \lambda_N P_D(f_D).$$

We would like the convergence rates of $\Omega\left(\tilde{f}_D\right)$ to be similar to the convergence rates of the exact solution of a variation problem, $\hat{f}$, given by

$$\hat{f} = \underset{f \in \mathcal{F}_{P,\infty}}{\operatorname{argmin}} L(f) \tag{2.39}$$

where

$$L(f) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - f(x_i)\right)^2 + \lambda_N P(f).$$

We define a gridding function by $D : \mathcal{F} \to \mathbb{R}^m$ such that for $g \in \mathcal{F}$ and $D = \{d_1, \ldots, d_m\}$, $D(g) = (g(d_1), \ldots, g(d_m))^\top$. Finally, for $f \in \mathcal{F}$, we note the empirical $\mathcal{L}_2$-norm by $\|f\|_n^2 = \frac{1}{n}\sum_{i=1}^{n} f^2(x_i)$.

Using these notation conventions, we outline then present our theoretical results. We consider the interpolated solution to the discretized problem, $\Omega\left(\tilde{f}_D\right)$, as an approximate solution to the functional problem given by (2.39). However, MBS is computationally sub-optimal in the sense that

$$L\left(\hat{f}\right) \leq L\left(\Omega\left(\tilde{f}_D\right)\right).$$

We characterize this computational sub-optimality in Lemma 2.6.1. We then show how this optimization sub-optimality affects statistical performance in Theorem 2.6.2. We begin with

the sub-optimality lemma.

Lemma 2.6.1 (Sub-Optimality Inequality) *For all $f \in \mathcal{F}_{P,\infty}$, suppose there exist δ_m and ϵ_m such that*

$$\sup_{f \in \mathcal{F}_{P,\infty}} |P_D(D(f)) - P(f)| \leq \epsilon_m \tag{2.40}$$

and

$$\sup_{f \in \mathcal{F}_{P,\infty}} |\Omega_x(D(f)) - f(x)| \leq \delta_m. \tag{2.41}$$

With $\lambda_N > 0$, we have

$$L\left(\Omega\left(\tilde{f}_D\right)\right) \leq \min \begin{cases} 3L(\hat{f}) + O_P\left(\delta_m^2 \vee \epsilon_m \lambda_N\right) \\ L(\hat{f}) + O_P\left(C\delta_m \vee \epsilon_m \lambda_N\right) \end{cases}. \tag{2.42}$$

Proof See Appendix C.1.

Lemma 2.6.1 shows MBS minimizes the functional problem nearly as well as the exact penalized regression solution depending on the penalty approximation error (2.40) and interpolation error (2.41).

We aim to show that under certain conditions, $\Omega\left(\tilde{f}_D\right)$ is rate optimal. To prove rate optimality, we need to characterize the class $\mathcal{F}$ that we are estimating over. One standard way to do this is entropy. Let $H(\delta, \mathcal{F}, \|\cdot\|_n) = \log N(\delta, \mathcal{F}, \|\cdot\|_n)$ denote the δ-entropy of a function class $\mathcal{F}$ for the $\|\cdot\|_n$-metric, where $N(\delta, \mathcal{F}, \|\cdot\|_n)$ is the number of balls of radius δ needed to cover $\mathcal{F}$, also known as the δ-covering number. We use entropy and δ-entropy interchangeably. Entropy is useful when describing the "size" of a class $\mathcal{F}$: larger entropy implies a more rich class of functions. However, if entropy is too large, then it becomes harder to find the function that minimizes an objective, which implies a slower minimax rate for $\mathcal{F}$. To specify the size of $\mathcal{F}$, we can assume a growth rate in the δ-covering number: with

$0 < \alpha < 2$, we suppose that

$$H\left(\delta, \mathcal{F}_{P,1} := \{f \in \mathcal{F} : P(f) \leq 1\}, \|\cdot\|_n\right) \leq c\delta^{-\alpha}, \tag{2.43}$$

i.e. the δ-covering number grows polynomially in δ. This condition essentially holds for many functional classes, such as functions with bounded total variation ($\alpha = 1$) or in rth order Sobolev spaces ($\alpha = \frac{1}{r}$) [33]. To be more precise, in these problems the entropy condition above only formally holds when we intersect the space with a bounded ball centered around the truth. For ease of exposition, we do not dwell on this issue (though a straightforward modification to our proofs using the Cauchy-Schwartz inequality would rectify this issue).

The same entropy bound holds for the normalized functions when $P(f) + P(f^*) > 0$:

$$H\left(\delta, \left\{\frac{f - f^*}{P(f) + P(f^*)} : f \in \mathcal{F}_{P,1}, P(f) + P(f^*) > 0\right\}, \|\cdot\|_n\right) \leq c_2\delta^{-\alpha}. \tag{2.44}$$

Furthermore, we assume the errors have sub-Gaussian tails:

$$\sup_n \max_{i=1,\ldots,n} K^2\left(\mathbb{E}e^{|\epsilon_i|^2/K^2} - 1\right) \leq \sigma^2. \tag{2.45}$$

Let $(w, f)_N = \frac{1}{N}\sum_{i=1}^N w_i f(x_i)$ denote the empirical inner product between the errors and regression function. By Lemma 8.4 in [33], with $P(f^*) > 0$,

$$\sup_{f \in \mathcal{F}_{P,1}} \frac{|(w, f - f^*)_N|}{\|f - f^*\|_N^{1-\alpha/2}(P(f) + P(f^*))^{\frac{\alpha}{2}}} = O_P(N^{-1/2}). \tag{2.46}$$

(2.46) is important since it relates the empirical process term $(w, f - f^*)_N$ to the complexity of the functions measured by $P(\cdot)$. In Theorem 10.1 of [33], an optimal rate of convergence is established as a consequence of (2.46). For our rate of convergence result, we modify Theorem 10.1.

Using these entropy conditions and empirical process results, we show in the following theorem that the rate of convergence for MBS is off from the minimax rate of the exact

solution of (1.2) by an amount characterized by the optimization sub-optimality.

Theorem 2.6.2 (Rate of Convergence) *Assume* (2.44) *and* (2.45). *Let* $P(f^*) > 0$ *and* $\lambda_N^{-1} = O_p\left(N^{\frac{2}{2+\alpha}}\right)$ *(the minimax rate under* $\mathcal{F}_{P,1}$*). If*

$$L\left(\Omega\left(\tilde{f}_D\right)\right) \leq L(\hat{f}) + \Gamma_{N,m}, \tag{2.47}$$

then we have

$$\left\|\Omega\left(\tilde{f}_D\right) - f^*\right\|_N^2 = O_p\left(\lambda_N + 6\Gamma_{N,m}\right). \tag{2.48}$$

Proof See Appendix C.2.

Based on Lemma 2.6.1, we know that $\Gamma_{N,m} = O_P(\delta_m \vee \epsilon_m \lambda_N)$. Since λ_N is the dominant term in $\lambda_N + \lambda_N \epsilon_m$ with $\epsilon_m \to 0$, we are not concerned about $\epsilon_m \lambda_N$. We need only worry about δ_m, which is the excess error due to interpolation. Ideally, $\lambda_N \geq \delta_m$ so that

$$\left\|\Omega\left(\tilde{f}_D\right) - f^*\right\|_N^2 = O_p\left(\lambda_N\right),$$

i.e. an MBS estimate achieves the minimax rate.

2.6.1 MBS Convergence Rates using Polynomial Interpolators

In the simulations of this chapter, we used a piecewise polynomial interpolator, i.e. SLP, to fit the data. It is not difficult to derive point-wise rates for the interpolation error from an interpolating polynomial of kth degree on a regular m-mesh. Assuming $f^{(k)}(x) < K$ for all $x \in \mathcal{X}$ (see Appendix C.2.1),

$$\left|\Omega_{x_i}\left(D(f)\right) - f(x_i)\right| = O(m^{-(k+1)}). \tag{2.49}$$

Ideally, the additional error from using a mesh is dominated by the error of the penalized regression problem

$$N^{-\frac{2}{2+\alpha}} > m^{-(k+1)}.$$

For the penalized regression problems that we have been approximating, with $P(f) = \int \left|f^{(r)}(x)\right| \partial x$, the entropy conditions are satisified by $\alpha = \frac{1}{r}$. So, we require

$$N^{-\frac{2r}{2r+1}} > m^{-(k+1)}.$$

Fortunately, modestly grown m achieves this property. Conservatively, if $m > N^{\frac{1}{k+1}}$, then $\lambda_N > N^{-1} > m^{-(k+1)}$. In summary, MBS estimates can have negligible error and achieve the minimax rate with a modestly grown mesh.

2.7 Discussion

It can be intractable to solve the exact problem given by (1.2). In this chapter, we have introduced an inexact problem, the MBS objective given by (2.1), whose calculable solutions efficiently approximate (1.2). MBS approximates (1.2) via discretization: We optimize over f_D, the values of our function on a mesh, and use an interpolator and discrete derivative/integral approximations to relate this back to the original problem (1.2). Using the proposed simple piecewise polynomial interpolator, we implemented our solution using an ADMM solver that has a per iteration complexity that is linear in the number of mesh points m.

Other methods that approximate (1.2) include `TF` and `LocARS`. These methods depend on the true solution falling in a finite dimensional subfamily that we can parametrize using a basis set. However, instead of finding the true solution in its finite dimensional subfamily, MBS aims to grow a mesh until we have achieved an efficient estimator of f^*.

A primary goal of our approximation framework is to reduce computational complexity for difficult problems. Through simulation, we showed that $m \geq \sqrt{N}$ can be enough for many

problems with $k = 1$ (linear interpolation) and $r \geq 2$. This allows us to simplify previously solved problems by requiring fewer optimization parameters and decrease computational costs while solving problems with difficult penalties $P(f)$. However, in practice, we may need a strategy to determine a suitable m. A practical strategy may be to train two MBS models of differing mesh size and compare RMSE of each fitted model on the training data. We can estimate $\Omega\left(\tilde{f}_{D_1}\right)$ with $m_1 = \sqrt{N}$ and $\Omega\left(\tilde{f}_{D_2}\right)$ with $m_2 = 2\sqrt{N}$ (each with tuning parameter chosen via cross-validation). If

$$\left| \frac{RMSE\left(\Omega(\tilde{f}_{D_1})\right)}{RMSE\left(\Omega(\tilde{f}_{D_2})\right)} - 1 \right| < \gamma,$$

for some small number γ, then accept $\Omega(\tilde{f}_{D_1})$ as the fitted solution. Otherwise, the mesh can be increased further.

We showed that the MBS solution becomes computationally difficult for large values of m. Using higher order interpolators, the number of mesh points needed for an optimal MBS can be reduced, but problems with particularly large sample sizes may still be difficult to optimize ($m = \sqrt{N}$ is still large for $N > 1,000,000$). It will be important to implement the MBS solution with multi-threading so that computation time is not an obstacle given enough computational resources, as discussed in 2.4.1.

Using entropy and assuming finite $P(f) < \infty$, we were able to prove that under some conditions on the interpolator ($\Omega_{x_i}(f_D)$), the MBS achieves the minimax rate of convergence. In the case of polynomial interpolators, we show that the closer the interpolation order k is to the order of smoothness we penalize r, the fewer the number of mesh points m necessary for our estimate $\Omega\left(\tilde{f}_D\right)$ to be rate-optimal. However, there is no theoretical cost for using too many mesh points, only a computational cost. One issue is that our results in Section 2.6.1 assume that the function class over-which we are optimizing has a uniformly bounded k-th derivative — this is not strictly true in Sobolev classes (only the Sobolev semi-norm of this derivative is bounded). It would be interesting to investigate whether the minimax optimality

of $\Omega\left(\tilde{f}_D\right)$ can be proven without such a condition.

Chapter 3

MESH BASED SOLUTIONS TO MULTIVARIATE PENALIZED REGRESSION

3.1 Introduction

We now consider the learning problem of predicting a response from *multiple* predictors. Suppose we are interested in understanding a nonlinear association between an outcome and some predictors. As a regression problem, we measure a response y_i and a covariate $\boldsymbol{x}_i \in [a,b]^p$, on each of $i = 1, \ldots, N$ observations. As before, we can assume a generative model of the form

$$y_i = f^*(\boldsymbol{x}_i) + w_i,$$

where f^* is an unknown function from a function class $\mathcal{F}$, and w_i are iid errors with $\mathrm{E}[w_i|\boldsymbol{x}_i] = 0$ and $\mathrm{var}[w_i|\boldsymbol{x}_i] = \sigma^2 < \infty$. We aim to find an estimate $\hat{f}$ of f^* based on the observed data. Ideally, we find an easily calculable $\hat{f}$ that is minimax optimal under a flexible class $\mathcal{F}$.

As in the univariate case, we would like to solve over the infinite-dimensional class $\mathcal{F}_{P,\infty}$:

$$\mathcal{F}_{P,\infty} := \{f : [a,b]^p \to \mathbb{R} | P(f) \leq \infty\},$$

where $P(f)$ is some measure of roughness. For example, bivariate thin plate splines solve over $\mathcal{F}_{P,\infty}$ with $P(f)$ defined as follows:

$$P(f) = \int\int \left(\left(\frac{\partial^2 f}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 f}{\partial x_2^2} \right)^2 \right) \partial x_1 \partial x_2.$$

The Sobolev-norm representations of $P(f)$ we considered in Chapter 2 can also be extended

to multivariate data, which is discussed in Section 3.2.1 of this chapter.

We can solve over $\mathcal{F}_{P,\infty}$ using penalized regression ([8], [33]):

$$\hat{f} = \underset{f \in \mathcal{F}_{P,\infty}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - f\left(\boldsymbol{x}_i\right)\right)^2 + \lambda_N P\left(f\right), \tag{3.1}$$

where $\lambda_N \geq 0$ is a tuning parameter. We treat the roughness measure $P(\cdot)$ as a penalty function which penalizes "complexity." As discussed before, with carefully chosen λ_N, we know that $\hat{f}$ converges at the minimax rate under $\mathcal{F}_{P,\infty}$ for many choices of $P(f)$ [21, 33].

It is generally statistically and computationally hard to calculate a good $\hat{f}$ in multivariate scenarios. To estimate a function using multivariate predictors, in general, the sample size, N, needed, even with a minimax optimal estimator, to achieve a fixed MSE increases exponentially as the number of covariates increases — this is known as the curse of dimensionality [3]. Furthermore, before solving over $\mathcal{F} = \mathcal{F}_{P,\infty}$, we must parametrize $\mathcal{F}_{P,\infty}$ often via basis functions, as in the univariate case. However, the basis functions we might consider using for problems such as thin plate splines can have computational challenges such as knot selection.

In Chapter 2, we introduced a computationally tractable framework for approximately solving over $f^* \in \mathcal{F}_{P,\infty}$ with univariate data, which we called MBS. In this chapter, we extend MBS such that we can tractably approximate the solution to the penalized regression problem for multivariate data given by (3.1). The optimization problem in (3.1) is altered slightly just as with univariate MBS. However, we now need to select a mesh of points over the domain of *each* x_i and use the fitted-values on the Cartesian product of all mesh points as our optimization parameters. As before, we replace the penalty function with a finite-difference/Riemann approximation over the mesh points. Finally, the *fitted values at the data points* are approximated by a multivariate interpolation scheme between *fitted values at mesh points*. As with univariate MBS, approximating $P(f)$ using only differences and sums allows us to select a parametrization of $\mathcal{F}$ strictly for computational tractability.

More formally, we consider a finite set of points, or mesh, $D_j \equiv \{d_{1j}, \ldots, d_{m_j j}\} \subset [a, b]$,

such that for a feature $\boldsymbol{x}_{\cdot j}$ the observed $x_{1j}, \ldots, x_{Nj}$ are within the convex hull of D_j. The multivariate mesh is given by $D = D_1 \times \ldots \times D_p$, the Cartesian product of the univariate meshes, yielding $M = \prod_{j=1}^{p} m_j$ many mesh points. Using D, we formulate an approximation to our original problem (3.1):

$$\tilde{f}_D = \underset{f_D \in \mathbb{R}^M}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \Omega_{\boldsymbol{x}_i}(f_D)\right)^2 + \lambda_N P_D\left(f_D\right), \tag{3.2}$$

where f_D denotes the fitted values of the M mesh points. $P_D(f_D)$ is an approximation to $P(f)$; and $\Omega : \mathbb{R}^M \to \mathcal{F}$ is a *multivariate interpolator*. We call $\Omega\left(\tilde{f}_D\right)$ the MBS.

It is perhaps not immediately clear how to interpolate on the multivariate mesh D, much less how to approach the approximation $P_D(f_D)$ of $P(f)$. In Section 3.2, we describe the form we assume for $P(f)$. We construct $P_D(f_D)$ and $\Omega_{\boldsymbol{x}_i}(f_D)$ such that the approximate solution has closeness to the exact solution depending on the total number of mesh points M, while maintaining computational tractability. We state the objective function for multivariate MBS and describe how the ADMM solver we defined in Chapter 2 readily extends to the multivariate problem in Section 3.3.1. In Section 3.4, we run a simulation study demonstrating our solution to the previously unsolved problem of finding solutions with bounded total variation for multivariate data. We highlight that with enough mesh points we can produce minimax-achieving solutions to total variation problems that can computationally and statistically outperform other methods such as thin plate splines. We end with a discussion in Section 3.5 where we note the minimax optimality of a multivariate MBS.

3.2 Multivariate MBS

We begin by introducing the Sobolev-like form we assume for $P(f)$ in the general case, i.e. $p \geq 2$. Suppose for p covariates, we are interested in p orders of differences given in the multi-index $\boldsymbol{r} = (r_1, \ldots, r_p)$. Let $\mathcal{D}^{\boldsymbol{r}} f = \frac{\partial^{|\boldsymbol{r}|} f}{\partial x_1^{r_1} \ldots \partial x_p^{r_p}}$ denote the partial derivative, where $|\boldsymbol{r}| = \sum_{i=1}^{p} r_i$. In general, we may be interested in collections of partial derivatives, i.e. $\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_S\}$. We assume that our penalty takes the form of a Sobolev-like semi-norm of the

following form:

$$P(f) = \|f\|_\ell^\ell = \begin{cases} \sum_{s=1}^{S} \|\mathcal{D}^{\boldsymbol{r}_s} f\|_\ell^\ell & 1 \le \ell < +\infty \\ \sup_s \|\mathcal{D}^{\boldsymbol{r}_s} f\|_\infty & \ell = \infty \end{cases}. \tag{3.3}$$

For example, with $p = 2$ and $\ell = 1$, the collection of first order differences, $\{\boldsymbol{r}_1 = (1,1), \boldsymbol{r}_2 = (1,0), \boldsymbol{r}_3 = (0,1)\}$, specifies the penalty for estimating a function of bounded total variation [33] (essentially a bivariate fused lasso if the data fall on a grid). In Section 3.4, we discuss our approach to estimating solutions with bounded total variation for any $p > 1$.

For our approximation, we introduce some notation. Suppose we have p covariates each with a regular mesh of sizes denoted in the vector $\boldsymbol{m} = (m_1, \ldots, m_p)$. We indicate the bin widths for each of the meshes by $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_p)$, where $\delta_j = d_{i+1,j} - d_{i,j}$ for any i. Furthermore, we will assume in this section that

$$\mathbb{R}^{\boldsymbol{m}} \equiv \mathbb{R}^{m_1} \times \ldots \times \mathbb{R}^{m_p}.$$

The functional values on the grid are denoted by the p-tensor $f_D \in \mathbb{R}^{\boldsymbol{m}}$. Let $(f_D)_{\boldsymbol{i}} = f(d_{\boldsymbol{i}}) = f(d_{i_1,\ldots,i_p})$, where $\boldsymbol{i} = (i_1, \ldots, i_p)$. Furthermore, we denote the unit vectors of length p by $\boldsymbol{e}_j$ for $j = 1, \ldots, p$, where

$$\boldsymbol{e}_j = (0, \ldots \overbrace{, 1,}^{j-th} 0, \ldots, 0).$$

3.2.1 Riemann Approximations to Sobolev-like Norms on a Bivariate Mesh

Recall the univariate normalized first order difference function for an m-mesh $\Delta_m^1 : \mathbb{R}^m \to \mathbb{R}^{m-1}$ defined in the previous section. We generalize Δ_m^1 (via an extra index) so that we have a normalized first order difference function for the jth covariate and any pair of indices such that $\Delta_{\boldsymbol{m},j}^1 : \mathbb{R}^{\boldsymbol{m}} \to \mathbb{R}^{\boldsymbol{m}-\boldsymbol{e}_j}$ and

$$\left[\Delta_{\boldsymbol{m},j}^1 f_D\right]_{\boldsymbol{i}} = \frac{f(d_{\boldsymbol{i}+\boldsymbol{e}_j}) - f(d_{\boldsymbol{i}})}{\delta_j}.$$

For any pair of indices, we define the rth order normalized difference operator $\Delta^r_{\boldsymbol{m},j} : \mathbb{R}^{\boldsymbol{m}} \to \mathbb{R}^{\boldsymbol{m}-r\boldsymbol{e}_j}$ by the recursive formula

$$\left[\Delta^r_{\boldsymbol{m},j} f_D\right]_{\boldsymbol{i}} = \left[\Delta^{r-1}_{\boldsymbol{m}-\boldsymbol{e}_j,j}\left[\Delta^1_{\boldsymbol{m},j} f_D\right]\right]_{\boldsymbol{i}},$$

where $i_j = 1, \ldots, m_j - 1$ and $i_r = 1, \ldots, m_r$ for $r = 1, \ldots, p$ $(r \neq j)$.

With the generalized first order difference, we approximate $\mathcal{D}^{\boldsymbol{r}} f$ by

$$\Delta^{\boldsymbol{r}}_{\boldsymbol{m}} f_D = \Delta^{r_1}_{\boldsymbol{m}_1,1} \Delta^{r_2}_{\boldsymbol{m}_2,2} \ldots \Delta^{r_p}_{\boldsymbol{m}_p,p} f_D,$$

where $\boldsymbol{m}_p = \boldsymbol{m}$ and $\boldsymbol{m}_{j-1} = \boldsymbol{m}_j - r_j \boldsymbol{e}_j$. Thus, our $\boldsymbol{r} = (r_1, \ldots, r_p)$-order Riemann approximation of $P(f) = \|f\|^\ell_\ell$ using regular meshes for each covariate is given by

$$P_D(f_D) = \sum_{\boldsymbol{i} \preceq \boldsymbol{m}-\boldsymbol{k}} |(\delta_1 \delta_2 \ldots \delta_p)^{1/\ell} [\Delta^{\boldsymbol{r}}_{\boldsymbol{m}} f_D]_{\boldsymbol{i}}|^\ell \tag{3.4}$$

$$= \sum_{\boldsymbol{i} \preceq \boldsymbol{m}-\boldsymbol{k}} |(\delta_1 \delta_2 \ldots \delta_p)^{1/\ell} [\Delta^{r_1}_{\boldsymbol{m}_1,1} \Delta^{r_2}_{\boldsymbol{m}_2,2} \ldots \Delta^{r_p}_{\boldsymbol{m}_p,p} f_D]_{\boldsymbol{i}}|^\ell, \tag{3.5}$$

where $\boldsymbol{i} \preceq \boldsymbol{m} - \boldsymbol{r} = \{i_1 \leq m_1 - r_1, \ldots, i_p - r_p\}$. For a collection of partials $\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_S\}$, we use the following approximation:

$$P_D(f_D) = \sum_{s=1}^{S} \sum_{\boldsymbol{i} \dot{\preceq} \boldsymbol{m}-\boldsymbol{r}_s} |(\delta_1 \delta_2 \ldots \delta_p)^{1/\ell} [\Delta^{\boldsymbol{r}_s}_{\boldsymbol{m}} f_D]_{\boldsymbol{i}}|^\ell$$

As can be seen, the Riemann approximation extends in a straightforward manner from the univariate case to the multivariate case. In Appendix A, we present a matrix notation for both the univariate and bivariate MBS problems.

3.2.2 Multivariate Interpolation:

We now describe our approach to multivariate interpolation from a mesh at a sample point $\boldsymbol{x} = (x_1, \ldots, x_p)$. In Chapter 2, we described the "simple piecewise polynomial interpolator,"

which defines a k degree polynomial interpolation at a point using only $k+1$ local points. In this section, we aim to extend the SPPI to the multivariate case. First, we modify some of the notation previously defined in the section on the univariate SPPI, Chapter 2 Section 2.2.2. Next, we discuss multivariate polynomial interpolation in a general framework. Then we introduce the *multivariate simple piecewise polynomial interpolator*.

Recall the previously defined $\tilde{\boldsymbol{d}}$, which denoted in the univariate case the neighborhood of $k+1$ mesh values about a sample point. In parallel, $N^{\boldsymbol{x}}$ denotes a neighborhood of the mesh surrounding $\boldsymbol{x}$. For an order R interpolation, $N^{\boldsymbol{x}}$ contains the $L = \binom{k+p}{p}$ nearest mesh elements, e.g. $N^{\boldsymbol{x}} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_L\}$. We denote the fitted values for the mesh points used in the interpolation as $\theta_{N^{\boldsymbol{x}}} = (\theta_1, \ldots, \theta_L)$.

Next, we discuss multivariate polynomial interpolation then introduce the multivariate SPPI. Suppose we want to interpolate at a point $\boldsymbol{x} = (x_1, \ldots, x_p)$ via an kth order polynomial: the kth order polynomial in general form is given by

$$f_k(\boldsymbol{x}) = \beta_0 + \sum_{j \leq p} \beta_j x_j + \sum_{j_1 \leq j_2 \leq p} \beta_{j_1,j_2} x_{j_1} x_{j_2} + \ldots + \sum_{j_1 \leq \ldots j_k \leq p} \beta_{j_1,\ldots,j_k} x_{j_1} \cdots x_{j_k}.$$

For the kth order polynomial in p dimensions, we have $T = \left[1 + p + \left(\binom{p}{2} + p\right) + \ldots\right]$ total parameters contained in $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p, \ldots)^\top \in \mathbb{R}^T$. Using basis elements as in Chapter 2, we can write $f_R(\mathbf{x})$ as

$$f_k(\mathbf{x}) = \boldsymbol{\beta}^\top \left[\psi_1(\mathbf{x}), \ldots, \psi_T(\mathbf{x})\right]^\top,$$

where

$$\begin{aligned}
\psi_1(\mathbf{x}) = 1, \psi_2(\mathbf{x}) = x_1, \ldots, \psi_{p+1}(\mathbf{x}) = x_p, \\
\psi_{p+2}(\mathbf{x}) = x_1^2, \psi_{p+3}(\mathbf{x}) = x_1 x_2 \ldots, \psi_{2p+2}(\mathbf{x}) = x_1 x_p \\
\psi_{2p+3}(\mathbf{x}) = x_2^2, \ldots\ldots \psi_T(\mathbf{x}) = x_p^k.
\end{aligned}$$

For a point $\boldsymbol{x}$, we form a system of linear equations to find the coefficients $\boldsymbol{\beta}$. Using

$\{\boldsymbol{d}_i\}_{i=1,\ldots,L} \in N^{\boldsymbol{x}}$, we get the following system of linear equations:

$$\Psi\boldsymbol{\beta} = \tilde{\theta}$$

with $\tilde{\theta} = [\theta_1, \ldots, \theta_L]$ (our approximation of $\theta_{N^{\boldsymbol{x}}}$), and $\Psi_{ij} = \psi_j(\mathbf{d}_i)$ ($j = 1, \ldots, T$ and $i = 1, \ldots, L$). The coefficients are given by $\boldsymbol{\beta} = \Psi^{-1}\tilde{\theta}$. At a new sample point, $\mathbf{x}_{new}$, in that region, we interpolate with

$$\begin{aligned}\Omega_{\mathbf{x}_{new}}(f_D) &= \left([\psi_1(\mathbf{x}_{new}), \ldots, \psi_L(\mathbf{x}_{new})]\, \Psi^{-1}\right)\tilde{\theta} \\ &= \boldsymbol{a}^\top\tilde{\theta},\end{aligned}$$

where $\boldsymbol{a}$ denote the weights for a linear combination.

We have described multivariate interpolation over L-interpolants at a point of interest $\mathbf{x}$. We would like a linear operator similar to the univariate case, i.e. $\Omega_{\mathbf{x}}(f_D) = Of_D$. In the univariate case, the $L = k+1$ nearest mesh points made a neighborhood of $k+1$-consecutive points about an observed data value x_i, making the interpolation matrix O banded. However, in this case, the $L = 3$ points will not be consecutive in one direction. For example, in a bivariate scenario with $\mathbf{m} = (4, 4)$, suppose we have

$$f_D = \begin{pmatrix} f(d_{1,1}) & f(d_{1,2}) & f(d_{1,3}) & f(d_{1,4}) \\ f(d_{2,1}) & f(d_{2,2}) & f(d_{2,3}) & f(d_{2,4}) \\ f(d_{3,1}) & f(d_{3,2}) & f(d_{3,3}) & f(d_{3,4}) \\ f(d_{4,1}) & f(d_{4,2}) & f(d_{4,3}) & f(d_{4,4}) \end{pmatrix},$$

The L interpolants to an observation $\mathbf{x}_i$ could be $(f(d_{1,1}), f(d_{1,2}), f(d_{2,2}))$. We can define an

observation specific interpolation matrix O_i such that

$$O_i = \begin{pmatrix} a_{1,1} & a_{1,2} & 0 & 0 \\ 0 & a_{2,2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

where $\mathbf{a} = (a_{1,1}, a_{1,2}, a_{2,2})$ are weights determined as previously described. Using O_i, we can describe an interpolation using the inner product $\langle \cdot \rangle$:

$$\begin{aligned} \Omega_{\mathbf{x}_i}(f_D) &= \langle O_i, f_D \rangle && (3.6) \\ &= \mathbf{tr}(O_i^\top f_D) && (3.7) \\ &= a_1 f_1 + a_2 f_2 + a_3 f_6. && (3.8) \end{aligned}$$

Alternatively, we could define $\vec{f_D}$ as the stacking of the rows of f_D into a single column, i.e.

$$\vec{f_D} = (f(d_{1,1}), \ldots, f(d_{1,4}), f(d_{2,1}), \ldots, f(d_{3,1}), \ldots, f(d_{4,1}), \ldots, f(d_{4,4}))^\top . \quad (3.9)$$

In turn, we could define a vector $\vec{o_i}$ as the stacking of the rows of O_i, i.e

$$\vec{o_i} = (a_{1,1}, a_{1,2}, 0, 0, 0, a_{2,2}, 0, \ldots, 0)^\top .$$

Using this notation, we arrive at an interpolation matrix O, i.e

$$O = (\vec{o_1}, \vec{o_2}, \ldots, \vec{o_n})^\top .$$

For multivariate data $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p)^\top \in \mathbb{R}^{n \times p}$, we define $\Omega_X(f_D)$ as the interpolation of the observed multivariate data using the p-tensor $f_D \in \mathbb{R}^{\boldsymbol{m}}$, i.e.

$$\Omega_X(f_D) = O\vec{f_D}. \quad (3.10)$$

Note that (3.10) applies for any p. When the dimensions of the tensor f_D grow with m and p, $\vec{f_D}$ will grow in length (because of the stacking).

Here, we discuss the relationship between the kth order of interpolation and the $\boldsymbol{r}$th order penalty we have assumed is bounded. Recall that in univariate MBS, we required $k \leq r$, which guarded us against using an interpolator that assumed more smoothness than we assumed by $P(f)$. However, univariate penalized regression as we have shown it only requires considering a single order of smoothness. In multivariate MBS, the collections of partials $\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_S\}$ can designate both isotropic (same order) and anisotropic (mixed order) partial derivatives. For example, in a bivariate setting, the collection of first partials contains the isotropic (or pure) partial, $\boldsymbol{r}_1 = (1, 1)$, and the anisotropic (or mixed) partials $\boldsymbol{r}_2 = (1, 0)$ and $\boldsymbol{r}_3 = (0, 1)$. Here, we do not want to interpolate with $k > 1$, since we are assuming at most first order smoothness in *both* predictors by $P(f)$. Furthermore, since we approximate the penalty (via differences/sums) independently of the interpolation, we can choose $k < 1$ or $k = 0$. Let $\{\boldsymbol{r}'_1 = (1, 1), \ldots, \boldsymbol{r}'_{S'} = (S', S')\}$ denote the collection of orders for isotropic partials such that $1 \leq 2 \ldots \leq S'$ for integer $S' \geq 1$. Thus, intuitively, the rule we follow with multivariate MBS is $k \leq S'$.

3.3 The Multivariate MBS Objective

Suppose we observe response $y_i = f(\boldsymbol{x}_i) + w_i$ for multivariate predictors $\boldsymbol{x}_i \in \mathbb{R}^p$ $(i = 1, \ldots, N)$ with $w_i \sim (0, \sigma^2)$. The $\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_S\}$-order MBS with kth-order interpolation estimates $\tilde{f_D} = (f(d_1), \ldots, f(d_m))^\top$ on a regular m-mesh are given by

$$\min_{f_D \in \mathbb{R}^{\boldsymbol{m}}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \langle O_i, f_D \rangle\right)^2 + \lambda \sum_{s=1}^{S} \sum_{\boldsymbol{i} \dot{\leq} \boldsymbol{m} - \boldsymbol{r}_s} |(\delta_1 \delta_2 \ldots \delta_p)^{1/\ell} [\Delta_{\boldsymbol{m}}^{\boldsymbol{r}_s} f_D]_{\boldsymbol{i}}|^\ell. \tag{3.11}$$

O_i is the kth-order interpolation matrix specific to an observation $\boldsymbol{x}_i$ as described in the previous subsection.

Often, it will be useful to use O and $\vec{f_D}$, the stacked versions of the interpolation matrices

$O_1, \ldots, O_n$ and the p-tensor f_D. We can rewrite the problem in (3.11) as

$$\min_{f_D \in \mathbb{R}^{\boldsymbol{m}}} \left\| y - O\vec{f_D} \right\|_N^2 + \lambda \sum_{s=1}^{S} \sum_{\boldsymbol{i} \leq \boldsymbol{m} - \boldsymbol{r}_s} |(\delta_1 \delta_2 \ldots \delta_p)^{1/\ell} [\Delta_{\boldsymbol{m}}^{\boldsymbol{r}_s} \vec{f_D}]_{\boldsymbol{i}}|^{\ell} \tag{3.12}$$

or (by absorbing the constants into λ)

$$\min_{f_D \in \mathbb{R}^{\boldsymbol{m}}} \left\| y - O\vec{f_D} \right\|_N^2 + \lambda \|\mathcal{D}\vec{f_D}\|_1, \tag{3.13}$$

for carefully constructed difference operator $\mathcal{D}$. Our fitted values for X are given by $\tilde{f} = O\hat{\vec{f}_D}$.

3.3.1 Solving Multivariate MBS

With the MBS objective given in (3.13), we see that we can immediately apply the same convex solver, the alternating direction method of multipliers (ADMM), as in Chapter 2. The ADMM updates themselves do not change in form:

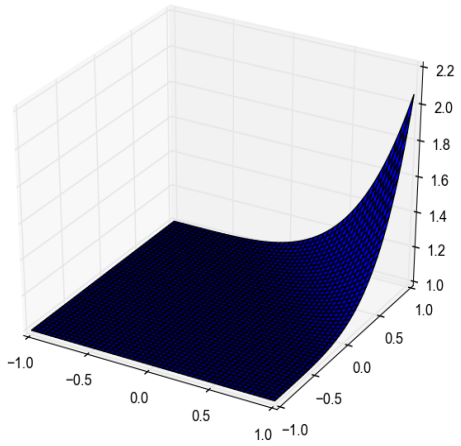
$$f_D^{j+1} \leftarrow \left(O^\top O + \rho(\Delta_m^{(r+1)})^\top \Delta_m^{(r+1)}\right)^{-1} \left(O^\top y + \rho(\Delta_m^{(r+1)})^\top (\alpha^j + u^j)\right), \tag{3.14}$$

$$\alpha^{j+1} \leftarrow S_{\lambda/\rho}\left(\Delta_m^{(r+1)} f_D^{j+1} - u^j\right), \tag{3.15}$$

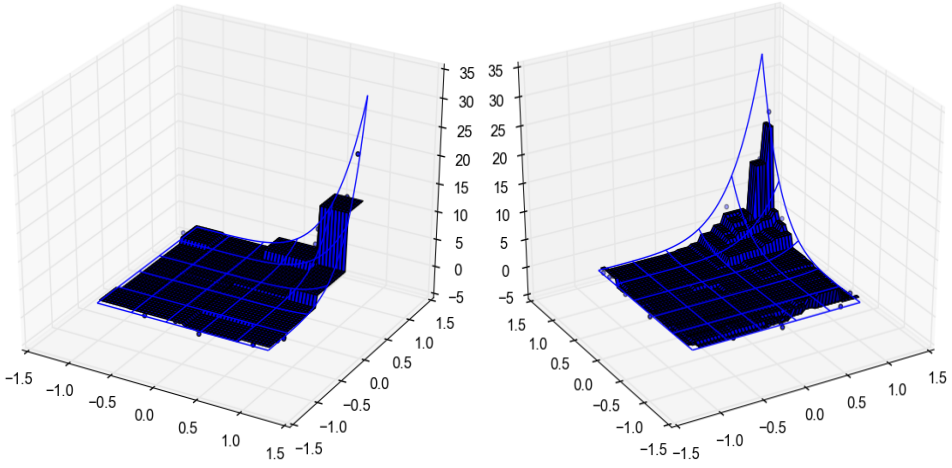
$$u^{j+1} \leftarrow u^j + \alpha^{j+1} - \Delta_m^{(r+1)} f_D^{j+1}. \tag{3.16}$$

All details of the ADMM are the same as shown in Chapter 2 Section 2.4. However, the computational complexity of these iterates has changed from the univariate problem. The simple piecewise polynomial in the univariate case permitted banded and sparse interpolation matrix O. When using the kth order multivariate local polynomial that we have defined, we require the $L = \binom{k+p}{p}$ nearest points, which will not be consecutive in a given row of O. Although O is now not banded, it is still a sparse matrix. Similarly, the difference matrix $\mathcal{D}$ used in the penalty approximation is still sparse. `C++` and `Python` can be utilized to efficiently operate over sparse matrices.

3.4 Simulation Study



(a) True f



(b) $\boldsymbol{m} = (4, 4)$ (c) $\boldsymbol{m} = (15, 15)$

Figure 3.1: Comparison of MBS estimates fit on simulated multivariate data with underlying conditional mean given by bivariate exponential function. We observe $N = 100$ noisy observations (transparent) of a bivariate exponential function shown in (a) and wireframe. We draw MBS fits (blue) using $\{\boldsymbol{r}_1 = (1, 1), \boldsymbol{r}_2 = (1, 0), \boldsymbol{r}_3 = (0, 1)\}$ and $k = 0$.

In the multivariate setting, MBS approximates problems previously thought to be intractable. Consider the difficult total variation problem. For p covariates, $(x_1, \ldots, x_p)$, we define a

multi-index of integers $r = (r_1, \ldots, r_p)$, where $r_j \in \{0, 1\}$ and $|r| = \sum_{j=1}^{p} r_j$. In the MBS framework, we approximate the penalty

$$P(f) = \sum_{r \in \{0,1\}^p} \int \left| \frac{\partial^{|r|}}{\partial^{r_1} x_1 \ldots \partial^{r_p} x_p} f(x) \right| \partial x_1 \ldots \partial x_p,$$

where $\{0, 1\}^p \equiv \{0, 1\} \times \ldots \times \{0, 1\}$, i.e. the Cartesian product. In the bivariate setting, our penalty parameter becomes

$$P(f) = \int \left| \frac{\partial}{\partial x_1} f(x_1, x_2) \right| \partial x_1 \partial x_2 + \int \left| \frac{\partial}{\partial x_2} f(x_1, x_2) \right| \partial x_1 \partial x_2 + \int \left| \frac{\partial}{\partial x_1 \partial x_2} f(x_1, x_2) \right| \partial x_1 \partial x_2.$$

When $P(f)$ is finite, solutions have bounded total variation. [33] derived uniform convergence rates for solutions with bounded total variation: showing that the error of the total variation solution decreases like $N^{-\frac{1+p}{2+4p}}$. We created an `R` package that uses our `C++` implementation, **MTV**, to find solutions with bounded total variation in p dimension. **MTV** uses MBS by approximating $P(f)$ using differences across a mesh and piecewise constant interpolation. For details, see Appendix E.

In Figure 3.1, we show MBS fits approximating the bivariate fused lasso based on noisy data generated from the exponential function shown in Figure 3.1a. MBS is fit to the exponential curve using $m = 15$, i.e. a 15 by 15 mesh on the predictor space. The fit is piecewise-constant and at the level of discretization shown, it is not clear how well we are approximating the bivariate exponential function. In Figure 3.2, we show results for a simulation study using much larger sample sizes and a wide range of m. For this smooth exponential function, using as small as $m = 40$ or $M = 1,600$ begins to produce near-optimal fits for large problems such as $N = 10,000$.

As we saw in Figure 3.2, the total variation solution is piecewise constant, which may not be appealing. Often, when we think of fitting complex functions or potentially non-linear functions, we think of using wavy functions. Intuitively, we might think that total variation with it's piecewise-constant fits will be best suited for fitting functions that are piecewise

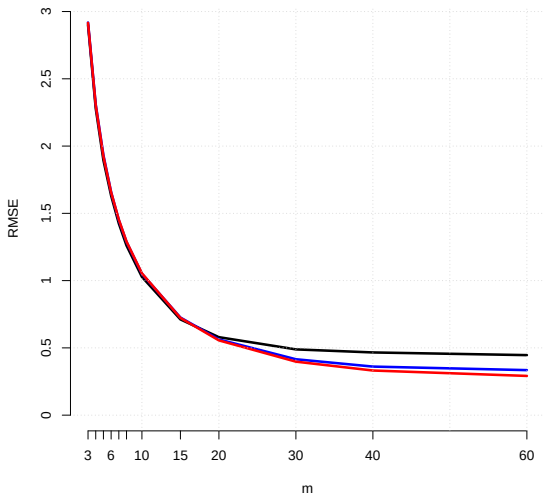
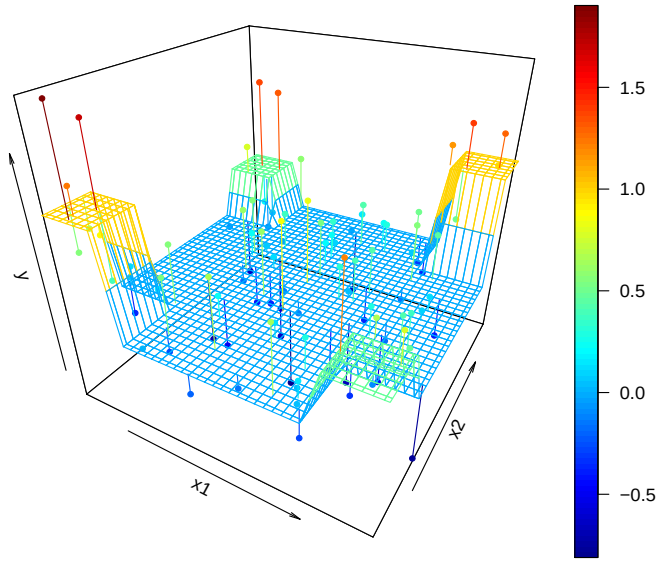


Figure 3.2: Comparison of MBS estimates fit on data with underlying conditional mean given by bivariate exponential function. We simulated $N = 1,000, 5,000, 10,000$ noisy observations of a bivariate exponential function shown in Figure 3.1. For each of 500 simulations, we approximated the bivariate fused lasso using MBS over a range of m.

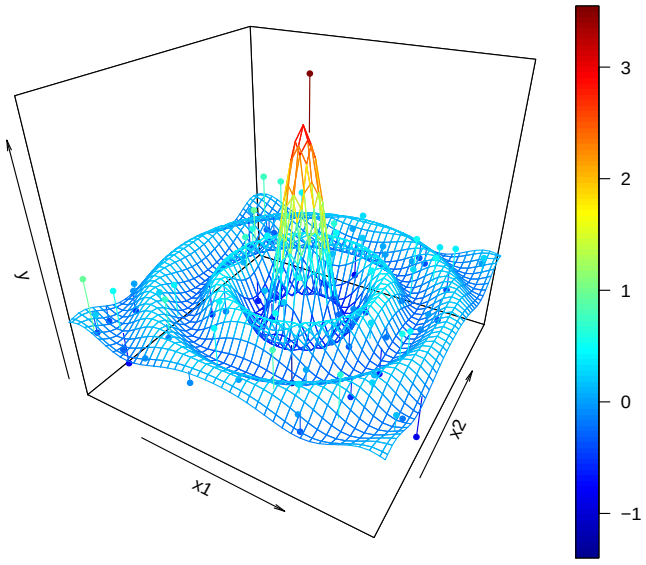
constant, such as the Tower function in Figure 3.3a. Furthermore, we might think that total variation will not accurately fit functions more wiggly than the exponential function we have shown, such as the Sombrero function in Figure 3.3b. In the more wiggly case, we might prefer something like the thin plate spline solution (`TPS`), which extends the smoothing spline problem into multivariate space [11, 36].

We conduct an experiment to investigate how well the total variation solution fits the true underlying function compared to `TPS`, which produces smooth fitted functions. In a simulation, we generated $N = 100$ noisy observations from both the Tower and Sombrero function and fit the data using MBS for total variation and `TPS` over 500 iterations. MBS was fit using a regular mesh on each covariate using $m = 5, 10, 15, 20, 40, 60, 80$ mesh points per covariate to see how the fit changes as the mesh becomes more fine. The tuning parameter λ was selected as in the univariate problems of Chapter 2: we calculate a λ_{max}, choose 100 values between 0.001 and λ_{max}, then perform cross-validation (10-fold here). For each MBS and `TPS`

fit, we calculate the mean squared error (MSE). We calculate the median of the 500 MSEs and evaluate the results of the simulations using the ratio of those medians, i.e. *Relative Median MSE* $= \frac{MedianMSE(MBS)}{MedianMSE(TPS)}$.



(a) Towers



(b) Sombrero

Figure 3.3: Plotting the Tower and Sombrero functions. We observe $N = 100$ noisy observations (points) from a set of towers (a) and a sombrero (b) in bivariate space.

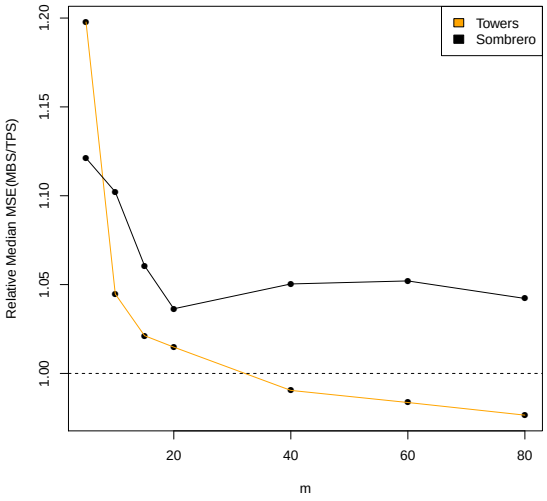


Figure 3.4: Comparing MBS and thin plate spline estimates of tower and sombrero functions. We simulated $N = 100$ noisy observations from bivariate functions shown in Figure 3.3. For each of 500 simulations, we approximated the bivariate fused lasso using MBS over a range of m and fit `TPS` using the `fields` package in `R`. We plot the *Relative RMSE* $= \frac{RMSE(MBS)}{RMSE(TPS)}$.

In Figure 3.4, we see that MBS tends to have smaller MSE than `TPS` by $m \geq 40$ when fitting the Towers function. Figure 3.4 suggests that it may be possible to improve our fit of the Towers function further with a larger m. However, for the Sombrero function, our simulation suggests `TPS` tends to have smaller MSE than MBS. There is some non-monotonicity in the trend for the relative MMSE for the Sombrero function. For MBS, we might expect monotonic relative MMSE, but there can be irregularity due to placement of the mesh points, so it is not surprising to see non-monotonically decreasing relative MMSE. Overall, the MBS fits have similar error to the TPS fits: by $m \geq 40$, MBS tends to produce fits with MSE within 5% of the MSE for TPS fits. For larger sample sizes, $N > 500$, the `TPS` required an enormous amount of computational time, so we do not include a comparison. In Figure 3.5, we show results for MBS total variation solutions for the bivariate functions using $N = 1,000$. Although fitting total variation solutions may not be aesthetically appealing, these solutions achieve the minimax rate over functions of bounded variation and can compete

with existing methodology for much less computational costs.

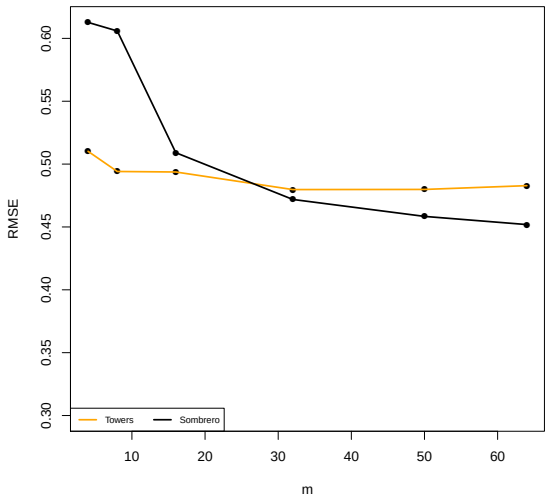


Figure 3.5: Simulation results of fitting the tower and sombrero functions with a large sample size. We simulated $N = 1,000$ noisy observations from bivariate functions shown in Figure 3.3. For each of 500 simulations, we approximated the bivariate fused lasso using MBS over a range of m.

3.5 Discussion

In this chapter, we have shown the extension of MBS for multivariate problems for a broad class of variational problems controlled by $P(f)$. Our approximation $P_D(f_D)$ to $P(f)$ is easily calculable using only differences of adjacent mesh points. The multivariate local polynomial we propose requires little computation by only requiring a fraction of the mesh fitted points to fit an observed point. Thus, we have defined $P_D(f_D)$ and $\Omega_{\boldsymbol{x}_i}(f_D)$ to be inherently sparse. Consequently, the ADMM algorithm can solve quickly utilizing both convexity and sparsity of the MBS objective.

Thus, we can now solve problems that were previously intractable. We have implemented our solution to total variation problem for multivariate predictors in an `R` package called **MTV**. The vignette is given in Appendix E. The total variation solution will be

useful in problems where we need to account for variation with special structure. In disease mapping, for example, to estimate disease risk over a geographic region we often assume spatial structure, since it is difficult to collect all information relevant to disease variation. In the disease mapping literature, Bayesian hierarchical modeling approaches are common, which can be computationally intense. Furthermore, their accuracy in fitting disease risk depends on well-specified models for difficult problems such as those that require locally-adaptive behavior when modeling a rare outcome [13, 22, 5]. As more complex models are specified, more optimization parameters are introduced usually increasing the complexity of the Markov chain Monte Carlo estimation procedure used in these problems. Although the total variation solution we have proposed also is computationally intense for even moderate p, it can be useful for these kinds of problems, since it is locally-adaptive and requires no more optimization parameters than the size of the mesh.

In Chapter 2, we proved minimax optimality of MBS for $\mathcal{F}$ characterized by entropy. For higher dimension problems with $p > 1$, the theory is essential the same, but the entropy increases which in turn results in slower minimax rates. For example, we discussed in Chapter 2 that for univariate regression functions of bounded variation the entropy bound (2.43) is satisfied with $\alpha = 1$, which leads to the rate $\|\hat{f} - f^*\|_N = O_P(N^{-\frac{1}{3}})$. [33] shows that for functions of bounded variation in $\mathbb{R}^2$, (2.43) is satisfied with $\alpha = \frac{4}{3}$ and that $\|\hat{f} - f^*\|_n = O_P(N^{-\frac{3}{10}})$. For functions of bounded variation in $\mathbb{R}^p$, the rate becomes $O_P(N^{-\frac{1+p}{2+4p}})$ — this is the minimax rate over that class. The same proof techniques used in Chapter 2 show that our mesh-based multivariate estimator attains the minimax rate over the corresponding non-parametric multivariate class (given that the selected mesh is sufficiently fine, as a function of N and our interpolator). While our method requires a potentially large number of optimization variables to use in the mesh ($\sim N^{p/2}$), it can leverage sparsity, and in many applications provides more computational tractable estimators than other alternatives (such as thin plate splines).

Chapter 4

EXTENSIONS OF MESH BASED SOLUTIONS

4.1 Introduction

By this point, we have explored our approach to the prediction problem that we intended to solve in Chapters 2 and 3. Statistically, we observe an outcome, $y_i \in \mathbb{R}$, and set of predictors, $\boldsymbol{x}_i \in \mathbb{R}^p$, generated from

$$y_i = f^*(\boldsymbol{x}_i) + w_i, \tag{4.1}$$

where f^* is an unknown function from a function class $\mathcal{F}$, and w_i are iid errors with $\mathrm{E}\left[w_i|\boldsymbol{x}_i\right] = 0$ and $\mathrm{var}\left[w_i|\boldsymbol{x}_i\right] = \sigma^2 < \infty$. We assume f^* lives in an infinite dimensional function class such that $P(f^*) < \infty$, i.e. $f^* \in \mathcal{F}_{P,\infty}$, where

$$\mathcal{F}_{P,\infty} := \{f : \mathbb{R}^p \to \mathbb{R} | P(f) \leq \infty\}$$

and $P(\cdot)$ is some measure of roughness. It is common to estimate f^* using penalized regression by solving (3.1):

$$\hat{f} = \underset{f \in \mathcal{F}_{P,\infty}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - f\left(\boldsymbol{x}_i\right)\right)^2 + \lambda_N P\left(f\right),$$

where $\lambda_N \geq 0$ is a tuning parameter. We treat the roughness measure $P(\cdot)$ as a penalty function which penalizes "complexity."

Instead of finding the exact solution of an infinite dimensional problem, we have proposed approximating $\hat{f}$ using the solution to a finite dimensional problem with optimization parameters given as the fitted values at a mesh of points, which we call a *mesh based solution*

or MBS. For each covariate $\boldsymbol{x}_j$, we choose m_j evenly spaced points as a convex hull over $\boldsymbol{x}_j$. For p covariates, a mesh, D, is the Cartesian product over each of the m_j points. Using D, we formulated an approximation to our original problem (3.1) as given in (3.2):

$$\tilde{f}_D = \underset{f_D \in \mathbb{R}^M}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \Omega_{\boldsymbol{x}_i}(f_D)\right)^2 + \lambda_N P_D\left(f_D\right),$$

where f_D denotes the fitted values of the M mesh points. $P_D(f_D)$ is an approximation to $P(f)$; and $\Omega : \mathbb{R}^M \to \mathcal{F}$ is a multivariate interpolator (such as the simple piecewise polynomial). We called $\Omega\left(\tilde{f}_D\right)$ the MBS. We showed that $\Omega\left(\tilde{f}_D\right)$ is a minimax rate-optimal estimator of f^* under conditions dependent on the number of mesh points and the interpolator function, Ω.

In this chapter, we show how MBS can be tailored to address other problems. It will be demonstrated that the ADMM algorithm is a flexible approach to solving convex problems. In Appendix D, we give an overview of ADMM and its properties. We consider three extensions. For each of the extensions, we discuss scientific context followed by how an MBS changes computationally.

First, we discuss *partially linear additive models*, or PLAMs. In a common experimental setting, we may be interested in estimating the linear relationship between an outcome y_i and predictors $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip}) \in \mathbb{R}^p$, but we suspect that features $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{iq}) \in \mathbb{R}^q$ may have a potentially non-linear association to both the outcome and predictors, i.e.

$$y_i = \boldsymbol{x}_i\beta + f^*(\boldsymbol{z}_i) + w_i,$$

where $\beta = (\beta_1, \ldots, \beta_p)^\top \in \mathbb{R}^p$ and otherwise we are in the typical regression setting of Chapters 2 and 3. Using penalized regression and MBS, we can estimate potentially non-linear $f^*(\boldsymbol{z}_i)$ and β. As a penalized regression problem, we want to solve:

$$\left(\hat{f}, \hat{\beta}\right) = \underset{f \in \mathcal{F}_{P,\infty}, \beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \boldsymbol{x}_i\beta - f\left(\boldsymbol{z}_i\right)\right)^2 + \lambda_N P\left(f\right). \tag{4.2}$$

We aim to simultaneously estimate β and f^*, while using $P(\cdot)$ to penalize the wiggliness of our estimate of f^*. In Section 4.2, we discuss the MBS approach and derive ADMM iterates for solving it.

For our second class of problem, we consider the problem of predicting a potentially non-linear, as well as non-decreasing, association between an outcome and predictor. These problems are common. For example, although there is empirical evidence to the contrary, scientists often initially think that incidence of heart disease increases potentially non-linearly as body mass index (BMI) increases with perhaps some intervals of BMI sharing the same risk, i.e. a monotone non-decreasing relationship with potential curvature. We have been discussing penalized regression where we find an estimate that is flexible by solving under $\mathcal{F}_{P,\infty}$. We modify that penalized regression solution to include a monotonicity constraint, i.e. univariate *penalized isotonic regression*. As a penalized regression problem with outcome y_i and predictor x_i, we are interested in finding the following minimizer:

$$\hat{f}_{mon} = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - f\left(x_i\right)\right)^2 + \lambda_N P\left(f\right) \quad \text{s.t. } x_l \leq x_u \implies f(x_l) \leq f(x_u), \tag{4.3}$$

for any $x_l, x_u \in \mathbb{R}$. In Section 4.3 of this chapter, we derive ADMM iterates for penalized isotonic regression.

The third problem we discuss is *an interaction problem*. In the general problem, an outcome depends on p-many primary features, but each dependence has variation that can be described by q-many secondary features. We build up to the general problem by introducing the problem here in the case where $p = 1$ and in Section 4.4 we generalize to $p > 1$. In the interaction problem for $p = 1$, we want to estimate the effect of a primary feature x_i on an outcome y_i as it varies as a function of other secondary features that share coefficients over a surface $\beta^*(\boldsymbol{z}_i)$ ($\boldsymbol{z}_i = (z_{i1}, \ldots, z_{iq}) \in \mathbb{R}^q$). In terms of data generating mechanism, we believe

$$y_i = x_i \beta^*(\boldsymbol{z}_i) + w_i.$$

These types of problems are readily motivated in spatial modeling contexts, where we suspect heterogeneity in the effect of an exposure on an outcome and want to characterize it. For example, an epidemiologist may be interested in predicting the effect of age on disease incidence as it varies over other information such as sociodemographics. As a penalized regression problem, we aim to estimate the heterogenous effect of x_i on y_i governed by β^* with $\hat{\beta}$:

$$\hat{\beta} = \underset{\beta \in \mathcal{F}_{P,\infty}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - x_i \beta\left(\boldsymbol{z}_i\right)\right)^2 + \lambda_N P\left(\beta\right). \tag{4.4}$$

We use MBS to estimate the coefficients $\beta^*(\boldsymbol{z}_i)$ of x_i. In Section 4.4, we discuss the problem in generality for p primary features and derive the ADMM-like iterates for the general interaction problem.

For each of the problems, we make particular use of the ADMM. The ADMM algorithm from Chapters 2 and 3 can be easily modified to solve constrained problems, provided we preserve convexity of the objective function. We end this chapter with a discussion of other possible extensions and future work, Section 4.5.

4.2 Partially Linear Additive Models

In a partially linear additive model, we estimate the linear relationship between an outcome y_i and predictors $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip}) \in \mathbb{R}^p$, while adjusting for the potentially non-linear effect of features $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{iq}) \in \mathbb{R}^q$, i.e.

$$y_i = \boldsymbol{x}_i \beta + f^*(\boldsymbol{z}_i) + w_i, \tag{4.5}$$

where $\beta = (\beta_1, \ldots, \beta_p)^\top \in \mathbb{R}^p$. Using MBS, we estimate $f^*(\boldsymbol{z}_i)$ and β by approximating (4.2). To solve the MBS approximation: we select a mesh D over $\boldsymbol{z}_i$ with M total mesh points and approximate (4.2) with (4.6):

$$\left(\tilde{f}_D, \hat{\beta}\right) = \underset{f_D \in \mathbb{R}^M, \beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \boldsymbol{x}_i \beta - \Omega_{\boldsymbol{z}_i}(f_D)\right)^2 + \lambda_N P_D\left(f_D\right). \tag{4.6}$$

Let $\boldsymbol{x}_i$ be the ith row of the matrix $X \in \mathbb{R}^{N\times p}$. With interpolation matrix $O \in \mathbb{R}^{N\times M}$ and difference matrix $\mathcal{D}$ and ℓ_1-based penalty, we can rewrite (4.6) as:

$$\left(\tilde{f}_D, \hat{\beta}\right) = \underset{f_D\in\mathbb{R}^m, \beta\in\mathbb{R}^p}{\operatorname{argmin}} \|y - X\beta - Of_D\|_N^2 + \lambda \|\mathcal{D}f_D\|_1 . \tag{4.7}$$

Since linear operations on parameters preserve convexity, the addition of $X\beta$ in the squared loss term preserves convexity. Since the ℓ_1-norm is also convex, (4.7) is a convex objective function. Hence, we can proceed with deriving the ADMM knowing the iterates will converge.

To solve via ADMM, we first set our primal variables: $\theta = \begin{bmatrix} f_D \\ \beta \end{bmatrix} = (f_D; \beta)$ and $\alpha = A\theta = \mathcal{D}f_D$, where $A = (\mathcal{D}, \mathbf{0})$. The ADMM problem is then given by

$$\min_{f_D\in\mathbb{R}^m, \beta\in\mathbb{R}^p, \alpha\in\mathbb{R}^{m-r-1}} \|y - X\beta - Of_D\|_N^2 + \lambda \|\alpha\|_1 \quad \text{s.t. } \alpha = A\theta. \tag{4.8}$$

Our goal now is to calculate the gradient of $\theta = (f_D; \beta)$ in an equation that introduces the constraints via multiplier terms, i.e. a Lagrangian function. We introduce a dual variable u and write the Lagrangian function as

$$\mathcal{L}_\rho(f_D, \beta, \alpha, u) = \|y - X\beta - Of_D\|_N^2 + \lambda\|\alpha\|_1 + u\top(\mathcal{D}f_D - \alpha) + \rho\|\mathcal{D}f_D - \alpha\|_2^2.$$

Next, we find the gradients for β and f_D:

$$\nabla_\beta \mathcal{L}_\rho(f_D, \beta, \alpha, u) = -X^\top(y - X\beta - Of_D) \tag{4.9}$$

$$\nabla_{f_D} \mathcal{L}_\rho(f_D, \beta, \alpha, u) = -O^\top(y - X\beta - Of_D) + \mathcal{D}^\top u + \rho\mathcal{D}^\top(\mathcal{D}f_D - \alpha). \tag{4.10}$$

Solving for β and f_D in ∇_β and ∇_{f_D}, respectively, we get the ADMM iterates for PLAMs:

$$\beta^{j+1} \leftarrow \left(X^\top X\right)^{-1} X^\top \left(y - Of_D^j\right), \tag{4.11}$$

$$f_D^{j+1} \leftarrow \left(O^\top O + \rho(\mathcal{D}^\top\mathcal{D})\right)^{-1} \left(O^\top(y - X\beta^{j+1}) + \rho\mathcal{D}^\top(\alpha^j + u^j)\right). \tag{4.12}$$

The iterates for α and u are the same as the iterates for (2.1), since the gradient of α does not depend on β and u is defined as a step variable updating the dual. Computationally, we have only added a least squares estimate at each iteration: we can cache $(X^\top X)^{-1}X^\top y$ at the first iteration, but we will need to multiply the storable $(X^\top X)^{-1}X^\top O$ by the updating $f_D \in \mathbb{R}^M$. Thus, the procedure as a whole still runs in linear time in M per iteration. Note that the iterate for β is the difference of the least-squares estimates of slope for y and Of_D (the smoothed version of $\boldsymbol{z}_i$), which is precisely the first order trend we aimed to estimate.

4.3 Penalized Isotonic Regression

For our second problem, we believe a predictor, x_i, and outcome, y_i, have a potentially non-linear and monotonic relationship. We are interested in estimating f^* that is isotonic, but has unknown curvature otherwise. We aim to find a data-driven estimate of f^* via penalized isotonic regression.

As a penalized isotonic regression problem, we aim to find the following minimizer in (4.3):

$$\hat{f}_{mon} = \underset{f\in\mathcal{F}}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N}\left(y_i - f\left(x_i\right)\right)^2 + \lambda_N P\left(f\right) \quad \text{s.t. } x_l \le x_u \implies f(x_l) \le f(x_u),$$

for all $x_l, x_u \in \mathbb{R}$. As an MBS problem, we select a mesh $D \equiv \{d_1, \ldots, d_m\}$ and an appropriate kth order interpolator Ω relative to the rth order $P(\cdot)$ we will be approximating ($k \le r$) and solve the minimization problem:

$$\tilde{f}_D = \underset{f_D\in\mathbb{R}^m}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \Omega_{x_i}(f_D)\right)^2 + \lambda_N P_D\left(f_D\right) \quad \text{s.t. } d_l \le d_u \implies f(d_l) \le f(d_u), \quad (4.13)$$

for all $d_l, d_u \in \mathbb{R}$. Note that $f_D = \left(f(d_1), \ldots, f(d_m)\right)^\top$, where $d_1 < \ldots < d_m$. With interpolation matrix O, difference matrix $\mathcal{D}$ (let $\mathcal{D}^1$ denote the first difference matric) and

ℓ_1-based penalty, we can rewrite (4.13) as:

$$\tilde{f}_D = \underset{f_D \in \mathbb{R}^m}{\operatorname{argmin}} \|y - Of_D\|_N^2 + \lambda \|\mathcal{D}f_D\|_1 + \quad \text{s.t. } d_l \leq d_u \implies f(d_l) \leq f(d_u). \tag{4.14}$$

In our previous problems, we set a primal variable $\alpha = \mathcal{D}f_D$, which was accompanied by a dual variable u and step-size ρ. In this problem, we use the constraint

$$\begin{bmatrix} \alpha \\ \gamma \end{bmatrix} := z = \begin{bmatrix} \mathcal{D} \\ I \end{bmatrix} f_D.$$

The ADMM problem we aim to solve is

$$\min_{f_D \in \mathbb{R}^m, \alpha \in \mathbb{R}^{m-r-1}, \gamma \in \mathbb{R}^m} \|y - Of_D\|_N^2 + \lambda \|\alpha\|_1 + I_+\left(\mathcal{D}^{1\top}\gamma\right) \quad \text{s.t. } z = (\alpha; \gamma) = (\mathcal{D}f_D; f_D), \tag{4.15}$$

where $I_+(\cdot)$ is the convex indicator of the non-negative orthant. We introduce a dual variable $u = (u_\alpha; u_\gamma)$ and write the augmented Lagrangian as

$$L_\rho(f_d, z, u) = \|y - Of_D\|_N^2 + \lambda\|\alpha\|_1 + I_+(\mathcal{D}^{1\top}\gamma) + u^\top((\mathcal{D}f_D; f_D) - z) + \frac{\rho}{2}\|(\mathcal{D}f_D; f_D) - z\|_2^2.$$

This results in the following updates:

$$f_D \leftarrow \underset{f_D}{\operatorname{argmin}} \|y - Of_D\|_N^2 + \frac{\rho}{2}\|(\mathcal{D}f_D; f_D) - z + u\|_2^2 \tag{4.16}$$

$$z \leftarrow \underset{z}{\operatorname{argmin}} \lambda\|\alpha\|_1 + I_+(\mathcal{D}^{1\top}\gamma) + \frac{\rho}{2}\|(\mathcal{D}f_D; f_D) - z + u\|_2^2 \tag{4.17}$$

$$u \leftarrow u + z - (\mathcal{D}f_D; f_D). \tag{4.18}$$

The f_D-update is straightforward to derive. We can break the z-update into two parallel

updates since $z = (\alpha; \gamma)^\top$:

$$\alpha \leftarrow \underset{\alpha}{\operatorname{argmin}} \lambda \|\alpha\|_1 + \frac{\rho}{2} \|\mathcal{D} f_D - \alpha + u_\alpha\|_2^2 \tag{4.19}$$

$$\gamma \leftarrow \underset{\gamma}{\operatorname{argmin}} I_+(\mathcal{D}^{1\top} \gamma) + \frac{\rho}{2} \|f_D - \gamma + u_\gamma\|_2^2. \tag{4.20}$$

As before, the α-update is solved by the soft-thresholding operator. In the γ-update, setting $\gamma = IR(f_D - u_\gamma)$ makes $\|f_D - \gamma + u_\gamma\|_2^2 = 0$ and $I_+(\mathcal{D}^{1\top}\gamma) = 0$, since $IR(\cdot)$ indicates isotonic regression on $f_D + u_\gamma$.

The iterates reduce to the following form:

$$f_D^{j+1} \leftarrow \left(O^\top O + \rho(\mathcal{D}^\top \mathcal{D})\right)^{-1} \left(O^\top y + \rho(\mathcal{D}^\top(\alpha^j + u_\alpha^j) + \gamma^j + u_\gamma^j)\right), \tag{4.21}$$

$$\alpha^{j+1} \leftarrow S_{\lambda/\rho}\left(\mathcal{D} f_D^{j+1} - u_\alpha^j\right), \tag{4.22}$$

$$\gamma^{j+1} \leftarrow IR(f_D^{j+1} - u_\gamma^j), \tag{4.23}$$

$$u_\alpha^{j+1} \leftarrow u_\alpha^j + \alpha^{j+1} - \mathcal{D} f_D^{j+1}, \tag{4.24}$$

$$u_\gamma^{j+1} \leftarrow u_\gamma^j + \gamma^{j+1} - f_D^{j+1}. \tag{4.25}$$

Since $IR(f_D^{j+1} + u_\gamma^j)$ can be run in linear time in m ($f_D \in \mathbb{R}^m$), each iteration of the procedure as a whole can still be run in linear time in m.

4.4 The Interaction Problem

In Section 4.1, we introduced the so-called *interaction problem* as the problem of estimating a heterogenous effect of a covariate on an outcome. For the general case, we want to estimate the effect of a set of primary features $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})$ on an outcome y_i as it varies as a function of other secondary features that share coefficients on a surface, $\beta^*(\boldsymbol{z}_i)$ ($\boldsymbol{z}_i = (z_{i1}, \ldots, z_{iq}) \in \mathbb{R}^q$). In terms of data generating mechanism, we believe

$$y_i = \boldsymbol{x}_i^\top \beta^*(\boldsymbol{z}_i) + w_i.$$

As a penalized regression problem, we aim to estimate the heterogenous effect of $\boldsymbol{x}_i$ on y_i governed by β^* with $\hat{\beta}$:

$$\hat{\beta} = \underset{\beta \in \mathcal{F}_{P,\infty}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \boldsymbol{x}_i^\top \beta \left(\boldsymbol{z}_i \right) \right)^2 + \lambda_N Q \left(\beta \right). \tag{4.26}$$

In our approach to this problem, we assume each covariate has a conditional association with y_i that varies over $\boldsymbol{z}_i$ and the surface over $\boldsymbol{z}_i$ can be different per feature. One way we can simplify (4.26) is to assume the penalty decouples as $Q(\beta) = \sum_{j=1}^{p} P(\beta^j)$:

$$\left(\hat{\beta}^1, \ldots, \hat{\beta}^p \right) = \underset{\forall_j \beta^j \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} x_{i,j} \beta^j \left(\boldsymbol{z}_i \right) \right)^2 + \lambda_N \sum_{j=1}^{p} P \left(\beta^j \right), \tag{4.27}$$

where $P(\beta^j)$ governs the shape of each coefficient surface β^j.

Let $\boldsymbol{x}_i$ and $\boldsymbol{z}_i$ be the ith rows of matrix $X \in \mathbb{R}^{N \times p}$ and $Z \in \mathbb{R}^{N \times q}$. For each β^j we aim to estimate, we construct a mesh D_j of m_j mesh points over Z and solve for the fitted values of the mesh points, β_D^j, as the solution to the following minimization:

$$\left(\tilde{\beta}_D^1, \ldots, \tilde{\beta}_D^p \right) = \underset{\forall_j \beta_D^j \in \mathbb{R}^{m_j}}{\operatorname{argmin}} \| y - \sum_{j=1}^{p} X O^j \beta_D^j \|_N^2 + \lambda \sum_{j=1}^{p} \left\| \mathcal{D}^j \beta_D^j \right\|_1, \tag{4.28}$$

where O^j and $\mathcal{D}^j$ are interpolation and difference matrices, respectively. We treat $O^j \tilde{\beta}_D^j$ as the coefficients of the jth primary feature, $\boldsymbol{x}_j$.

Our procedure for finding iterates to the problem is similar to the partially linear additive model, but we have increased the number of optimization parameters. Furthermore, the iterates are not traditional ADMM iterates, which is easier to see if we do not attempt to separate the parameters into separable primals: first set $\alpha_j = \mathcal{D}^j \beta_D^j$ and introduce dual variables u_j. Let $\beta_D^{\cdot}, \alpha_{\cdot}, u_{\cdot}$ denote the collection of j elements for each variable. We write

the Lagrangian function as:

$$\mathcal{L}_\rho(\beta_D^{\cdot}, \alpha_\cdot, u_\cdot) = \left\| y - \sum_{j=1}^{p} XO^j \beta_D^j \right\|_N^2 + \lambda \sum_j \|\alpha_j\|_1 + \sum_j u_j^\top (\mathcal{D}^j \beta_D^j - \alpha_j) + \sum_j \rho_j \|\mathcal{D}^j \beta_D^j - \alpha_j\|_2^2.$$

$\mathcal{L}_\rho(\beta_D^{\cdot}, \alpha_\cdot, u_\cdot)$ looks difficult, but it is not much harder than previous problems for finding the gradient of f_D^j. The iterates can be found as:

$$\beta_D^j \leftarrow \left(O^{j\top} X^\top (XO^j + \sum_{k \neq j} XO^k \beta_D^k) + \rho_j (\mathcal{D}^{j\top} \mathcal{D}^j) \right)^{-1} \left(O^{j\top} X^\top y + \rho_j \mathcal{D}^{j\top} (\alpha_j + u_j) \right), \tag{4.29}$$

$$\alpha_j \leftarrow S_{\lambda/\rho_j} \left(\mathcal{D}^j \beta_D^j - u_j \right), \tag{4.30}$$

$$u_j \leftarrow u_j + \alpha_j - \mathcal{D}^j \beta_D^j. \tag{4.31}$$

The β_D^j iterates depend on all other β_D^k $(k \neq j)$, which means there are $p-1$ interior points. When $p = 1$, we avoid this problem and ADMM works with theoretical guarantees. For now, we can discuss this as a pseudo-ADMM and update each β_D^j in turn, but it is not clear what convergence properties this procedure will have.

As an MBS solution, the complexity of each iterate is governed by the size of the mesh D_j over $Z \in \mathbb{R}^{N \times q}$, which for $p = 1$ could be a straightforward problem to solve. However, suppose we chose the same number of mesh points, m^q, per mesh D_j. For $p > 1$ primary features of interest, we would need to solve for pm^q mesh fitted values, as well as the $2p$ other dual variables. This problem can easily become computationally intense, even as an MBS.

4.5 Discussion

In this chapter we discussed extensions of MBS. We described MBS formulations for a penalized isotonic regression problem, partially linear additive models, and interaction problems. Our goal was to demonstrate how the MBS approach can be included into many problems with constraints that can be translated into a convex objective. In the problems we con-

sidered, the MBS offers novel approaches that reduce the overall computation compared to previous methods. For example, there exist nonparametric penalized regression methods that solve the penalized isotonic regression problem [14, 27]. In our approach to penalized isotonic regression, we can reduce the number of optimization targets, while still producing flexible and computationally tractable fits to data. Similar to [27], we used ADMM, but in our derivation of ADMM-based penalized isotonic regression, we only required one step-size parameter instead of two, which is a useful reduction in the overall complexity of the problem.

We discussed partially linear additive models and gave an example where we wanted to estimate the linear relationship between a predictor and outcome in the presence of confounding. For the confounding features, we provided a flexible nonparametric estimate of their shared surface. Then we were able to estimate the linear trend between predictor and outcome adjusted for those confounders. The confounder surface and linear trend are simultaneously estimated using ADMM.

Under the PLAM problem, we found the coefficient of the linear relationship to be $\hat{\beta} = \left(X^\top X\right)^{-1} X^\top \left(y - O\tilde{f}_D\right)$. It may be possible to make an inference procedure based on this coefficient. We can re-write (4.5) as

$$y = X\beta + Of_D + w. \tag{4.32}$$

To test $H_{null} : \beta = 0$, we might assume normally distributed errors w, $w_i \sim N(0, \sigma^2)$ and design a score test based on the derivative of the log-likelihood of the model under the null hypothesis. Based on the partially linear additive model in (4.32), a score-like statistic might be given as

$$S = X^\top (y - y_{null})/\sqrt{N}, \tag{4.33}$$

where $y_{null} = Of_D$. However, determining asymptotic and finite-sample behavior of this

procedure would be difficult, since our estimation of β depends on a tuning parameter. It could be an interesting topic for future work.

Finally, motivated by an epidemiological problem, we considered an interaction problem. Our goal was to estimate the heterogenous effect of a set of p-many predictors on an outcome, where that heterogeneity may come from (in part) a set of q-many secondary features. As discussed in Chapter 2, we can grow a mesh until we achieve a minimax rate optimal estimator of the surface shared by the secondary features. For $p = 1$ and $m \geq 1$, our ADMM-based approach works with guaranteed convergence (though even moderate m has computational difficulties). However, our proposed pseudo-ADMM for $p > 1$ lacks theoretical guarantees and will be computationally intense for moderate sizes of p and q.

For each application case, we translated the problems into a convex objective to be solved via ADMM. The ADMM iterates changed in a way that allowed us to easily track changes in computational complexity. Of course, there are other problems we could consider, but these were only to show the user interface of the MBS framework.

Chapter 5

DISCUSSION

This dissertation has been about fitting functions when we assume the true data generating function lives in an infinite dimensional functional class. We began with the experiment where we observe an outcome, $y_i \in \mathbb{R}$, and set of predictors, $\boldsymbol{x}_i \in \mathbb{R}^p$, as

$$y_i = f^*(\boldsymbol{x}_i) + w_i.$$

We have chosen to assume f^* lives in an infinite dimensional function class, i.e. $f^* \in \mathcal{F}_{P,\infty}$, where

$$\mathcal{F}_{P,\infty} := \{f : \mathbb{R}^p \to \mathbb{R} | P(f) \leq \infty\}$$

and $P(f)$ is some measure of smoothness. We can solve over $\mathcal{F}_{P,\infty}$ to find an estimate $\hat{f}$ of f^* using penalized regression as in (3.1) [8, 33]:

$$\hat{f} = \underset{f \in \mathcal{F}_{P,\infty}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - f\left(\boldsymbol{x}_i\right)\right)^2 + \lambda_N P\left(f\right),$$

where $\lambda_N \geq 0$ is a tuning parameter and we treat the roughness measure $P(\cdot)$ as a function penalizing "complexity." As discussed before, with carefully chosen λ_N, we know that the exact solution $\hat{f}$ converges at the minimax rate under $\mathcal{F}_{P,c}$ for many choices of $P(f)$ [21, 33]. However, that exact solution can be difficult to compute as we have discussed for many choices of $P(f)$ whether univariate or multivariate.

Instead of finding the exact solution of an infinite dimensional problem, we have proposed approximating $\hat{f}$ using the solution to a finite dimensional problem with optimization parameters given by a mesh, which we call a *mesh based solution* or MBS. For each covariate $\boldsymbol{x}_j$,

we choose m_j evenly spaced points as a convex hull over $\boldsymbol{x}_j$. For p covariates, a mesh, D, is the Cartesian product over each of the m_j points. Using D, we formulated an approximation to our original problem (3.1) as given in (3.1):

$$\tilde{f}_D = \underset{f_D \in \mathbb{R}^M}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \Omega_{\boldsymbol{x}_i}(f_D)\right)^2 + \lambda_N P_D\left(f_D\right),$$

where f_D denotes the fitted values of the M mesh points. $P_D(f_D)$ is an approximation to $P(f)$; and $\Omega : \mathbb{R}^M \to \mathcal{F}$ is a multivariate interpolator. We called $\Omega\left(\tilde{f}_D\right)$ the MBS. We showed that $\Omega\left(\tilde{f}_D\right)$ converges to f^* at the minimax rate for the class $\mathcal{F}_{P,1}$ under some conditions dependent on Ω.

The MBS approach is similar to other penalized regression methods such as trend filter and locally adaptive regression splines. However, previous methods used observed points as knots of a basis set then adaptively choose the optimal set of knots, which can be problematic computationally either because of the basis set itself or the poor conditioning that comes with knots at observed points. MBS overcomes this by optimizing over the fitted values f_D at mesh points D that need not have been observed. The fitted values f_D are then used in both the penalty approximation and in fitting the observed data in a computationally inexpensive way. Interestingly, [24] described a piecewise linear Bayesian regression approach to isotonic regression that also does not require specification of knots, but places priors on the knot locations. However, it requires Markov chain Monte Carlo to retrieve a smooth estimate of the regression, which introduces computational difficulties for large problems. In Section 4.3, we discussed our approach to univariate penalized isotonic regression that is agnostic of knot locations and can be defined to be computationally tractable.

MBS allowed for a novel solution to the total variation problem for any number of covariates. The `MTV` package will be available on `CRAN` soon, which provides an implementation for the total variation solution. Other solvers written in `Python` can be found on `GitHub` that solve for univariate and bivariate nonparametric regression problem with the Sobolev-like penalties that we have been discussing. In the future, we may pursue implementing multi-

variate MBS for higher order variational problems. Furthermore, as discussed in Chapter 2, it will be important to implement these MBS solutions using consensus ADMM to overcome computational obstacles when a large mesh is required, as is the case for multivariate MBS.

In Chapter 4, we discussed some extensions of MBS for an isotonic regression problem, a partially linear additive model, and an interaction model. We demonstrated the flexibility of the ADMM algorithm that we have been using to solve MBS. However, we did not provide real data applications of the methods proposed. It will be useful to package into software the isotonic regression and partially linear additive model solutions.

Finally, our MBS proposal has been built throughout these chapters using a regular or evenly spaced mesh. Using a large regular mesh, we were able to fit a computationally tractable and minimax rate-optimal estimator based on data that may or may not be regularly spaced. However, it may be possible to exploit irregular spacing of the inputs for computational benefit. It would be interesting work to consider a mesh of points defined at user specified quantiles: choose m percentiles of interest, $p_1, \ldots, p_m$, and estimate the points $d_j = F^{-1}(p_j)$ with sample quantiles, $\hat{d}_j = \hat{F}^{-1}(p_j)$. Quantile-driven mesh points, $\hat{d}_1, \ldots, \hat{d}_m$, could be useful when fitting data with clustering, especially in the multivariate setting. Currently, as proposed, we need to grow the mesh quite finely to account for sparsely and densely populated areas of the sample space simultaneously. There is no statistical cost for having too many mesh points under our penalized framework, but there is a computational cost in storing and defining matrices by so many optimization parameters. However, using quantiles, we could focus on growing the mesh in regions with the most information.

BIBLIOGRAPHY

[1] Taylor B Arnold and Ryan J Tibshirani. genlasso: Path algorithm for generalized lasso problems. *R package version*, 1, 2014.

[2] Richard E Barlow and Hugh D Brunk. The isotonic regression problem and its dual. *Journal of the American Statistical Association*, 67(337):140–147, 1972.

[3] Richard E Bellman. *Adaptive control processes: a guided tour*. Princeton university press, 2015.

[4] David Benkeser and Mark Van Der Laan. The highly adaptive lasso estimator. In *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*, pages 689–696. IEEE, 2016.

[5] Julian Besag and Peter J Green. Spatial statistics and bayesian computation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 25–37, 1993.

[6] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[7] Lawrence D Brown, Michael Levine, and Lie Wang. A semiparametric multivariate partially linear model: a difference approach. *Journal of Statistical Planning and Inference*, 178:99–111, 2016.

[8] Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

[9] Richard L Burden and J Douglas Faires. Numerical analysis pws, 1989.

[10] Peter Craven and Grace Wahba. Smoothing noisy data with spline functions. *Numerische Mathematik*, 31(4):377–403, 1978.

[11] Jean Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*, pages 85–100. Springer, 1977.

[12] Reinhard Furrer, Douglas Nychka, and Stephen Sain. fields: Tools for spatial data. *R package version*, 6(11), 2009.

[13] Peter J Green and Sylvia Richardson. Hidden markov models and disease mapping. *Journal of the American statistical association*, 97(460):1055–1070, 2002.

[14] Zheng Han and Frank E Curtis. Primal-dual active-set methods for isotonic regression and trend filtering. *arXiv preprint arXiv:1508.02452*, 2015.

[15] Wolfgang Härdle and Hua Liang. *Partially linear models*. Springer Berlin Heidelberg, 2007.

[16] Trevor Hastie and Robert Tibshirani. Generalized additive models. *Statistical science*, pages 297–310, 1986.

[17] Trevor J Hastie and Robert J Tibshirani. *Generalized Additive Models*, volume 43. CRC Press, 1990.

[18] Seung-Jean Kim, Kwangmoo Koh, Stephen Boyd, and Dimitry Gorinevsky. \ell_1 trend filtering. *SIAM review*, 51(2):339–360, 2009.

[19] Andrei Nikolaevich Kolmogorov and Vladimir Mikhailovich Tikhomirov. ε-entropy and ε-capacity of sets in function spaces. *Uspekhi Matematicheskikh Nauk*, 14(2):3–86, 1959.

[20] Enno Mammen, Sara van de Geer, et al. Locally adaptive regression splines. *The Annals of Statistics*, 25(1):387–413, 1997.

[21] Pascal Massart. *Concentration inequalities and model selection*, volume 6. Springer, 2007.

[22] Annie Mollié. Bayesian mapping of disease. *Markov chain Monte Carlo in practice*, 1:359–379, 1996.

[23] Patric Müller and Sara Van de Geer. The partial linear model in high dimensions. *Scandinavian Journal of Statistics*, 2014.

[24] Brian Neelon and David B Dunson. Bayesian isotonic regression and trend analysis. *Biometrics*, 60(2):398–406, 2004.

[25] JA Nelder and RWM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society, Series A (General)*, 135(3):370–384, 1962.

[26] Brayan Ortiz and Noah Simon. Mesh based solutions to nonparametric regression. *arXiv preprint*, 2018.

[27] Aaditya Ramdas and Ryan J Tibshirani. Fast and flexible admm algorithms for trend filtering. *Journal of Computational and Graphical Statistics*, (just-accepted), 2015.

[28] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[29] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[30] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B (Methodological)*, pages 267–288, 1996.

[31] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[32] Ryan J Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323, 2014.

[33] Sara A Van De Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge university press, 2000.

[34] Mark J van der Laan and David Benkeser. Highly adaptive lasso (hal). In *Targeted Learning in Data Science*, pages 77–94. Springer, 2018.

[35] Grace Wahba. Smoothing noisy data with spline functions. *Numerische Mathematik*, 24(5):383–393, 1975.

[36] Simon N Wood. Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):95–114, 2003.

Appendix A

MATRIX NOTATION OF UNIVARIATE/BIVARIATE MBS

We denote a first order difference matrix by

$$\Delta_n^{(1)} = \begin{pmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & \dots & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(n-1)\times n}. \tag{A.1}$$

The rth order difference matrix is defined recursively as follows:

$$\Delta_n^{(r)} = \Delta_{n-1}^{(r-1)} \cdot \Delta_n^{(1)} = \Delta_{n-r+1}^{(1)} \cdot \Delta_{n-r}^{(1)} \cdot \ldots \cdot \Delta_n^{(1)} \in \mathbb{R}^{(n-r)\times n}. \tag{A.2}$$

We define an averaging operator matrix by

$$A_n^k = \frac{1}{r+1} \begin{pmatrix} \overbrace{1 \;\; 1 \;\; \dots \;\; 1}^{\text{(r+1)-many}} & 0 & \dots & 0 \\ 0 \;\; 1 \;\; \dots \;\; 1 & 1 & \dots & 0 \\ \vdots & & & \\ 0 \;\; 0 \;\; 0 \;\; \dots \;\; 1 & 1 & 1 & 1 \end{pmatrix} \in \mathbb{R}^{(n-r)\times n}. \tag{A.3}$$

A_m^0 gives the identity matrix.

Suppose $p = 1$ and we choose an m-mesh D (not necessarily regular). We use finite-differences/Riemann sums to approximate the Sobolev norm denoted by $P(f)$:

$$P_D(f_D) = \left\| \bar{\Delta}_m^{(r)} f_D \right\|_\ell^\ell \tag{A.4}$$

where $f_D = [f(d_1), \ldots, f(d_m)]^\top$ and $\bar{\Delta}^{(r)}_{m,D}$ is an rth order normalized difference over D:

$$\bar{\Delta}^{(r)}_m = \left(\tilde{\Delta}^{(r)}_m\right)^{1/\ell} \left((\tilde{\Delta}^{(r)}_m)^{-1}\Delta^{(1)}_{m-r}\right)\left((\tilde{\Delta}^{(r-1)}_m)^{-1}\Delta^{(1)}_{m-r-1}\right)\ldots\left((\tilde{\Delta}^{(1)}_m)^{-1}\Delta^{(1)}_{m-1}\right)(\tilde{\Delta}^{(0)}_m)^{-1}\Delta^{(1)}_m, \tag{A.5}$$

with

$$\tilde{\Delta}^{(r)}_m = \operatorname{diag}\left(\Delta^{(1)}_{m-r}A^r_m D\right) \in \mathbb{R}^{(m-r-1)\times(m-r-1)}.$$

$A^r_m D$ produces the averages of the $r+1$ adjacent values of the mesh. $\Delta^{(1)}_{m-r}A^r_m D$ gives the rth order differences of the r adjacent mesh point averages.

Through this generalized matrix formulation, we can formulate the approximating norms with some algebra. For example, with $r = 1$ and $\ell = 2$, we have

$$P_D(f_D) = \sum_{i=1}^{m-2} \frac{\left(\frac{f(d_{i+2})-f(d_{i+1})}{d_{i+2}-d_{i+1}} - \frac{f(d_{i+1})-f(d_i)}{d_{i+1}-d_i}\right)^2}{\frac{d_{i+2}-d_i}{2}}.$$

On a regular m-mesh D with mesh widths δ, (A.5) reduces nicely. For integers $r \geq 0$ and $\ell > 0$, our Riemann approximation to $P(f)$ takes the following form:

$$P_D(f_D) = \left\|\delta^{\frac{1}{\ell}-r}\Delta^{(r)}_m f_D\right\|^\ell_\ell. \tag{A.6}$$

In the bivariate case with regular meshes chosen for each covariate, we arrive at simple expressions of the Riemann approximation. Let $D = (D_1, D_2)$ denote regular meshes for covariates x_1 and x_2, respectively, i,e. $\boldsymbol{m} = (m_1, m_2)$. Define $\theta \in \mathbb{R}^{\boldsymbol{m}}$ such that

$$\theta = \begin{pmatrix} f(d_{1,1}, d_{2,1}) & f(d_{1,2}, d_{2,1}) & \ldots & f(d_{1,m_1}, d_{2,1}) \\ f(d_{1,1}, d_{2,2}) & f(d_{1,2}, d_{2,2}) & \ldots & f(d_{1,m_1}, d_{2,2}) \\ \vdots & & & \\ f(d_{1,1}, d_{2,m_2}) & f(d_{1,2}, d_{2,m_2}) & \ldots & f(d_{1,m_1}, d_{2,m_2}) \end{pmatrix}. \tag{A.7}$$

Furthermore, let $\boldsymbol{r} = (r_1, r_2)$ denote the partials we seek to estimate. Similar to previous

notation, but with sub-indices for the covariates, the r_jth-order differences for the jth variable can be calculated through $\Delta_{m_j}^{(r)}\theta^{[j]}$, where $\theta^{[1]} = \theta$ and $\theta^{[2]} = \theta^\top$. In this bivariate case with pure partials, i.e. taking differences only for one covariate or isotropic differences, with mesh widths denoted by $\boldsymbol{\delta} = (\delta_1, \delta_2)$,

$$P_D\left(f_D\right) = \left\| \delta_j^{\frac{1}{\ell}-r_j} \delta_{j'}^{\frac{1}{\ell}} \Delta_{m_j}^{(r_j)} \theta^{[j]} \right\|_\ell^\ell,$$

where $j = 1, 2$ and $j' = 1, 2$ $(j \neq j')$.

For mixed partials or anisotropic derivatives with $\boldsymbol{r} = (r_1, r_2)$, we can calculate the approximating differences using

$$\Delta_{\boldsymbol{m}}^{(\boldsymbol{r})}\theta = \Delta_{m_1}^{(r_1)}\theta(\Delta_{m_2}^{r_2})^\top$$

or

$$\Delta_{\boldsymbol{m}}^{(\boldsymbol{r})}\theta = \Delta_{m_2}^{(r_2)}\theta^\top(\Delta_{m_1}^{(r_1)})^\top.$$

Thus, we can estimate $P(f)$ with

$$P_D\left(f_D\right) = \left\| \delta_1^{\frac{1}{\ell}-r_1} \delta_2^{\frac{1}{\ell}-r_2} \Delta_{\boldsymbol{m}}^{(\boldsymbol{r})} \theta \right\|_\ell^\ell.$$

In the general bivariate case, for $\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_S\}$,

$$P_D\left(f_D\right) = \sum_{s=1}^{S} \left\| \delta_1^{\frac{1}{\ell}-r_{1,s}} \delta_2^{\frac{1}{\ell}-r_{2,s}} \Delta_{\boldsymbol{m}}^{(\boldsymbol{r}_s)} \theta \right\|_\ell^\ell$$

Appendix B

RISING POLYNOMIAL BASIS

Here, we define the Kth order rising polynomial basis. Suppose we define a mesh $D = (d_1, \ldots, d_m)$. For some integer Q, set $m = 1 + QK$. Let $T = \left(d_{1+K}, d_{1+2K}, \ldots, d_{1+(Q-1)K}, d_m\right) = (t_1, \ldots, t_Q)$. The Kth order rising polynomial has basis elements given by

$$\psi_1(x) = 1, \psi_2(x) = x, \ldots, \psi_{K+1}(x) = x^K, \text{ and } \psi_{q,k}(x) = (x - t_{q+1})_+^k - (x - t_q)_+^k,$$

where $q = 1, \ldots, Q$ and $k = 1, \ldots, K$.

To see the structure of this basis, suppose we have a mesh with 5 points, $D = \{d_1 < d_2 < \ldots < d_5\}$. Let points $x_1 \in [d_1, d2]$, $x_2 \in [d_2, d_3]$, and $x_3 \in [d_4, d_5]$. For a linear interpolator $k = 1$, we evaluate the basis elements for each element to form the basis matrix, Ψ_D, as follows:

$$\Psi_D = \begin{pmatrix} 1 & x_1 & -(x_1 - d_1) & 0 & 0 & 0 \\ 1 & x_2 & d_1 - d_2 & -(x_2 - d_2) & 0 & 0 \\ 1 & x_3 & d_1 - d_2 & d_2 - d_3 & d_3 - d_4 & -(x_3 - d_4) \end{pmatrix}. \tag{B.1}$$

For ordered points $x_1 < x_2 < \ldots < x_N$, Ψ_D based on the rising polynomial basis will tend to have a roughly lower triangular form.

Appendix C

PROOFS OF THEORETICAL RESULTS

C.1 Sub-Optimality Lemma

First, we introduce notation for the estimators compared in this section. The exact solution of PR is given by (1.2)

$$\hat{f} = \underset{f \in \mathcal{F}_{P,\infty}}{\operatorname{argmin}} L\left(f\right)$$

where

$$L(f) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - f\left(x_i\right)\right)^2 + \lambda_N P\left(f\right).$$

We refer to $L(f)$ as the functional loss of f.

On a mesh D, we solve our approximation to penalized regression via (2.1):

$$\tilde{f_D} = \underset{f_D \in \mathbb{R}^m}{\operatorname{argmin}} L_D(f_D),$$

where

$$L_D\left(f_D\right) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \Omega_{x_i}(f_D)\right)^2 + \lambda_N P_D\left(f\right).$$

The functional estimate, i.e. MBS, is given by $\Omega\left(\tilde{f_D}\right)$.

At best, MBS, $\Omega\left(\tilde{f_D}\right)$, approximates the solution of a penalized regression problem, $\hat{f}$. MBSis computationally sub-optimal in the sense that

$$L\left(\hat{f}\right) \leq L\left(\Omega\left(\tilde{f_D}\right)\right).$$

However, under certain conditions, it can be as optimal as penalized regression exact solutions, as we show next.

Lemma C.1.1 (Sub-Optimality Inequality) *For all $f \in \mathcal{F}_{P,\infty}$, suppose there exist δ_m and ϵ_m such that* (2.40) *holds,*

$$\sup_{f \in \mathcal{F}_{P,\infty}} |P_D(D(f)) - P(f)| \leq \epsilon_m,$$

and (2.41) *holds,*

$$\sup_{f \in \mathcal{F}_{P,\infty}} |\Omega_x(D(f)) - f(x)| \leq \delta_m.$$

With $\lambda_N > 0$, we have

$$L\left(\Omega\left(\tilde{f}_D\right)\right) \leq \min \begin{cases} 3L(\hat{f}) + O_P\left(\delta_m^2 \vee \epsilon_m \lambda_N\right) \\ L(\hat{f}) + O_P\left(C\delta_m \vee \epsilon_m \lambda_N\right) \end{cases}. \tag{C.1}$$

Proof We know $L_D\left(\tilde{f}_D\right) \leq L_D\left(D\left(\hat{f}\right)\right)$. After some algebra, we find

$$L_D\left(\tilde{f}_D\right) = L\left(\Omega\left(\tilde{f}_D\right)\right) + \lambda_N\left(P_D\left(\tilde{f}_D\right) - P\left(\Omega\left(\tilde{f}_D\right)\right)\right). \tag{C.2}$$

With similar algebra and by applying Cauchy-Schwarz Inequality, we show

$$L_D\left(D\left(\hat{f}\right)\right) \leq L\left(\hat{f}\right) + \delta_m^2 + 2\|y - \hat{f}\|_N \|\Omega(D(\hat{f})) - \hat{f}\|_N + \epsilon_m \lambda_N \tag{C.3}$$

From here, we have two bounds for $L_D\left(D\left(\hat{f}\right)\right)$. We know $\|y - \hat{f}\|_N^2 \leq L(\hat{f})$, since $P(\hat{f}) > 0$ and $\lambda_N \geq 0$. Hence, we can deduce

$$\begin{aligned} L_D\left(D\left(\hat{f}\right)\right) &\leq L\left(\hat{f}\right) + \delta_m^2 + 2\left(\|y - \hat{f}\|_N^2 \vee \|\Omega(D(\hat{f})) - \hat{f}\|_N^2\right) + \epsilon_m \lambda_N & \text{(C.4)} \\ &\leq L\left(\hat{f}\right) + \delta_m^2 + 2\left(L(\hat{f}) \vee \delta_m^2\right) + \epsilon_m \lambda_N & \text{(C.5)} \\ &\leq 3L\left(\hat{f}\right) + O_P\left(\delta_m^2 \vee \epsilon_m \lambda_N\right). & \text{(C.6)} \end{aligned}$$

However, by using the fact that $L\left(\hat{f}\right) \leq L\left(f^*\right)$, it is not difficult to show

$$\|y - \hat{f}\|_N \leq \|w\|_N + O_P(\lambda_N^{1/2}) < C.$$

Hence, another bound follows that will be useful:

$$L_D\left(D\left(\hat{f}\right)\right) \leq L\left(\hat{f}\right) + \delta_m^2 + 2C\delta_m + \epsilon_m\lambda_N \tag{C.7}$$

$$\leq L\left(\hat{f}\right) + O_P\left(C\delta_m \vee \epsilon_m\lambda_N\right). \tag{C.8}$$

∎

C.2 Rate of Convergence for $\Omega\left(\tilde{f}_D\right)$

We repeat some of the details given in Chapter 2 then follow it with a proof of the minimax rate-optimality of MBS. Let $H\left(\delta, \mathcal{F}, Q_n\right) = \log N\left(\delta, \mathcal{F}, Q_n\right)$ denote the δ-entropy of $\mathcal{F}$ for the $L_2(Q_n)$-metric, where $N\left(\delta, \mathcal{F}, Q_n\right)$ is the δ-covering number and Q_n denotes the empirical measure. Let us suppose that $\mathcal{F}$ is a cone, and that (2.43) holds:

$$H\left(\delta, \{f \in \mathcal{F} : P(f) \leq 1\}, Q_n\right) \leq c_1\delta^{-\alpha},$$

for all $\delta > 0$ and some constants $c_1 > 0$ and $0 < \alpha < 2$. Let $v > \frac{2\alpha}{2+\alpha}$. The same entropy bound holds for the normalized functions when $P(f) + P(f^*) > 0$, i.e. (2.44) holds:

$$H\left(\delta, \left\{\frac{f - f^*}{P(f) + P(f^*)} : f \in \mathcal{F}, P(f) + P(f^*) > 0\right\}, Q_n\right) \leq c_2\delta^{-\alpha}.$$

Furthermore, we assume the errors have sub-Gaussian tails as shown in (2.45):

$$\sup_n \max_{i=1,\ldots,n} K^2\left(\mathbb{E}e^{|\epsilon_i|^2/K^2} - 1\right) \leq \sigma^2.$$

By Lemma 8.4 in [33], with $P(f^*) > 0$, (2.46) holds:

$$\sup_{f \in \mathcal{F}} \frac{|(w, f - f^*)_N|}{\|f - f^*\|_N^{1-\alpha/2} (P(f) + P(f^*))^{\frac{\alpha}{2}}} = O_P(N^{-1/2}).$$

In Theorem 10.1 of [33], an optimal rate of convergence is established assuming (2.44) and (2.45). Since the sub-optimality of our estimator has been quantified in Lemma 1.1, we need only modify Theorem 10.1 for a rate of convergence.

Lemma C.2.1 (Rate of Convergence) *Let* $P(f^*) > 0$ *and*

$$\lambda_N = O_p\left(N^{-\frac{2}{2+\alpha}}\right) \tag{C.9}$$

for $0 < \alpha < 2$. *If*

$$L\left(\Omega\left(\tilde{f}_D\right)\right) \leq L\left(\hat{f}\right) + \Gamma_{N,m}, \tag{C.10}$$

then we have (2.48)*:*

$$\left\|\Omega\left(\tilde{f}_D\right) - f^*\right\|_N^2 = O_p\left(\lambda_N + 6\Gamma_{N,m}\right).$$

Proof Rewriting $L\left(\Omega\left(\tilde{f}_D\right)\right) \leq L\left(\hat{f}\right) + \Gamma_{N,m} \leq L(f^*) + \Gamma_{N,m}$, we get a basic inequality:

$$\left\|\Omega\left(\tilde{f}_D\right) - f^*\right\|_N^2 + \lambda_N P\left(\Omega\left(\tilde{f}_D\right)\right) \leq 2\left(w, \Omega\left(\tilde{f}_D\right) - f^*\right) + \lambda_N P\left(f^*\right) + 2\Gamma_{N,m} \tag{C.11}$$

$$\leq 3\max\left\{2\left(w, \Omega\left(\tilde{f}_D\right) - f^*\right)_N, \lambda_N P(f^*), 2\Gamma_{N,m}\right\}. \tag{C.12}$$

When $\Gamma_{N,m}$ is the maximum, then (2.48) follows:

$$\left\|\Omega\left(\tilde{f}_D\right) - f^*\right\|_N^2 \leq O_P\left(6\Gamma_{N,m}\right).$$

Otherwise, using similar techniques as Theorem 10.2 of [33] gives us

$$\left\|\Omega\left(\tilde{f}_D\right) - f^*\right\|_N^2 \leq O_P\left(\lambda_N\right).$$

Hence,

$$\left\|\Omega\left(\tilde{f}_D\right) - f^*\right\|_N^2 = O_p\left(\lambda_N + 6\Gamma_{N,m}\right).$$

∎

C.2.1 Interpolation Error on the Mesh

Consider $x_i \in [0,1]$. Let $D = (d_1, \ldots, d_m)$ denote the equally spaced grid such that $\delta_m = d_{(i)+1} - d_{(i)} = \frac{1}{m}$. Suppose $f \in C^{k+1}[a,b]$. One approach to fitting an observation x_i over the specified grid D is to use a kth-order Lagrange interpolating polynomial:

$$\Omega_{x_i}(f_D) = \sum_{j=0, j\neq i}^{m} f(d_j) L_{m,j}(x_i), \tag{C.13}$$

where $L_{m,j}(x_i) = \prod_{j'=0}^{m} \frac{x_i - d_j}{d_j - d_{j'}}$. By Theorem 3.3 of [9],

$$f(x_i) = \Omega_{x_i}(f_D) + \frac{f^{(k+1)}(\zeta_i)}{(k+1)!} \prod_{j=0}^{m} (x_i - d_j), \tag{C.14}$$

where $\zeta_i \in [0,1]$. With $f^{(k+1)}(\zeta_i) < K$, it follows that

$$|\Omega_{x_i}(f_D) - f(x_i)| = \frac{f^{(k+1)}(\zeta_i)}{(k+1)!} \prod_{j=0}^{m} |x_i - d_j| \tag{C.15}$$

$$\leq \frac{K}{(k+1)!} (m-1)! \delta_m^{k+1} \tag{C.16}$$

$$\leq K' \delta_m^{k+1}. \tag{C.17}$$

Hence, using a regular grid and a kth-order interpolating polynomial for $\Omega_{x_i}(f_D)$,

$$|\Omega_{x_i}(f_D) - f(x_i)| = O\left(m^{-(k+1)}\right).$$

Appendix D
ADMM OVERVIEW

We discussed extensions of MBS. Since we solve MBS as the solution to a convex optimization problem using the alternating direction method of multipliers (ADMM), MBS readily permits additional constraints. Consider the constrained optimization problem for some parameters $\theta \in \mathbb{R}^n$ in general:

$$\min h(\theta) \quad \text{subject to} \quad \theta \in \mathcal{C}, \tag{D.1}$$

where $h(\cdot)$ is some cost function and $\mathcal{C}$ is a set of constraints we would like to impose on the problem. An ADMM modifies (D.1) by introducing a separable primal variable, α, and including the constraints into the cost of the minimization:

$$\min h(\theta) + g(\alpha) \quad \text{subject to} \quad A\theta - B\alpha = c, \tag{D.2}$$

where $g(\cdot)$ is an indicator function that is non-zero when $\theta \in \mathcal{C}$. If there were more constraints, then we would modify the matrices A and B to include other primal variables, as we have done for the problems of isotonic regression and partially linear additive models. (D.1) is then translated and modified into a single Lagrangian function:

$$\mathcal{L}_\rho(\theta, \alpha, u) = h(\theta) + g(\alpha) + u^\top(A\theta - B\alpha - c) + \frac{\rho}{2}\|A\theta - B\alpha - c\|_2^2,$$

where u is a dual-update step variable and ρ denotes the length of the step. Provided $f(\cdot)$ and $g(\cdot)$ are convex, the solution $(\hat{\theta}, \hat{\alpha}, \hat{u})$, which is the saddle point of $\mathcal{L}_\rho(\theta, \alpha, u)$, has guaranteed global convergence. We used in Section 3 the general form for iterates of ADMM, which we

can state formally here as

$$\theta \leftarrow \underset{\theta}{\operatorname{argmin}}\, h(\theta) + \frac{\rho}{2}\|A\theta - B\alpha - c + u\|_2^2 \tag{D.3}$$

$$\alpha \leftarrow \underset{\alpha}{\operatorname{argmin}}\, g(\alpha) + \frac{\rho}{2}\|A\theta - B\alpha - c + u\|_2^2 \tag{D.4}$$

$$u \leftarrow u + A\theta - B\alpha - c. \tag{D.5}$$

Usually, the minimizations here are solved taking gradients or using projection arguments. For example, in our problems, we have defined $h(\cdot)$ as a squared-loss term, which is smooth, so we can calculate the gradient without much difficulty.

In the penalized regression problems in this dissertation where we solve for $\mathcal{F}_{P,\infty}$ with Sobolev-like norms $P(\cdot)$, $h(\theta)$ has contained the goodness of fit or squared loss term over linear operations on θ or f_D in our case, which preserve convexity. Furthermore, $g(\alpha)$ holds the penalty term, which we have been defining with the convex ℓ_1-norm both for structure and convexity, i.e. $\|\mathcal{D}f_D\|_1$ for some difference matrix $\mathcal{D}$. The iterates of the original MBS problem were given in Chapters 2 and 3 as:

$$f_D \leftarrow \left(O^\top O + \rho(\mathcal{D}^\top\mathcal{D})\right)^{-1}\left(O^\top y + \rho\mathcal{D}^\top(\alpha + u)\right), \tag{D.6}$$

$$\alpha \leftarrow S_{\lambda/\rho}\left(\mathcal{D}f_D - u\right), \tag{D.7}$$

$$u \leftarrow u + \alpha - \mathcal{D}f_D. \tag{D.8}$$

As we introduce further constraints and structure, we need to modify these iterates, as we have shown for the problems of isotonic regression and partially linear additive models. For any additional constraints, we need only preserve convexity of $h(\cdot)$ and $g(\cdot)$ to ensure convergence of iterates. Stopping criteria for the ADMM can be determined from the primal and dual variable residuals. Although each ADMM iteration can be cheap, difficult problems require a large number ($> 1,000$) of iterations. Using varying step-size ρ based on the magnitude of the primal and dual variable residuals, it is possible to reduce the overall

number of iterations. Although, this is only proven empirically.

Appendix E

VIGNETTE TO R PACKAGE, MTV

E.1 Abstract

Total variation is a widely applicable regularization problem that is commonly used in signal and image denoising. We introduce the R package **MTV**, which is used for solving the multivariate total variation problem. Using an alternating direction method of multipliers algorithm (ADMM), we are able to solve problems large in sample size and number of features.

E.2 Introduction

To introduce total variation, we consider the univariate or one-dimensional case first. We begin by observing a response y_i as a function of covariates $x_i \in \mathcal{X}$:

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

where $\mathbb{E}[\epsilon_i] = 0$, $\lessapprox \supset \backslash[\epsilon_i] = \sigma^2 < \infty$ and $f \in \mathcal{F}$ (for $i = 1, \ldots, N$). A total variation problem seeks to estimate f by $\hat{f}$, such that

$$\hat{f} = \min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - f(x_i)\right)^2 + \lambda P(f),$$

where $P(f) = \int |f^{(1)}(t)| \partial t$ (i.e. the total variation norm). **MTV** in the univariate setting uses a mesh based solution, which is described in [26]. For a mesh based solution, we consider a set of m points, $d_1 \leq \ldots \leq d_m$ such that $d_1 \leq x_i$ and $d_m \geq x_i$ for all $x_i \in \mathcal{X}$. The points d_i need not have been observed. We aim to estimate their value $\theta_j = f(d_j)$. Let $\theta_{(i)} = \theta_j$ such

that $d_j \leq x_i < d_{j+1}$ (nearest neighbor). The mesh based solution to the univariate problem approximates $f(x_1), \ldots, f(x_N)$ by the values $\hat{\theta} = (\hat{\theta}_{(1)}, \ldots, \hat{\theta}_{(N)})$:

$$\hat{\theta} = \min_{\theta \in \mathbb{R}^m} \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \theta_{(i)}\right) + \lambda \sum_{j=1}^{m} |\theta_{j+1} - \theta_j| \,,$$

where $\|D_m \theta\|_1 = \sum_{j=1}^{m} |\theta_{j+1} - \theta_j|$ using the ℓ_1-norm and first difference matrix

$$D_m = \begin{bmatrix} -1 & 1 & 0 & \ldots & 0 & 0 \\ 0 & -1 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & ... & -1 & 1 \end{bmatrix} \in \mathbb{R}^{m-1 \times m}.$$

The optimization is quickly solved using an alternative direction method of multipliers (ADMM) algorithm [6]. Furthermore, using $m > N^{\frac{2}{3}}$ (with regular spacing), is enough for statistical optimality of the solution $\hat{\theta}$ [26].

Since the optimization problem encourages adjacent points θ to be the same, we draw piecewise-constant fits as our solution. To begin our demonstration, we generate some data from a piecewise constant function:

```
set.seed(123)
N <- 100
x <- matrix(runif(N),ncol=1)
y <- matrix(pcwise(x) + rnorm(N,0,0.1),ncol=1)
```

To fit a total variation solution to this data, we need only use the `mvtv` function, which by default will perform 5-fold validation and stores the solution path for each λ. Let $m = 20$ for this problem:

```
# Setting a seed for cross-validation procedure
set.seed(123)
m <- matrix(20,ncol=1)
```

```
# Verbose = FALSE to suppress algorithm details
fit <- mvtv(x,y,m, verbose=FALSE)
```

We include an S3 generic `plot` function with additional flags. By default, we plot the predicted surface for the model with minimum cross-validated mean squared error and overlay observed data (Figure E.1):

```
plot(fit)
```

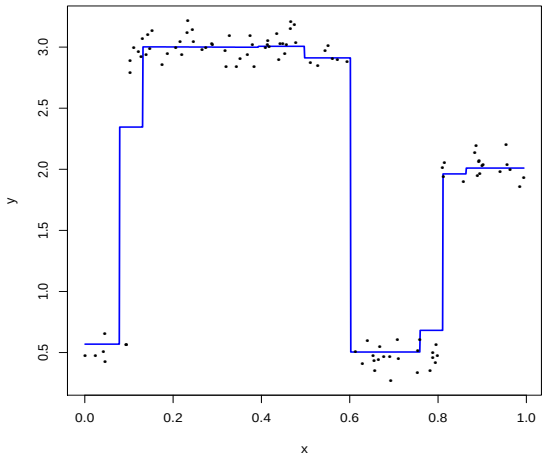


Figure E.1: Total variation solution for piecewise constant function at best cross-validated model.

The `plot` function can also be used to see fits at other values of λ, such as lambda corresponding to 1 standard error rule (for greater parsimony) or an arbitrary number (Figure E.2):

```
par(mfrow=c(1,2))
plot(fit, lambda = fit$lambda.1se)
title(main = "One standard error rule")
plot(fit, lambda = 2.0)
```

```
title(main = "Arbitrary value: 2.0")
```

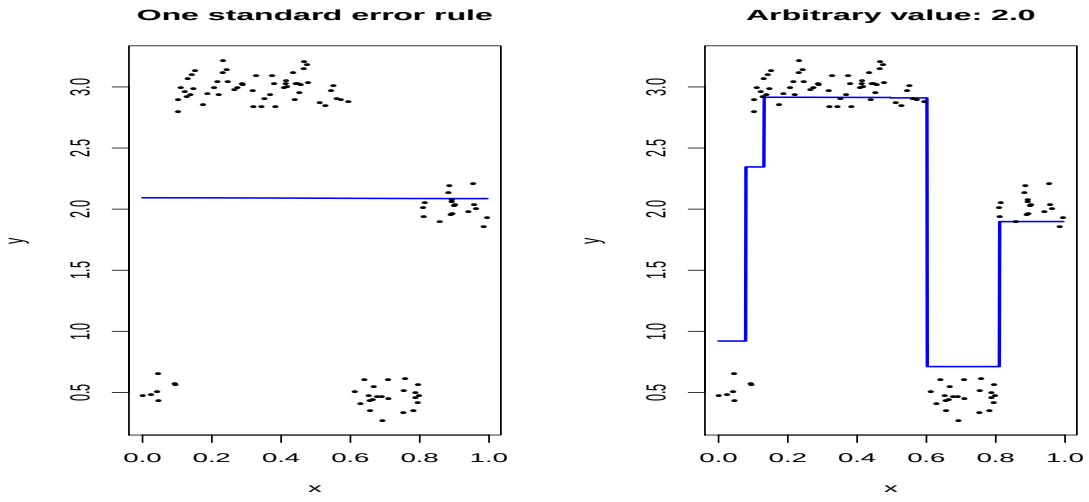


Figure E.2: Total variation solution for piecewise constant function at other solutions.

Aside from the number of knots, the user can also define the mesh itself. In Figure E.3, we show that by setting $m = N$ and $d_1 = x_1, \ldots, d_N = x_N$, we return the fused lasso solution [31, 1].

```
par(mfrow=c(1,2))
# Setting a seed for cross-validation procedure
set.seed(123)
m <- matrix(N, ncol=1)
xsorted <- sort(x, index.return=TRUE)
mesh <- matrix(xsorted$x, ncol = 1)
ysorted <- matrix(y[xsorted$ix], ncol = 1)
# Verbose = FALSE to suppress algorithm details
fl_mvtv <- mvtv(mesh,ysorted,m,mesh = mesh, verbose=FALSE)
plot(fl_mvtv)
title(sub = "(a)")
```

```
library(genlasso)
fl_genlasso <- fusedlasso1d(ysorted)
cvfl <- cv.trendfilter(fl_genlasso, verbose = FALSE)
plot(fl_genlasso, lambda=cvfl$lambda.min, xlab = "x", ylab="y", pch=20, cex=0.5)
title(sub = "(b)")
```

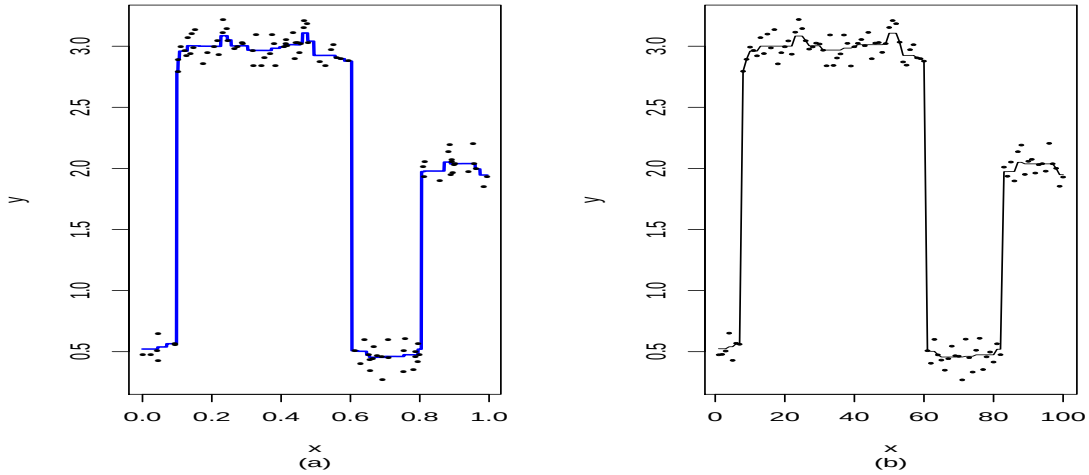


Figure E.3: (a) Fused lasso given by MultivarTV. (b) Fused lasso given by genlasso.

E.3 MTV *for Multivariate Data*

For multivariate data, the total variation problem becomes difficult. Let $(x_1, \ldots, x_p)$ denote p covariates. Furthermore, we define a multi-index of integers $r = (r_1, \ldots, r_p)$, where $r_j \in \{0, 1\}$ and $|r| = \sum_{j=1}^{p} r_j$. In our approach, we approximate the penalty

$$P(f) = \sum_{r \in \{0,1\}^p} \int \left| \frac{\partial^{|r|}}{\partial^{r_1} x_1 \ldots \partial^{r_p} x_p} f(x) \right| \partial x_1 \ldots \partial x_p,$$

where $\{0,1\}^p \equiv \{0,1\} \times \ldots \times \{0,1\}$, i.e. the Cartesian product. In the bivariate setting, our penalty parameter becomes

$$P(f) = \int \left| \frac{\partial}{\partial x_1} f(x_1, x_2) \right| \partial x_1 \partial x_2 + \int \left| \frac{\partial}{\partial x_2} f(x_1, x_2) \right| \partial x_1 \partial x_2 + \int \left| \frac{\partial}{\partial x_1 \partial x_2} f(x_1, x_2) \right| \partial x_1 \partial x_2.$$

When $P(f)$ is finite, solutions have bounded total variation. **MTV** continues to approximate $P(f)$ using differences across a mesh. For details, see [26]. We can fit the multivariate problem using the same ADMM algorithm as in the univariate case, so `mvtv` uses the same algorithm regardless of dimensionality.

E.3.1 Towers Function

We want to fit a multivariate surface as the solution to a total variation problem. One surface that we can use for demonstration is one that resembles four towers on a plain. In Figure E.4, we plot the towers as a wireframe and overlay points generated from the towers function with noise.

```
x1 <- seq(0,1,length.out=40); x2 <- x1

x1x2 <- expand.grid(x1,x2)
y <- matrix(towers(x1x2[,1],x1x2[,2]),nrow=40,ncol = 40)

set.seed(117)
z1 <- runif(100)
z2 <- runif(100)
z3 <- towers(z1,z2)
ynoisy <- z3 + rnorm(length(z3),0,0.5)
scatter3D(z1,z2,ynoisy,theta=30,phi=30,xlab="x1",ylab="x2",zlab="y",
          cex=1,pch=20,surf= list(x = x1,y = x2,z = y,facets=NA,fit = z3))
```

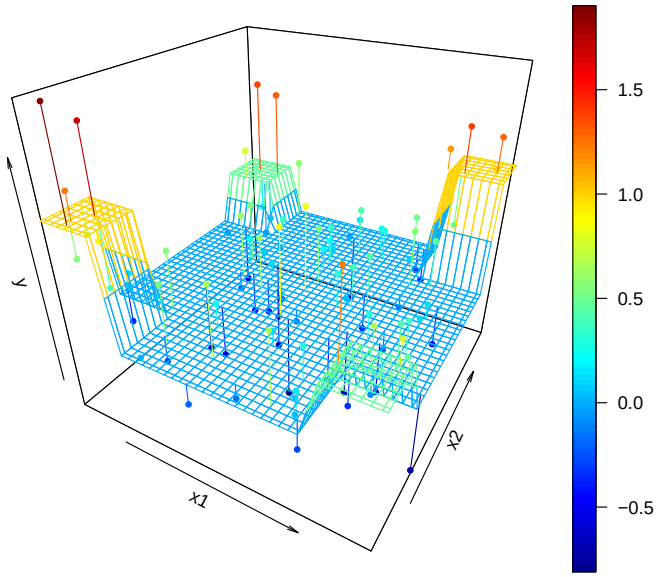


Figure E.4: Plot of N=100 points drawn from Towers function (drawn as mesh) with noise.

The noisy towers can be fit well by a total variation solution. In this example, we will also compare the total variation fit against a thin plate spline, the generalization of smoothing splines [11] fit using the **fields** package by [12]. Based on Figure E.5, we indeed get four towers from our total variation solution. In Figure E.6, we see the thin plate spline solution, which is much more smooth and provides hills instead of towers.

```
set.seed(117)
mym <- floor(length(ynoisy)^{2/3})
mym <- matrix(rep(mym,2),ncol=1)
data <- cbind(z1,z2)
tvmod <- mvtv(data,ynoisy,mym,verbose=FALSE)
plot(tvmod,adddata = TRUE)
```

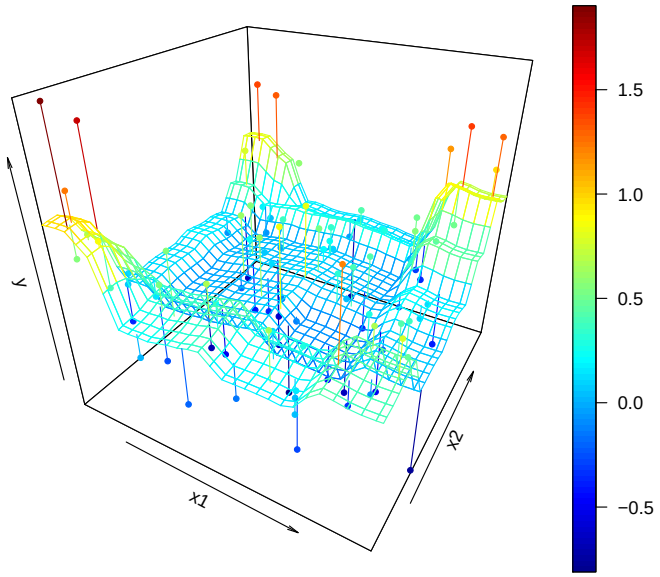


Figure E.5: Total variation solution for noisy towers.

```
fit <- Tps(data, ynoisy)
out.p <- predictSurface(fit, xy = c(1,2))
plot.surface(out.p, type="p")
```

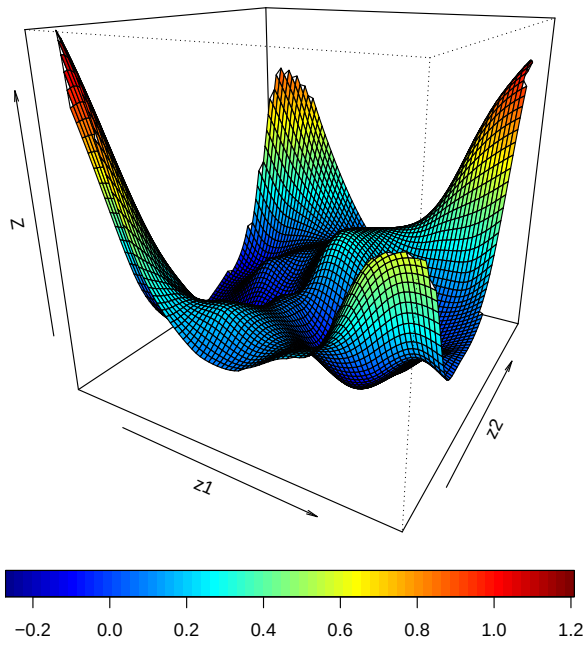


Figure E.6: Thin plate spline solution for noisy towers.

As we increase the sample size say to $N = 2,000$, the total variation solution more accurately predicts the Towers (Figure E.7). The thin plate spline solution takes a large amount of time to calculate, so we do not present it.

```
set.seed(117)
z1 <- runif(2000)
z2 <- runif(2000)
z3 <- towers(z1,z2)
ynoisy <- z3 + rnorm(length(z3),0,0.5)
data <- cbind(z1,z2)

mym <- floor(length(ynoisy)^{1/3})
```

```
mym <- matrix(rep(mym,2),ncol=1)
tvmod <- mvtv(data,ynoisy,mym,verbose=FALSE)
plot(tvmod,adddata = FALSE)
```

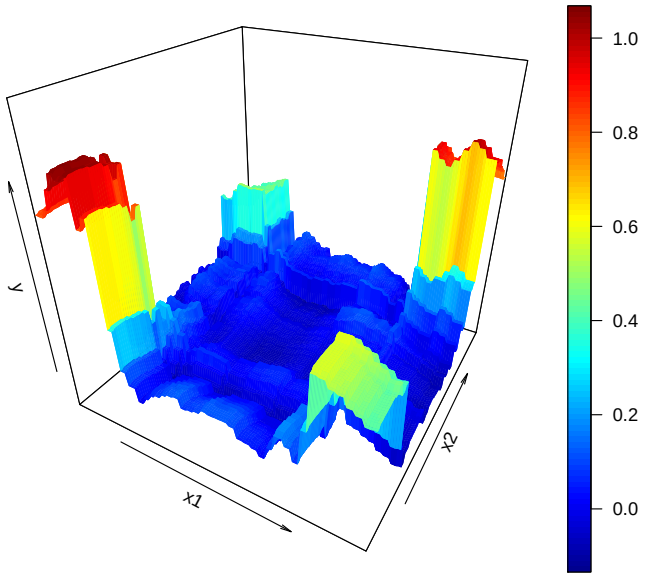


Figure E.7: Total variation solution for noisy towers with many data points.

E.4 Notes and Limitations

The regression model used by `MTV` performs optimally when the distance between the observations and the mesh points is small. In order to minimize that distance, we need to increase the number of knots. `MTV` allows the user to specify m_j knots per feature x_j. It will be best practice to choose a same large number of knots per each feature and allow regularization to smooth the solution. However, as we discuss next, the number of features does impact the functionality of `MTV`.

The primary implementation function `mvtv` accepts any number of features. However, the generic `plot` function does not accept `mvtv`-objects built on $p > 2$. We can still plot the residuals versus fitted values for any size problem using the `plotResiduals` function (Figure E.8).

```
set.seed(117)
n <- 300
p <- 3
m <- 4
data <- matrix(runif(n*p),ncol = p)
y <- matrix(runif(n), ncol=1)
m <- matrix(rep(m,p))
set.seed(123) # cvfold will yield different stuff without seed
tv3 <- mvtv(data,y,m,verbose = FALSE)
plotResiduals(tv3)
```

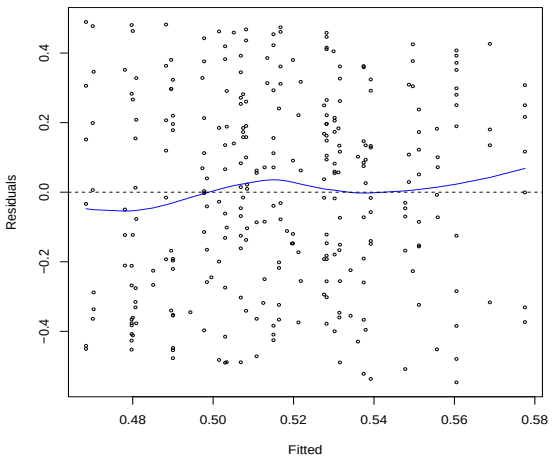


Figure E.8: Residuals plot for perfectly noisy data.

For uniform noisy data regressed on uniform noisy trivariate data, we expect fitted values

at the mean and evenly dispersed residuals, which is what we see with some small wiggling (Figure E.8).

The larger issue with the number of features has to do with scaling. Using the same number of knots for each feature, the number of optimization parameters is m^p. For modest m and large p, the difference matrices (even though they are stored as sparse matrices) become larger than most personal laptops can store. We recommend a large number of knots, $m > N^{\frac{2}{3}}$, for a quality solution, but it may be best practice to use fewer at experimental stages.