ABSTRACT

NORTH, REBECCA MARIE. Topics on Functional Variable Selection with Application to EMG Data Analysis. (Under the direction of Jonathan Stallrich).

Robotic hand prostheses require a prosthesis controller to translate electromyogram (EMG) signals into the user's desired movement. State-of-the-art controllers must undergo extensive training on data from a large number of EMG sensors, whereas a biomechanical model for a single movement degree-of-freedom shows that relatively few forearm muscles are needed to explain hand movement. A prosthesis controller based on such a biomechanical model should then require fewer EMG sensors to produce accurate predictions under a broad set of conditions. Using data collected by the North Carolina State University Department of Biomedical Engineering, Stallrich et al. [2020] proposed a dynamic scalar-on-function model to predict hand velocity with EMG signals and a multi-stage, adaptive penalized regression procedure to simultaneously perform sensor selection and EMG effect estimation. Although the method performed well overall, certain aspects of the data collection, processing, and model fitting raise important research questions.

The EMG data were collected with surface electrodes that were positioned fairly close to one another. This close proximity produces highly correlated signals due to cross-talk between sensors, and strong multicollinearity is known to increase the variance of coefficient estimates. This leads to unstable estimates that may switch signs and be difficult to interpret, reduced statistical power, and difficulty specifying the correct model. Moreover, the data were transformed into a usable format using a "sliding window" process, where consecutive time windows of observations were stacked to form the data matrix. This process also results in autocorrelated responses which may also cause model selection issues if not properly accounted for. Stallrich et al. [2020] thinned the data to reduce this potential correlation, but this thinning reduces the number of observations and potentially loses important information. Since the proposed fitting procedure utilizes penalized linear regression, there are concerns as to how the EMG signals' multicollinearity and the response's potential autocorrelation might affect support recovery and estimation performance. These concerns motivated the first part of this dissertation that develops a number of diagnostic tools that can help practitioners determine the suitability of a given data set for analysis with two penalized regression methods, the lasso and the group lasso, in order to make reliable inference.

The second component of this dissertation considers the form of the penalty used by Stallrich et al. [2020] as well as the chosen optimization algorithm. Specifically, the current

penalty jointly imposes sparsity and smoothness on the EMG effects, which conflates the behaviors and interpretations of the corresponding tuning parameters. With the goal of reducing the number of stages required for accurate selection, estimation, and prediction, two alternative penalties that separately impose sparsity and smoothness are investigated. In order to efficiently fit the model with these penalties, an alternate algorithm is also utilized and shown to greatly reduce overall computational expense.

The final part of this dissertation recognizes that the placement of the EMG sensors results in collection of redundant information. This is a result of electrodes detecting signals from neighboring muscles, not just their intended muscles. Thus, the muscle contractions are viewed as latent factors with the observed EMG signals being surrogate measurements. To leverage the underlying muscle activity more effectively for variable selection, a three-step procedure is extended from the multivariate data setting to the multivariate functional data setting. This procedure consists of a supervision stage that correlates each functional predictor with the response and screens those predictors with sufficiently low correlation; a multivariate functional principal components analysis that uses the estimated components to precondition the response; and a penalized regression of the preconditioned response on the reconstructed predictors to perform a final round of variable selection. A novel strategy for determining the correlation threshold in the supervision stage is proposed, as well as a method for multivariate functional principal components analysis that imposes sparsity in the loading vectors to model each latent feature with a sparse number of predictors.

Topics on Functional Variable Selection with Application to EMG Data Analysis

by
Rebecca Marie North

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Statistics

Raleigh, North Carolina
2021

APPROVED BY:

Ana-Maria Staicu	Luo Xiao
Leonard Stefanski	He Huang

Jonathan Stallrich
Chair of Advisory Committee

DEDICATION

To God be all of the glory.
You made me a promise and you are faithful to fulfill it.

BIOGRAPHY

Rebecca Marie North grew up in Middletown, Ohio. She was greatly encouraged to pursue a STEM career from an early age with participation in the Gifted and Talented Enrichment (GATE) program during elementary school as well as the Be WISE Camp for middle school girls as both a camper and counselor-in-training. After graduating from Madison High School in May 2012, she attended Eastern Kentucky University where she earned a Bachelor of Science in Mathematics in May 2016 with a second major in Statistics. The faculty within the EKU Department of Mathematics and Statistics played a crucial role in preparing Rebecca for graduate school, recognizing her potential early in her undergraduate experience and encouraging her to participate in multiple research-oriented opportunities. She began the Statistics PhD program at North Carolina State University in August 2016 and completed a Master of Statistics in May 2019.

ACKNOWLEDGEMENTS

I first want to thank my advisor, Dr. Jonathan Stallrich, for his guidance and mentorship. He fostered a safe, collegial environment that allowed me to rediscover a desire to do research and finish the PhD. I also want to thank my committee members for their guidance throughout the dissertation process.

There are many other people within the Department of Statistics who deserve my gratitude. First, thank you to Dr. Marie Davidian for appointing me to her training grant, which gave me practical experience in collaborative biostatistics. She has also been an invaluable mentor throughout my graduate career. Next, thank you to Drs. Alyson Wilson, Emily Griffith, Herle McGowan, and Emily Hector for their encouragement, support, and professional development opportunities through the Professional Strategies Working Group. Thank you to Alison McCoy and Lanakila Alexander for the conversations, laughter, support, and chocolate, to Terry Byron for the reliable technical support, and to Shannon Holloway for the computational assistance and Fortran tips. Finally, thank you to my friends and colleagues who provided amazing support in every area of life, including Cheyenne Swanson, Marschall Furman, Kara Martinez, Dana Johnson, Katherine Allen Moyer, Michael McKibben, Sarah Fairfax, Cole Manschot, Kade Young, Julia Holter, Ethan Davis, Nicholas Larson, and Kasia Dorzycka.

During my training at Duke Clinical Research Institute, I had the privilege of working with some incredible people. In particular, I want to thank Karen Pieper and Hillary Mulder for investing in me, believing in me, and supporting me through the challenging times. Also, words cannot express how thankful I am for the mentorship I have received from Gina-Maria Pomann in the Duke BERD core.

Last, but certainly not least, I want to thank the people who know me best, have been with me all along, and who will continue to support me after graduate school. First and foremost, thank you to my husband, Evan, and dog, Carlie, for the unconditional love and support. Without you, I would not have finished. Thank you to my parents, brothers, and in-laws for their unwaivering belief in my abilities and all of the ways they have supported me. In particular, I thank my mom for being the best cheerleader and friend that I could ask for. Finally, thank you to my Lifepointe Church family for all of the prayers, encouragement, and friendships throughout the last five years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation and Outline

Roughly 2.5 million Americans are living with limb loss due to amputation, 60 to 70 percent of whom use prosthetic devices [Uustal, 2020]. Depending on the limb and extent of amputation, functional prosthetic devices can be body-powered or externally-powered, the latter of which can be further classified as electric (button-controlled) or myoelectric (electromyographic signal-controlled). Despite many technological advances in recent years, modern devices still face certain challenges. In particular, a recent review of upper limb prosthesis use and abandonment noted that two of the top three desires for upper limb prosthetics are "real-time, direct, robust and simultaneous control of multiple DoFs [Degrees of Freedom for desired movement] in a natural and intuitive manner...and fast learning" [Cordella et al., 2016]. Satisfying the first desire will allow for easy completion of daily tasks like eating, personal hygiene, and household maintenance, and satisfying the second desire will reduce the amount of time spent training and re-calibrating the device and allow the amputee to achieve a greater sense of normalcy.

Of the currently available prostheses, myoelectric devices provide the most finely-tuned control of the device. For transradial amputees (TRAs)—individuals with amputations between the wrist and elbow—Figure 1.1 illustrates the functionality of a myoelectric device with surface electrodes. Specifically, electric signals travel through neurons from the brain to the forearm and activate muscles for contraction. In able-bodied (AB) subjects, the muscle contraction causes the desired type of hand movement. For TRAs, the limb may be missing but the residual muscles still contract upon initiation of a desired movement since they once had control of the amputated limb [Mercier et al., 2006]. Hence, the electromyogram (EMG) signals measure the electrical activation of the residual muscles and a prosthesis controller decodes the EMGs into movement of the robotic limb. To program the prosthesis controller,

advanced mathematical and statistical tools are required.

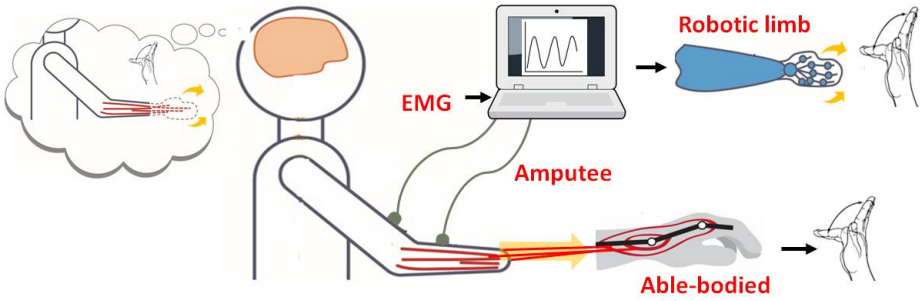

Figure 1.1 Depiction of the biomechanical process for hand and finger movement for an AB subject and a TRA.

State-of-the-art EMG decoders rely on pattern recognition (PR) for prosthesis control [Scheme & Englehart, 2011]. PR assumes that the EMG signals follow specific patterns that are associated with different movement patterns of the hand and wrist. Hence, feature sets are selected from the EMG signals and classified as a movement type. EMG sensors are not placed according to prior anatomical knowledge of specific muscle locations. Instead, the accuracy of motion classification depends on the amount and variety of recorded neural information. Thus, redundant [Huang et al., 2010] or high-density electrodes [Zhou et al., 2007] are frequently used to ensure a sufficient amount of information is collected. However, electrode saturation of the residual limb requires frequent, arduous calibration, makes designing computer hardware for the purpose of real-time EMG processing more challenging, and may hinder performance of the device due to the surplus of potentially noisy signals. Additionally, PR devices are not robust to arm posture modifications or displacement of the muscles during contractions. Although PR devices can achieve multiple DoFs, these limitations need to be resolved to satisfy the prosthesis desires previously described.

As an alternative to PR, Crouch & Huang [2016] work with an EMG-driven musculoskeletal model that uses human physiology to translate EMG signals into movement. Since there are only 20 muscles in the human forearm [O'Rahilly & Müller, 1983] and even fewer that actually control the movement DoFs in the hand, electrode saturation of the limb does not necessarily capture more data. In fact, the number of EMG sensors can be significantly reduced and still yield accurate prediction if the model incorporates relevant knowledge regarding forearm biomechanics [Crouch & Huang, 2017]. However, the biomechanical system in a TRA's forearm is altered by amputation, so standard anatomical knowledge cannot solely be leveraged for sensor placement. Hence, there is need for a dynamic model that can

select a sparse number of necessary EMG sensors, the effect estimates of which will result in accurate predicted movement, and the training of which will be less time-consuming than current myoelectric standards like PR.

To work toward this goal, researchers at the North Carolina State University Department of Biomedical Engineering collected EMG and movement data from an AB subject in a controlled setting. Fifteen EMG surface electrodes were placed on the subject's forearm near muscles that are known to contribute to specific movement DoFs, as depicted in Figure 1.2. The DoFs of interest are finger flexion/extension and wrist flexion/extension, where the finger DoF involves simultaneous, equal movement of all fingers excluding the thumb. The EMG sensors denoted by gray triangles in Figure 1.2 measure muscles that are irrelevant to the DoFs of interest. Along with one externally generated signal that is unrelated to movement (EMG-9), these EMG signals provide a baseline for selection performance.

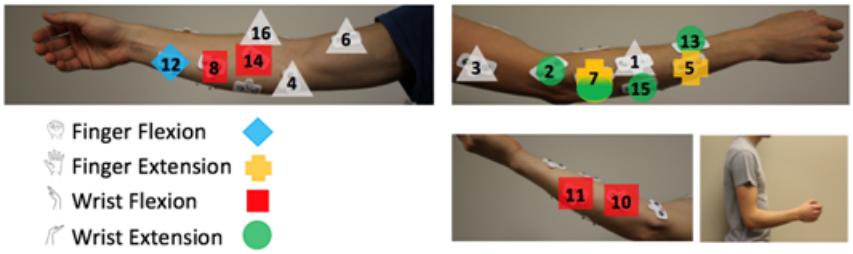

Figure 1.2 Placement of EMG surface electrodes on AB subject with movement DoF denoted by shape and color. Gray triangles denote EMG sensors that measure muscles irrelevant to the DoFs of interest.

With the arm fixed in the position shown in Figure 1.2, the AB subject performed basic hand movements as approximately 30 seconds of EMG and joint angle data were synchronously collected. The hand movements consisted of single DoF movements performed in either a consistent or random pattern. For the consistent pattern, the AB subject alternately performed maximal flexion and extension, allowing the hand to return to a neutral position before performing the opposite movement. The AB subject autonomously determined the movement series for the random pattern. For each DoF, the subject performed each consistent and random movement pattern three times for approximately 30 seconds, yielding six independent data sets. Figure 1.3 visualizes roughly 3 seconds of observed finger movement and EMG signals for each movement pattern. The reader is referred to Stallrich et al. [2020] for further details on data collection and processing.

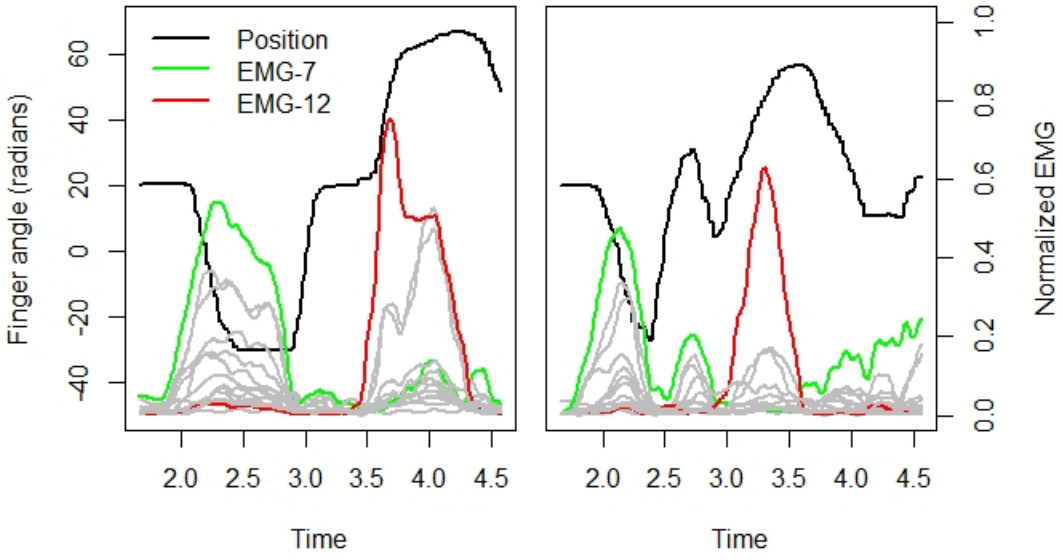

Figure 1.3 A snapshot of position and normalized EMG curves captured from consistent (left) and random (right) finger movement.

Stallrich et al. [2020] proposed a dynamic scalar-on-function linear model that explicitly uses the full EMG signals as opposed to feature sets like PR. The model represents hand velocity y_i as a function of EMG signals $X_{ij}(t)$ measured over recent past time $t \in \mathcal{T} = [-\delta, 0]$ for $\delta > 0$, with bivariate functional effects $\gamma_j(t, z_i)$ that vary over time t and current position z_i. The model is written as

$$y_i = \alpha_0 + \sum_{j=1}^{m} \int_{\mathcal{T}} X_{ij}(t)\gamma_j(t, z_i)dt + \varepsilon_i, \tag{1.1}$$

with intercept α_0 and mean-zero noise ε_i. To fit this model, the observed data underwent a "window" transformation depicted by Figure 1.4 to form functional predictors with common domain $\mathcal{T} = [-\delta, 0]$. For observation i, the concurrent signal and position z_i and previous δ EMG values $x_{ij}, x_{(i-1)j}, ..., x_{(i-\delta)j}$ were extracted to form $X_{ij}(t)$, $t \in \mathcal{T}$. Only those i that contained a complete set of $\delta + 1$ EMG values were included.

Stallrich et al. [2020] proposed an adaptive fitting algorithm to fit this model that performs simultaneous selection and estimation of the EMG effects. Their algorithm was able to detect the few EMG signals 'responsible' for the movement DoFs of interest and provide interpretable effects of each signal, a noted limitation of PR. However, there are certain aspects of the data processing and modeling in Stallrich et al. [2020] that require further

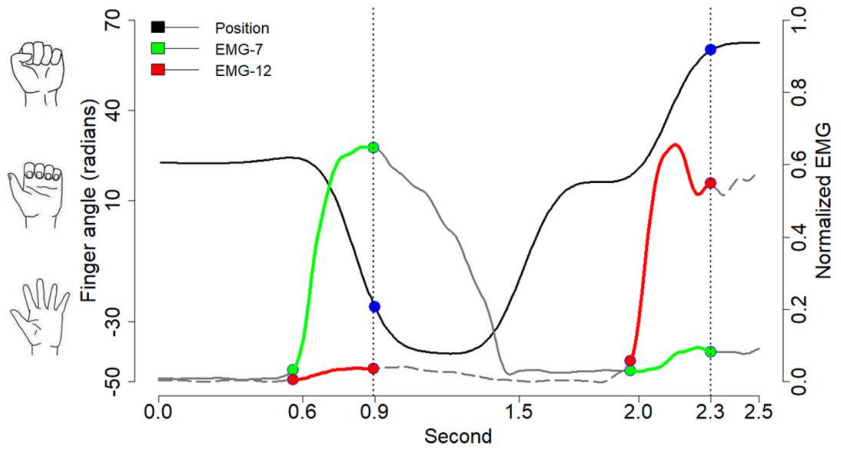

Figure 1.4 Visualization of the data windowing for two EMG signals and finger angles. The blue dots on the black line indicate concurrent position z_i. The green and red line segments denote the current and previous δ values of the corresponding EMG signals.

investigation, three of which are the research projects that comprise this dissertation. The following sections further detail their model and algorithm, introduce the three questionable aspects of their methods, and briefly describe how I resolve these inquiries in the remaining chapters of this dissertation. This chapter concludes with the foundations of functional data and relevant analysis techniques.

1.1.1 Correlated responses

First, consider the windowing process used to restructure the data. Consecutive signal observations in the resulting data set only differ by the first and last time points, so the response values at neighboring i's will be very similar. Thus, the observations exhibit autocorrelation. A similar windowing process is used for PR, but the autocorrelation is not a concern because PR uses feature sets of the signals. The algorithm proposed by Stallrich et al. [2020], to be detailed in the following section, conducts penalized least squares regression with a group lasso penalty, where each EMG signal is a group. Correlation in the response is known to result in underestimated coefficient variances and model selection issues for least squares regression, but its affect on the group lasso is relatively unknown. Stallrich et al. [2020] worked around the autocorrelation by thinning the data, i.e., by keeping every 20th observation, and showed in a simulation study that correlated responses have minimal effect on their method's selection performance. However, a general approach for evaluating robustness in terms of more than just selection is desired.

Additionally, consider the close proximity of EMG sensors in Figure 1.2. It is reasonable

that signals gathered from neighboring sensors would be highly correlated, and Figure 1.5 confirms this occurs. It is well known that high correlation between predictors inflates the standard errors of least squares estimators, which can lead to unstable or uninterpretable estimates, loss of statistical power, and difficulties in model specification. Penalized estimators that use the least squares loss function are negatively affected in similar ways. However, there is currently no practical strategy for evaluating the expected performance of penalized estimators under sufficiently high levels of correlation between the predictors.

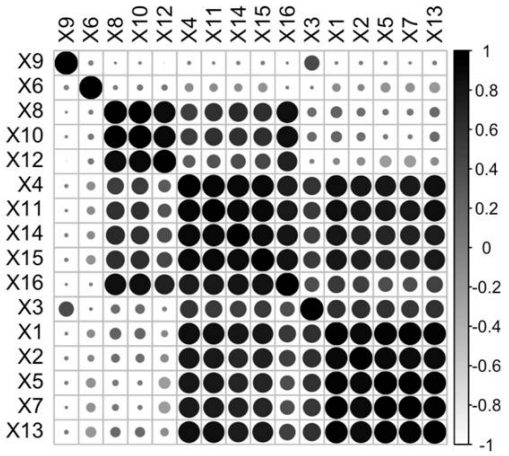

Figure 1.5 Concurrent correlation between EMG signals. Size and shade both denote size of correlation.

In Chapter 2, practical diagnostic tools are developed to study the robustness of the lasso and group lasso to correlated responses and predictors. The existing theory is leveraged for these developments, and the resulting tools will aid practitioners in confidently conducting reliable inference. A diagnostic metric is also developed for cross-validation that can provide strong indication of when an active effect has been incorrectly set to zero. The diagnostic tools for both penalized estimators are demonstrated through simulation. The tools for the lasso are then applied to a diabetes data set, and those for the group lasso are applied to the EMG data previously described.

1.1.2 Conflated tuning parameters

Using the functional linear model in (1.1), Stallrich et al. [2020] simultaneously conduct estimation and selection with an iterative, adaptive selection procedure called Sequential Adaptive Functional Estimation (SAFE). By adding a joint smooth-sparse penalty to the

least squares loss function, similar to Gertheiss et al. [2013], the optimization criterion is written as

$$\sum_{i=1}^{N}\left(y_i - \sum_{j=1}^{m}\int_{\mathcal{T}} X_{ij}(t)\gamma_j(t,z_i)dt\right)^2 + \lambda\sum_{j=1}^{m}(w_j||\gamma_j||^2 + \varphi_t w_{j,t}||\gamma''_{j,t}||^2 + \varphi_z w_{j,z}||\gamma''_{j,z}||^2)^{1/2}. \tag{1.2}$$

The coefficient norm is defined as $||\gamma_j|| = \int_{\mathcal{T}}\int_{\mathcal{Z}}\{\gamma_j(t,z)\}^2 dzdt$ with $\gamma''_{j,t}$ and $\gamma''_{j,z}$ denoting the second partial derivatives of γ_j with respect to t and z, respectively. Each regularization parameter is a positive value, where λ induces sparsity, and φ_t and φ_z induce smoothness of the functional coefficients with respect to t and z, respectively. The adaptive weights $\{w_j, w_{j,t}, w_{j,z}\}$ are calculated as $w_j = 1/||\widetilde{\gamma}_j||$, $w_{j,t} = 1/||\widetilde{\gamma}''_{j,t}||$, and $w_{j,z} = 1/||\widetilde{\gamma}''_{j,z}||$ for initial coefficient estimates $\{\widetilde{\gamma}_j(\cdot,\cdot)\}_{j=1}^m$. The weights are initially set to 1 and the criterion in (1.2) is optimized. Let $\mathcal{J}_1$ denote the subset of EMG signals with non-zero estimates of γ_j, denoted by Γ_1. The adaptive weights are calculated from Γ_1 and (1.2) is refit using only the covariates in $\mathcal{J}_1$ to obtain $\mathcal{J}_2$ and Γ_2. This sequential process may continue for a prescribed number of stages or until some stopping criterion is met. Although SAFE performed well in terms of selection, estimation, and prediction [see Stallrich et al., 2020], five stages of extensive computation were required to achieve convergence in the coefficient estimates. Since one of the top three desires for prosthetic devices is fast learning, there is great interest in determining why SAFE took five stages, if the number of needed stages can be reduced, and if total computation time can be decreased.

One possible explanation for the larger number of stages lies in the penalty's construction in (1.2). Notice that placing λ on the outside of the penalty gives it influence over not only the sparsity part of the penalty, $||\gamma_j||^2$, but also both smoothing components of the penalty, $||\gamma''_{j,t}||^2$ and $||\gamma''_{j,z}||^2$. Thus, instead of enforcing smoothness in the t direction with φ_t inside the square root, the penalization factor is multiplied by λ^2. A similar observation can be made regarding smoothness in the z direction. Theoretically, this can be resolved by distributing λ within the penalty and reparameterizing the smoothing parameters as $\varphi^* = \varphi/\lambda^2$. However, as will be described in Section 1.2.3, optimization problems such as (1.2) can be solved using the group lasso after certain linear approximations and a Cholesky decomposition of the penalty to reparameterize the model matrix and coefficient vector. Use of φ^* would require a Cholesky decomposition for each distinct φ^*, i.e. for each (φ, λ) set, resulting in separate, independent executions of the group lasso algorithm for each distinct set of tuning parameters. Group lasso algorithms built on coordinate descent would hence lose the computational efficiency enjoyed by using "warm starts" across a sequence of λ values

[Friedman et al., 2007], where the solution for a given λ is used as the initial value under the next λ in the sequence. In fact, no existing group lasso algorithm would be able to efficiently consider a full range of λ values for each φ. Thus, the conflation of tuning parameters is a problem in practice.

Additionally, specification of smoothing/tuning parameter grids is a known challenge that is further complicated by the interaction between λ and the smoothing parameters. A naive strategy to combat this challenge in practice is to specify a dense grid of tuning parameters. Although existing group lasso software can manage a denser grid of λ values relatively efficiently, increasing the density of smoothing parameter grids will drastically increase the computation time without an expedient group lasso algorithm. Alternatively, the interaction between tuning parameters could be removed altogether by separating the sparsity and smoothness penalties. In Chapter 3, two such penalties originally proposed by Meier et al. [2009] are investigated. Fitting a penalized regression with these alternate penalties requires the modification of existing group lasso algorithms, which receives much attention in Sections 3.2 and 3.3. The performances of all three penalties are compared through simulations and application to the EMG data.

1.1.3 Latent muscle activity

A second potential explanation for why SAFE requires 5 stages is the redundancy of observed information. Given the close proximity of sensors in Figure 1.2, it is highly likely that sensors detect the contractions of more than one muscle. That is, a given muscle contraction is measured by multiple different EMG sensors to varying degrees. This notion is supported by Figure 1.5, which depicts clusters of highly correlated EMGs. Hence, parsing out specific EMG sensors 'responsible' for a particular movement DoF is quite challenging. In the analysis of amputee data, this issue will be amplified by needing to saturate the limb with sensors to determine the active muscles' new locations. Not only would this increase the likelihood of muscle activity being measured by multiple sensors, but this would greatly inflate the number of observed signals and cause each stage of SAFE to be even slower. Hence, a method that more directly models the muscle activity, accounts for cross-talk between sensors, and can efficiently handle a large number of sensors is desired.

Mathematically speaking, the cross-talk phenomenon can be represented by a latent factor model, where the underlying muscle contractions are the latent factors and the EMG signals are the observed factors. Stallrich et al. [2020] briefly discuss this scenario in the context of their regression model. In particular, for two latent factors $v_1(t)$ and $v_2(t)$ with expected response $E(y_i|v_1(t), v_2(t)) = \int_{\mathcal{T}} v_1(t)\beta_1(t)dt + \int_{\mathcal{T}} v_2(t)\beta_2(t)dt$, suppose the observed

signals can be written as linear combinations of the latent factors, $X_j(t) = \alpha_{j1}v_1(t)+\alpha_{j2}v_2(t)$. Then the model fit with these signals provides the following model equivalency,

$$\sum_{j=1}^{m}\int_{\mathcal{T}} X_j(t)\gamma_j(t)dt = \sum_{j=1}^{m}\int_{\mathcal{T}}(\alpha_{j1}v_1(t)+\alpha_{j2}v_2(t))\gamma_j(t)dt \tag{1.3}$$

$$= \int_{\mathcal{T}} v_1(t)\sum_{j=1}^{m}\alpha_{j1}\gamma_j(t)dt + \int_{\mathcal{T}} v_2(t)\sum_{j=1}^{m}\alpha_{j2}\gamma_j(t)dt, \tag{1.4}$$

where the latent effects are measured as linear combinations of the observed effects, $\beta_1(t) \approx \sum_j \alpha_{j1}\widehat{\gamma}_j(t)$ and $\beta_2(t) \approx \sum_j \alpha_{j2}\widehat{\gamma}_j(t)$. If the latent factors are modeled directly, then the latent effects will be estimated directly and each underlying muscle's contribution to movement will be more interpretable.

A standard method for analyzing latent factor scenarios is principal components analysis (PCA). To ensure the estimated PCs are correlated with the response, the PCA needs to be supervised with information about the response. To accommodate the unique aspects of the EMG application, Chapter 4 extends the methods of Bair et al. [2006] and Paul et al. [2008] to develop a framework for supervised multivariate functional PCA (MFPCA) with sparse regression on a preconditioned response. Specifically, supervision is conducted by measuring the association between each observed signal and the response and screening out signals with weak associations. This screening procedure not only provides supervision for the PCA, but also performs a computationally inexpensive stage of variable selection that can accommodate a large number of observed signals. The threshold for determining the weak associations is derived from the simulation of fake signals and their measured association with the response, a new procedure called Fake Factor Thresholding. One of three MFPCA methods, either MFPCA [Happ & Greven, 2018], sparse MFPCA [Zhang et al., 2018, SMFPCA], or a novel extension of the work by Wang & Tsung [2020], is then used to estimate the principal components (PCs) and subsequently make predictions to precondition the response. Both SMFPCA and the novel method, Sparse Loadings Multivarite FPCA (SLiM FPCA), induce sparsity to further reduce the predictor set. Finally, SAFE is run on the reduced predictor set and preconditioned response to perform a final round of selection. The proposed framework with each of the three FPCA methods is thoroughly studied through simulations and application to the EMG data and compared to the SAFE procedure with and without supervision.

1.2 Functional Data Background

Functional data analysis refers to the study of random continuous functions defined over a closed, compact set, $\mathcal{T}$. There are an infinite number of observable points along the continuum, so functional data are typically not the functions themselves but rather measurements at a finite set of points. We denote such functional data by $\{(W_{ir}, t_{ir}) : i = 1, ..., N; r = 1, ..., n_i\}$, where W_{ir} is the r-th measured value of the i-th curve, $X_i(\cdot)$, at $t_{ir} \in \mathcal{T}$. More concisely, we may write $W_{ir} = X_i(t_{ir}) + \epsilon_{ir}$ where ϵ_{ir} is an independent source of measurement error and the $X_i(\cdot)$ are assumed to be independent and identically distributed (iid) square-integrable functions in $\mathcal{L}_2(\mathcal{T})$—the collection of functions f on $\mathcal{T}$ that satisfy $\int_{\mathcal{T}}\{f(t)\}^2 dt < \infty$—with unknown smooth mean and covariance functions defined point-wise as $\mu(t) = E[X_i(t)]$ and $\Gamma(t, t') = \text{Cov}[X_i(t), X_i(t')]$ for $t, t' \in \mathcal{T}$. The assumption of smoothness, or rather, continuity, of $\mu(t)$ and $\Gamma(t, t')$ are necessary for $X_i(\cdot) \in \mathcal{L}_2(\mathcal{T})$; see Chapter 7 of Hsing & Eubank [2015] for a full mathematical discussion. The covariance function is also assumed to be non-negative definite and symmetric in the sense that $\Gamma(t, t') = \Gamma(t', t)$, similar to the conditions required of a well-defined covariance matrix for multivariate data. The assumed continuity of $\Gamma(\cdot, \cdot)$ yielding smooth changes in $X_i(\cdot)$ along the domain distinguishes functional data and its associated analysis techniques from multivariate and longitudinal data methods.

The set of points $\{t_{ir}\}_{i,r}$ at which the underlying functions are observed can be dense or sparse. For dense data, observations are made on a high-frequency grid at a common set of points $\{t_r\}_r$ that are often equally-spaced. For sparse data, the observed points lie on a low-frequency grid, are often randomly-spaced, and can differ across i. Methods have been developed to analyze functional data of either frequency, but this thesis will focus on densely-observed data. This section will acknowledge some methods commonly utilized for sparse data and more thoroughly describe the dense data methods of interest.

1.2.1 Representations and Smoothing

As noted above, functional observations are realizations of some underlying process that are measured with noise. As detailed in Stallrich et al. [2020], the EMG signals were de-noised with a two-stage filtering process that utilized a fourth order Butterworth zero-phase filter to implement smoothing [Butterworth, 1930]. However, this technique is application specific and not widely used. Instead, one of the following two methods is commonly used to eliminate the noise and construct a functional representation of the underlying process $X_i(\cdot)$. The first is functional principal components analysis (FPCA), an informative way of exploring functional

data and its covariance structure to characterize the underlying process by its most prominent functional features. Specifically, FPCA is often defined in terms of the eigenanalysis of the covariance function $\Gamma(t,t')$. Define the inner product in $\mathcal{L}_2(\mathcal{T})$ as $\langle f(t), g(t)\rangle = \int_{\mathcal{T}} f(t)g(t)dt$ for $f, g \in \mathcal{L}_2(\mathcal{T})$, with the corresponding norm $||f|| = \langle f(t), f(t)\rangle^{1/2}$. Then each functional principal component weight function, or eigenfunction, satisfies the eigenequation

$$\int \Gamma(t,t')v_k(t')dt' = \rho_k v_k(t) \quad \text{such that } ||v_k||^2 = 1, \tag{1.5}$$

for the corresponding eigenvalue ρ_k, and $\langle v_k(t), v_{k'}(t)\rangle = 0$ for $k \neq k'$. Since Γ is continuous, symmetric, non-negative definite, Mercer's Theorem implies the spectral decomposition $\Gamma(t,t') = \sum_{k=1}^{\infty} \rho_k v_k(t)v_k(t')$. Further, the Karhunen-Loève Theorem [Karhunen, 1946; Loève, 1946] gives the FPCA expansion $X_i(t) = \mu(t) + \sum_{k=1}^{\infty} \xi_{ik}v_k(t)$ for FPC scores $\xi_{ik} = \langle X_i(t), v_k(t)\rangle$ with $E(\xi_{ik}) = 0$, $\text{Var}(\xi_{ik}) = \rho_k$, and $E(\xi_{ik}\xi_{ik'}) = 0$ for $k \neq k'$. The ρ_k decrease to zero as k increases and, more strongly, $\sum_{k>k^*} \rho_k \to 0$ as $k^* \to \infty$, causing the scores ξ_{ik} to also converge to zero. The corresponding eigenfunctions thus explain a decreasing amount of variability in the curves, and so the Karhunen-Loève expansion is often truncated at a finite number of terms,

$$X_i(t) \approx \mu(t) + \sum_{k=1}^{K} \xi_{ik}v_k(t). \tag{1.6}$$

The truncation parameter K is commonly determined by a minimum percent variance explained (PVE) threshold, where $PVE = \sum_{k=1}^{K} \rho_k / \sum_{k=1}^{\infty} \rho_k$, or by AIC [Yao et al., 2005].

Estimation of the spectral decomposition and K-L expansion is determined by the sampling frequency. For dense data with each curve observed at a common set of points, the sample covariance can be directly estimated as $\widehat{\Gamma}(t_r, t_{r'}) = \frac{1}{N-1}\sum_{i=1}^{N}[W_{ir} - \widehat{\mu}(t_r)][W_{ir'} - \widehat{\mu}(t_{r'})]$, where $\widehat{\mu}(t_r) = \frac{1}{N}\sum_{i=1}^{N} W_{ir}$ is the estimated mean function at the rth point. Since the data is often observed with noise, $\widehat{\Gamma}(t,t')$ is generally smoothed and diagonalized prior to estimation [Besse & Ramsay, 1986; Staniswalis & Lee, 1998; Xiao et al., 2016; Yao et al., 2005]. The smoothed covariance is then used to calculate the spectral decomposition and the scores are computed via numerical integration or the best linear unbiased predictors [Xiao et al., 2016; Yao et al., 2005]. The estimated eigenfunctions and scores, $\widehat{v}_k(\cdot)$ and $\widehat{\xi}_{ik}$, and the estimated mean function are then substituted in (1.6) to obtain the smoothed representation for $X_i(\cdot)$. For the more challenging case of sparse data, standard estimation approaches are given by Yao et al. [2005], Xiao et al. [2018], James et al. [2000], and Paul & Peng [2009].

Using FPCA to characterize $X_i(t)$ is akin to finding the optimal empirical orthonormal basis. An alternative approach for densely observed data is to use a finite set of pre-

specified basis functions to represent $X_i(t)$. In particular, denote this set of basis functions by $\{B_\ell(t)\}_{\ell=1}^L$. Then write

$$X_i(t) \approx \mu(t) + \sum_{\ell=1}^{L} c_{i\ell} B_\ell(t), \tag{1.7}$$

where $c_{i\ell}$ is the basis coefficient for the ℓth basis function. The choice of basis depends on the data application, but common options include Fourer bases, splines, and wavelets. As discussed in Ramsay & Silverman [2005], Fourier basis functions are preferred for periodic data, spline bases for non-periodic data, and wavelets combine the periodicity of the Fourier basis with the localized flexibility of splines to handle discontinuities or sudden behavior changes well. The number L of basis functions controls the dimensionality of the basis representation and is typically chosen via Akaike Information Criteria (AIC) or cross-validation [Wood, 2003]. The basis coefficients can be estimated by minimizing the least squares loss, $SSE = \sum_{r=1}^{n} [W_{ir} - \widehat{\mu}(t_r) - \sum_{\ell=1}^{L} c_{i\ell} B_\ell(t_r)]^2$, which can be replaced by weighted least squares to account for correlated errors or a smoothness penalty added to impose additional smoothing [Ramsay & Silverman, 2005].

In many instances throughout this thesis, we use orthogonal cubic B-splines [Boor, 2001] in our functional approximations. Popular for representing non-periodic data, spline functions enjoy fast computation and great flexibility. A spline is defined by first dividing the function domain into L sub-intervals separated by $L-1$ breakpoints, or knots. Polynomial functions of a particular order are then specified over each sub-interval with adjacent polynomials having equal values at the knots. The order denotes the number of constants required to define the polynomial and is equal to the polynomial degree plus one. Hence, cubic B-splines have order 4. Although the locations of the knots can be specified according to the relative density of observations at various parts of the domain, we let the knots be equally spaced.

1.2.2 Multivariate Functional Data

This thesis focuses on the case where m random functions are observed at a discrete set of points. That is, consider the observed data set $\{(W_{ijr}, t_{ijr}) : i = 1, ..., N; j = 1, ..., m; r = 1, ..., n_{ij}\}$ with $t_{ijr} \in \mathcal{T}_j$ for $r = 1, ..., n_{ij}$, $i = 1, ..., N$. As before, we assume the observations are noisy measurements of underlying random functions, written as $W_{ijr} = X_{ij}(t_{ijr}) + \epsilon_{ijr}$ with mean zero noise ϵ_{ijr}. Throughout this thesis we work with functions densely observed at the same number of time points, allowing for the simplified notation $\boldsymbol{W}_{ir} = \boldsymbol{X}_i(\boldsymbol{t}_r) + \boldsymbol{\epsilon}_{ir}$ with $\boldsymbol{W}_{ir} \in \mathbb{R}^m$, $\boldsymbol{t}_r = (t_{r1}, ..., t_{rm})^T \in \mathcal{T} = \mathcal{T}_1 \times \cdots \times \mathcal{T}_m$, vector functions $\boldsymbol{X}_i(\boldsymbol{t}_r) = (X_{i1}(t_{r1}), ..., X_{im}(t_{rm}))^T$ and noise vector $\boldsymbol{\epsilon}_{ir} \in \mathbb{R}^m$. We assume each $X_{ij}(\cdot)$ is a square-integrable function in $\mathcal{L}_2(\mathcal{T}_j)$, $j = 1, ..., m$, and the vector functions $\boldsymbol{X}_i(\boldsymbol{t}_i)$ are independent

realizations of the random vector function $\boldsymbol{X}(\boldsymbol{t}) = (X_1(t_1), ..., X_m(t_m))^T$ that lie in the Hilbert space $\mathbb{H} = \mathcal{L}_2(\mathcal{T}_1) \times \cdots \times \mathcal{L}_2(\mathcal{T}_m)$. As discussed in Section 1.1.1, the independence assumption is violated during the sliding window construction of the EMG curve observations, and so the curve vectors are thinned to reduce the effects of correlation. Define the mean vector function as $\boldsymbol{\mu}(\boldsymbol{t}) = E[\boldsymbol{X}_i(\boldsymbol{t}_i)] = (E[X_1(t_1)], ..., E[X_m(t_m)])^T$ and the covariance matrix as $\boldsymbol{\Gamma}(\boldsymbol{t}, \boldsymbol{t}') = \{\Gamma_{jl}(t_j, t_l)\}$ for $\boldsymbol{t}, \boldsymbol{t}' \in \mathcal{T}$ with elements $\Gamma_{jl}(t_j, t_l') = \text{Cov}(X_j(t_j), X_l(t_l'))$ for $t_j \in \mathcal{T}_j$ and $t_l' \in \mathcal{T}_l$.

The methods described for a single function can be applied or extended to the multivariate functional case. For instance, since smoothing via basis expansions is applied to each individual curve for dense observations, it can be applied without modification to multiple, densely observed functions. Sparsely observed functions can be smoothed using the pooled FPCA approach described in Yao et al. [2005]. Alternatively, FPCA has been extended to the multivariate setting either by achieving a multivariate functional Karhunen-Loève representation of the data [Chiou et al., 2014; Happ & Greven, 2018; Jacques & Preda, 2014; Ramsay & Silverman, 2005] or by repeatedly applying PCA for each point $t \in \mathcal{T}$ and interpolating to build the functional principal components [Berrendero et al., 2011]. Of the methods that aim for a multivariate Karhunen-Loève structure, Happ & Greven [2018] give the most general version that allows each function to lie on a different domain whereas the other methods require all functions to be observed on the same interval.

With each $X_j(\cdot) \in \mathcal{L}_2(\mathcal{T}_j)$, the vector function $\boldsymbol{X}(\boldsymbol{t}) \in \mathbb{H} := \mathcal{L}_2(\mathcal{T}_1) \times \cdots \times \mathcal{L}_2(\mathcal{T}_m)$ defined by the inner product

$$\langle \boldsymbol{f}, \boldsymbol{g} \rangle_{\mathbb{H}} = \sum_{j=1}^{m} \langle f_j, g_j \rangle = \sum_{j=1}^{m} \int_{\mathcal{T}_j} f_j(t_j) g_j(t_j) dt_j \tag{1.8}$$

for $\boldsymbol{f}, \boldsymbol{g} \in \mathbb{H}$, $\boldsymbol{f} = (f_1, ..., f_m)^T$, and corresponding norm $||\boldsymbol{f}||_{\mathbb{H}} = \langle \boldsymbol{f}, \boldsymbol{f} \rangle_{\mathbb{H}}^{1/2}$. Define the covariance operator $\mathcal{G} : \mathbb{H} \to \mathbb{H}$ as

$$(\mathcal{G}\boldsymbol{f})(\boldsymbol{t}) = \int \boldsymbol{\Gamma}(\boldsymbol{t}, \boldsymbol{t}') \boldsymbol{f}(\boldsymbol{t}') d\boldsymbol{t}' = \begin{pmatrix} \langle \boldsymbol{\Gamma}_1(t_1, \cdot), \boldsymbol{f} \rangle_{\mathbb{H}} \\ \vdots \\ \langle \boldsymbol{\Gamma}_m(t_m, \cdot), \boldsymbol{f} \rangle_{\mathbb{H}} \end{pmatrix}, \tag{1.9}$$

where $\boldsymbol{\Gamma}_j(t_j, \cdot) = (\Gamma_{j1}(t_j, \cdot), ..., \Gamma_{jm}(t_j, \cdot))^T$ and $t_j \in \mathcal{T}_j$. Assuming $\mathcal{G}$ is a compact positive operator on $\mathbb{H}$, the Hilbert-Schmidt Theorem [Reed & Simon, 1980] gives the existence of a complete orthonormal basis of eigenfunctions $\boldsymbol{\psi}_k(\boldsymbol{t}) = (\psi_{k1}(t_1), ..., \psi_{km}(t_m))^T \in \mathbb{H}$, $k = 1, 2, ...$, of $\mathcal{G}$ such that $\mathcal{G}\boldsymbol{\psi}_{\boldsymbol{k}} = \rho_k \boldsymbol{\psi}_k$ and $\rho_k \to 0$ as $k \to \infty$. The Spectral Theorem yields the decomposition $\mathcal{G}\boldsymbol{f} = \sum_{k=1}^{\infty} \rho_k \langle \boldsymbol{f}, \boldsymbol{\psi}_k \rangle_{\mathbb{H}} \boldsymbol{\psi}_k$ for all $\boldsymbol{f} \in \mathbb{H}$, and by a multivariate version of

Mercer's Theorem it holds with absolute and uniform convergence that

$$\text{Cov}(X_j(t_j), X_l(t_l')) = \Gamma_{jl}(t_j, t_l') = \sum_{k=1}^{\infty} \rho_k \psi_{kj}(t_j)\psi_{kl}(t_l'). \tag{1.10}$$

Finally, the Multivariate Karhunen-Loève Theorem gives the representation $\boldsymbol{X}(\boldsymbol{t}) = \sum_{k=1}^{\infty} \xi_k \boldsymbol{\psi}_k(\boldsymbol{t})$ for $\boldsymbol{t} \in \mathcal{T}$ and random variables $\xi_k = \langle \boldsymbol{X}(\boldsymbol{t}), \boldsymbol{\psi}_k(\boldsymbol{t}) \rangle_{\mathbb{H}}$ with mean zero and $\text{cov}(\xi_k, \xi_{k'}) = \rho_k \delta_{kk'}$ where $\delta_{kk'} = 1$ if $k = k'$ and zero otherwise. As in the univariate case, the leading eigenfunctions portray the most important features of $\boldsymbol{X}(\boldsymbol{t})$, so the Karhuen-Loève expansion is truncated in practice at K dimensions, $\boldsymbol{X}(\boldsymbol{t}) \approx \sum_{k=1}^{K} \xi_k \boldsymbol{\psi}_k(\boldsymbol{t})$.

The estimation details are provided in Happ & Greven [2018] with a description of the relationship between univariate and multivariate FPCA for finite Karhunen-Loève decompositions. Happ & Greven [2018] note that, since the first step of the estimation procedure is univariate FPCA for each element $X_j(t_j)$ of the vector function $\boldsymbol{X}(\boldsymbol{t})$, MFPCA can be applied to both dense and sparse functional data. As noted by Li et al. [2020], though, MFPCA may not perform well for sparse functional data because the shrinkage of univariate FPC scores toward zero for sparsely observed data can lead to poorly capturing cross-correlations between functions. Li et al. [2020] propose a covariance estimation procedure based on tensor-product B-splines with a smoothness penalty to avoid overfitting and note specific computational and theoretical benefits over the methods previously developed for sparse multivariate functional data by Zhou et al. [2008] and Chiou et al. [2014].

1.2.3 Functional Linear Regression

As in the EMG application described at the beginning of this chapter, functional linear regression [Ramsay & Dalzell, 1991] can be used to study the relationship between a scalar response y_i and m functional predictors $\{X_{ij}(t_j)\}_{j=1}^{m}$. To do so, the following functional linear model is used,

$$y_i = \alpha + \sum_{j=1}^{m} \int_{\mathcal{T}_j} X_{ij}(t)\gamma_j(t)dt + \varepsilon_i, \tag{1.11}$$

where the smooth, square-integrable coefficients $\gamma_j(t)$ quantify the effects of $X_{ij}(t)$ on the mean response $\mu_i = \alpha + \sum_j \int_{\mathcal{T}_j} X_{ij}(t)\gamma_j(t)dt$ for $j = 1, ..., m$, and the errors ε_i follow a zero-mean distribution with variance σ^2.

To fit the model in (1.11), we start by approximating it with a linear model. First, we assume the effects can be expressed as a linear combination of known basis functions, $\gamma_j(t) \approx \sum_{\ell=1}^{L} \beta_{j\ell}\omega_{j\ell}(t)$. Throughout this dissertation, we use the same set of orthogonal cubic B-spline basis functions $\{\omega_\ell(t)\}_{\ell=1}^{L}$ for each coefficient, but present the general framework

here. Next, we compute the integral with a Riemann sum

$$\int_{\mathcal{T}_j} X_{ij}(t)\gamma_j(t)dt \approx \sum_{\ell=1}^{L}\left\{\sum_{r=1}^{n}\Delta_{jr}X_{ij}(t_{jr})\omega_{j\ell}(t_{jr})\right\}\beta_{j\ell} = \widetilde{\boldsymbol{X}}_{ij}^T\boldsymbol{\beta}_j, \tag{1.12}$$

where $\boldsymbol{\beta}_j = (\beta_{j1}, ..., \beta_{jL})^T$, $\widetilde{\boldsymbol{X}}_{ij} = (\widetilde{X}_{ij1}, ..., \widetilde{X}_{ijL})^T$, $\widetilde{X}_{ij\ell} = \sum_r \Delta_{jr}X_{ij}(t_{jr})\omega_{j\ell}(t_{jr})$, and $\Delta_r = t_r - t_{r-1}$ denotes the distance between two consecutive time points. Thus, the functional model in (1.11) can be approximated by the linear regression model, $y_i = \alpha + \sum_{j=1}^{m}\widetilde{\boldsymbol{X}}_{ij}^T\boldsymbol{\beta}_j + \varepsilon_i$. The eigenfunctions from performing FPCA on $\{X_{ij}(t)\}_i$ could alternatively be used in the expansion of both $X_{ij}(t)$ and $\gamma_j(t)$, which would remove the need for a Riemann sum approximation. However, this approach assumes that the effects share the same principal directions of variation as the predictors, which may not be a valid assumption in all cases.

Fitting a simple least squares model implies that the smoothness of $\gamma(t)$ is strictly controlled by the number of basis functions L. This value must be large enough to capture the important features of $\gamma(t)$, but small enough so that $\gamma(t)$ retains interpretability. In practice, however, this balance is difficult to achieve and L must often assume a low value, which forces $\gamma(t)$ to lie in a low-dimensional space and risks missing important features. Cardot et al. [2003] introduced a more flexible approach where L assumes a large value to provide a rich set of basis functions, and the roughness of $\gamma(t)$ is penalized by an appropriate derivative penalty. In this dissertation, the second derivative is used, such that the penalty $\varphi\sum_{j=1}^{m}||\gamma_j''||^2$ is added to the least squares loss where $\gamma_j''(t) = \delta^2\gamma_j(t)/\delta t^2$. The degree of penalization is controlled by the tuning parameter $\varphi > 0$, such that larger values of φ encourage smoother, more linear effects and smaller values allow for more wiggly effects. The 'optimal' value of φ is commonly determined via generalized cross-validation or V-fold cross-validation, where we use the latter throughout this dissertation.

As in the EMG application, there are many scenarios where we also desire to impose sparsity in the coefficients. That is, we want to determine a set of important predictors with non-zero functional effects and set the other effects to zero. Gertheiss et al. [2013] proposed the *sparsity-smoothness penalty*, $P_{\lambda,\varphi} = \lambda(||\gamma_j||^2 + \varphi||\gamma_j''||^2)^{1/2}$, which Meier et al. [2009] originally introduced for variable selection in high-dimensional additive modeling. Larger values of the regularization parameter $\lambda > 0$ impose greater levels of sparsity. Using a rich set of orthogonal basis functions to approximate $\gamma_j(t)$, $j = 1, ..., m$, the penalty can be written as

$$P_{\lambda,\varphi}(\gamma_j) = \lambda\{\boldsymbol{\beta}_j^T(\boldsymbol{I} + \varphi\boldsymbol{\Omega}_j)\boldsymbol{\beta}_j\}^{1/2}, \tag{1.13}$$

where $\boldsymbol{\Omega}_j$ is the $L \times L$ matrix with (ℓ, ℓ') element $(\boldsymbol{\Omega}_j)_{\ell,\ell'} = \int \omega_{j\ell}(t)\omega_{j\ell'}(t)dt$ for $\ell, \ell' = 1, ..., L$. The penalty can be rewritten even more succinctly as $P_{\lambda,\varphi} = \lambda(\boldsymbol{\beta}_j^T\boldsymbol{Q}_{j\varphi}\boldsymbol{\beta}_j)^{1/2}$, the general

group lasso penalty [Yuan & Lin, 2006], where $\boldsymbol{Q}_{j\varphi} = \boldsymbol{I} + \varphi\boldsymbol{\Omega}_j$ is symmetric and positive definite. This leads to the optimization criterion by which α and $\gamma_j(t)$ are estimated, written as $\widehat{\alpha}$ and $\widehat{\gamma}_j(t) = \sum_{\ell=1}^{L} \widehat{\beta}_{j\ell}\omega_{j\ell}(t)$ for $j = 1, ..., m$, where $\widehat{\alpha}$ and the $\widehat{\boldsymbol{\beta}}_j$'s minimize

$$\sum_{i=1}^{N}(y_i - \alpha - \sum_{j=1}^{m} \widetilde{\boldsymbol{X}}_{ij}^T\boldsymbol{\beta}_j)^2 + \lambda\sum_{j=1}^{m}(\boldsymbol{\beta}_j^T\boldsymbol{Q}_{j\varphi}\boldsymbol{\beta}_j)^{1/2}. \tag{1.14}$$

To use existing group lasso software to optimize this expression, we must perform a simple reparameterization. Let $\boldsymbol{Q}_{j\varphi} = \boldsymbol{R}_{j\varphi}\boldsymbol{R}_{j\varphi}^T$ be the Cholesky decomposition of the penalty matrix where $\boldsymbol{R}_{j\varphi}$ is a lower triangular matrix. Then the general group lasso can be reparameterized as

$$\sum_{i=1}^{N}(y_i - \sum_{j=1}^{m} \boldsymbol{W}_{ij}^T\widetilde{\boldsymbol{\beta}}_j)^2 + \lambda\sum_{j=1}^{m} ||\widetilde{\boldsymbol{\beta}}_j||_2,$$

for $\boldsymbol{W}_{ij} = \boldsymbol{R}_{j\varphi}^{-1}\widetilde{\boldsymbol{X}}_{ij}$ and $\widetilde{\boldsymbol{\beta}}_j = \boldsymbol{R}_{j\varphi}\boldsymbol{\beta}_j$, where $||\cdot||_2$ is the Euclidean norm in $\mathbb{R}^L$.

Chapter 2

Lasso and Group Lasso Diagnostics

2.1 Introduction

For the linear model $\boldsymbol{y} = \mathbf{1}\beta_0^* + \boldsymbol{X}\boldsymbol{\beta}^* + \boldsymbol{e}$, the statistical properties of estimators for $\boldsymbol{\beta}^*$ depend on the $N \times p$ model matrix, $\boldsymbol{X}$, and distributional assumptions about the error vector, $\boldsymbol{e}$. Understanding these relationships and the properties' robustness to assumption violations is then paramount for reliable inference. For least-squares inference, this understanding has led to diagnostic tools such as variance inflation factors [James et al., 2013] that are commonly employed in practice. With high-dimensional problems increasing in popularity, so have penalized least-squares estimation and inference. The goal of this chapter is to develop comparable diagnostic tools for the lasso and group lasso so practitioners can make sound inference from their data.

The ordinary least squares (OLS) estimator, $\widehat{\boldsymbol{\beta}}^{OLS}$, minimizes the sum-of-squares loss function $||\boldsymbol{y} - \mathbf{1}\beta_0 - \boldsymbol{X}\boldsymbol{\beta}||_2^2$. If $\widehat{\boldsymbol{\beta}}^{OLS}$ is unique, it is unbiased when the mean model, $\boldsymbol{X}\boldsymbol{\beta}$, is correctly specified and $\mathbb{E}(\boldsymbol{e}) = \mathbf{0}$. Additionally, $\widehat{\boldsymbol{\beta}}^{OLS}$ has the smallest variance among all linear unbiased estimators when $\text{Var}(\boldsymbol{e}) = \boldsymbol{\Sigma} = \sigma^2\boldsymbol{I}$. Still, standard inferences based on $\widehat{\boldsymbol{\beta}}^{OLS}$ will suffer under inflated standard errors due to strong multicollinearity of the columns of $\boldsymbol{X}$. For example, $\widehat{\beta}_j^{OLS}$ may be estimated with the incorrect sign, power may be significantly reduced, and correct model specification may become more challenging. As standard errors are commonly reported in statistical output, this inflation is easy to spot and is a common diagnostic discussed during statistical training. Penalized estimators are touted for their more stable behavior under multicollinearity, yet they are not completely immune to it. In our experiences, practitioners rarely employ one of the recent inference methods developed for the lasso/group lasso (see Chapter 6 of [Hastie et al., 2015]) that would alert them to correlation issues. An exception is inference done indirectly through cross-validation or holdout sample prediction, as variability in the lasso estimator should be realized through

poor predictive performance. Not all data sets are amenable to cross-validation, such as data collected from a designed experiment which has increased in popularity [Mee et al., 2017]. It is important then to develop, for high-dimensional problems, diagnostics that are easy to compute and interpret and that will alert practitioners to data quality issues that hinder the statistical performance of the lasso and group lasso.

When $\boldsymbol{\Sigma} \neq \sigma^2 \boldsymbol{I}$, $\hat{\boldsymbol{\beta}}^{OLS}$ generally does not have the minimum variance property, which is held by the generalized least squares (GLS) estimator, $\widehat{\boldsymbol{\beta}}^{GLS}$.The discrepancy between the two estimators generally grows as $\boldsymbol{\Sigma}$ moves farther from $\sigma^2 \boldsymbol{I}$, but there are some combinations of $\boldsymbol{\Sigma}$ and $\boldsymbol{X}$ where $\widehat{\boldsymbol{\beta}}^{OLS} = \widehat{\boldsymbol{\beta}}^{GLS}$ [Puntanen & Styan, 1989]. The lasso and group lasso are related to OLS in that they penalize the same loss function, but this by no means implies that these estimators have been constructed around the standard inference assumption $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}$. Instead, the assumption is often made in the lasso literature for technical convenience, producing straightforward bounds on probabilities of events such as support recovery. Alquier & Doukhan [2011] and Kaul [2014] studied the robustness of the lasso estimator to correlated normal or non-normal errors in terms of $\mathcal{L}_1$-error, $||\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*||_1$, and squared-prediction loss, $||\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{X}\boldsymbol{\beta}^*||_2^2$. Analytically studying robustness properties of the lasso and group lasso is challenging due to the lack of closed-form expressions except for special cases. This chapter provides a simulation-based approach for practitioners to determine $\mathcal{L}_2$-error robustness of the lasso and group lasso to different error assumptions for the data at hand.

An important question concerning robustness is whether a GLS-based loss function that incorporates knowledge about the true $\boldsymbol{\Sigma}$ would improve the statistical efficiency of these penalized estimators. Econometrics literature provides many extensions and modifications of the lasso to linear models with autoregressive errors [Medeiros & Mendes, 2015; Wang et al., 2007; Wong et al., 2020; Yoon et al., 2013, 2017], and Gupta [2012] investigated properties of a GLS lasso. Details of our version of GLS lasso may be found in Appendix A.1. In practice, such an estimator would require an estimate of $\boldsymbol{\Sigma}$ which is challenging in high dimensions. Moreover, Jia & Rohe [2015] noted that this GLS lasso underperforms the usual lasso in terms of sign recovery for a high dimensional $\boldsymbol{X}$ and heterogenous variances. Therefore we are less concerned with developing a theory around the GLS lasso and advocate that the usual lasso be used instead.

We now demonstrate the aforementioned GLS lasso phenomenon with a short covariance robustness study with some toy examples. Suppose $\boldsymbol{X}$ is a 24×20 matrix derived from an appropriately scaled (see Section 2.2) Hadamard matrix with the intercept column and three other columns removed. Such $\boldsymbol{X}$ are common in the experimental design literature and, having uncorrelated columns, achieve the smallest possible variance for the OLS estimators. For $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}$, the lasso estimator has a closed-form expression that allows a closed-form

derivation of its expected value and variance (see Appendix A.2). We are then able to study the robustness of its mean-squared error ($MSE = \mathbb{E}(||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*||_2^2)$) to different covariance structures without having to perform a computationally intensive simulation study. We also calculated via simulation the MSE of the GLS-based lasso estimator and compared both to the MSE of $\widehat{\boldsymbol{\beta}}^{OLS}$ and $\widehat{\boldsymbol{\beta}}^{GLS}$. Note the unpenalized estimators' MSE's do not depend on $\boldsymbol{\beta}^*$, but they do for the two lasso estimators. We considered

$$\boldsymbol{\beta}^* = (4, 4, 4, 3, 3, 2, 2, 1, 0, 0, 0, 0, -1, -1, -2, -3, -4, -4, -5, -5)$$

and Normally-distributed errors under different $\boldsymbol{\Sigma}$: (1) diagonal and constant with $\sigma^2 = 1$, (2) completely symmetric (CS) with correlation $\rho = 0.7$, (3) first-order autoregressive ($AR(1)$) with correlation $\rho = 0.7$, (4) one-banded Toeplitz with correlation $\rho_T = 0.3$, and (5) diagonal and heterogeneous variances generated by a gamma distribution with shape and rate parameters equal to 0.5. As explained in Section 2.3, the different $\boldsymbol{\Sigma}$ are re-scaled to make the situations comparable. We also generated a high-dimensional 100×1000 $\boldsymbol{X}$ and $\boldsymbol{\beta}$ following [Jia & Rohe, 2015] to compare the lasso and GLS lasso. Figure 2.1 shows the MSE's across a range of values of tuning parameter values, λ.

The results from the Hadamard example show how the OLS and lasso's MSE patterns are consistent across scenarios. There also exist λ's where the lasso's MSE meets or dominates the MSE of the OLS and GLS estimators, while the MSE of the GLS-lasso is superior to the lasso, especially for the heterogeneous variance case. The observed behavior can be explained by both the lasso's regularization properties and the fact that we are dealing with a well-conditioned model matrix. The high-dimensional example does not have such a well-conditioned matrix but the lasso's MSE is still consistent across covariance structures and the GLS lasso's MSE generally improves on the usual lasso's MSE. The notable exception is the heterogeneous variance case, which was discussed by Jia & Rohe [2015], although their attention was on sign recovery, not MSE. Based on the results of these toy examples, we will focus our attention on the robustness of the lasso and group lasso estimators to different $\boldsymbol{\Sigma}$ in terms of MSE. A challenge with studying this robustness is its evident dependence on $\boldsymbol{\beta}^*$. Indeed, there are many alternative values of $\boldsymbol{\beta}^*$ for our high-dimensional toy example that would make the GLS-lasso's MSE always outperform the lasso's MSE. When studying robustness and determining it practically, we would rather not have to worry about restricting our attention to the unknown value of $\boldsymbol{\beta}^*$.

This paper is one of the first to advocate and develop tools for the investigation of model matrix conditions when performing an analysis under the lasso or group lasso. Our methods leverage existing results on the estimators' support recovery and $\mathcal{L}_2$-error properties

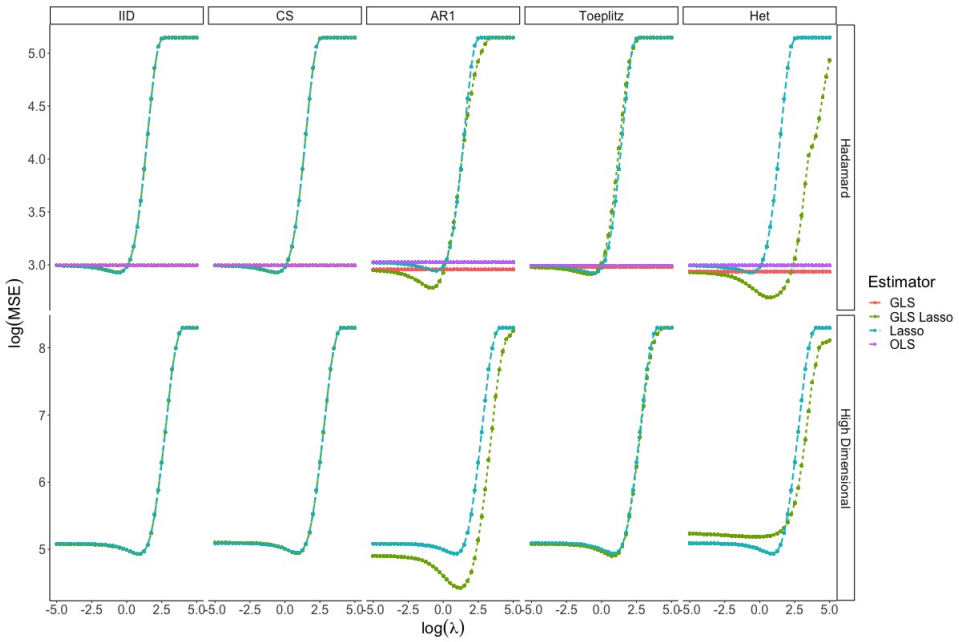

Figure 2.1 Plots of MSE versus $\log(\lambda)$ between the lasso, OLS, and GLS estimators for different structures of $\boldsymbol{\Sigma}$. The top row correspond to an $\boldsymbol{X}$ equal to a 24×20 Hadamard matrix. The bottom row is for a 100×1000 $\boldsymbol{X}$ matrix with iid Normal entries.

to motivate diagnostic measures for a fixed $\boldsymbol{X}$ and potential covariance structures. We also propose a straightforward cross-validation diagnostic that is a strong indicator that the estimators incorrectly set active effects to 0. The major advantage of our approaches is that they do not require specific values for $\boldsymbol{\beta}^*$, which would be needed for a detailed simulation study. Our investigations provide strong evidence that the statistical properties of the lasso and group lasso are less sensitive to covariance misspecification than unpenalized methods, a remarkable property of interest to practitioners.

The paper is structured as follows. Section 2.2 provides the necessary background information for the statistical properties of the lasso and the group lasso that motivate our diagnostic measures. Section 2.3 describes our proposed diagnostics and provides recommendations on how they should be used in practice. Section 2.4 demonstrates and justifies our approaches with a thorough simulation study, and Section 2.5 applies our approaches to two data sets: a lasso analysis of the diabetes data set (see online materials of Hastie et al. [2015]) and group lasso analysis for functional data depicting finger movement for the group lasso [Stallrich et al., 2020]. Section 2.6 provides a summary of our approach and a discussion of related topics and future work.

2.2 Background

A penalized least-squares estimator for $\boldsymbol{\beta}^*$ is

$$\hat{\boldsymbol{\beta}}^P = \underset{\beta_0,\boldsymbol{\beta}}{\arg\min} \frac{1}{2N}||\boldsymbol{y} - \mathbf{1}\beta_0 - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + P(\boldsymbol{\beta}) \tag{2.1}$$

where $P(\cdot)$ is a penalty function that involves one or more positive tuning parameters. Since β_0 is not penalized, both $\boldsymbol{y}$ and the columns of $\boldsymbol{X}$ are centered to remove $\mathbf{1}\beta_0$ from (2.1). The lasso penalty, $P(\boldsymbol{\beta}) = \lambda||\boldsymbol{\beta}||_1$, can perform simultaneous estimation and inference by shrinking some estimates to 0 [Tibshirani, 1996]. When the columns of $\boldsymbol{X}$ can be partitioned into m groups, $\boldsymbol{X}\boldsymbol{\beta} = \sum_{j=1}^m \boldsymbol{X}_j\boldsymbol{\beta}_j$, and we wish to impose group sparsity, the group lasso penalty is often recommended: $P(\boldsymbol{\beta}) = \sum_{j=1}^m \lambda_j||\boldsymbol{\beta}_j||_2$ with $\lambda_j = (\lambda\sqrt{k_j})$ where k_j denotes the number of columns in $\boldsymbol{X}_j$. The group lasso then performs estimation/inference on the groups of coefficients. This grouping structure is common for generalized additive models and functional linear models [Fan et al., 2015; Gertheiss et al., 2013; Stallrich et al., 2020] in which a functional predictor's contribution to the mean model is approximated by a basis expansion.

In addition to centering $\boldsymbol{y}$ and $\boldsymbol{X}$, the columns of $\boldsymbol{X}$ are often scaled so $\frac{1}{N}\sum_{i=1}^N x_{ij}^2 = 1$, where x_{ij} is the (i,j)th element of $\boldsymbol{X}$. Scaling is necessary for the lasso and group lasso penalties so that each β_j can contribute equally to the loss function and penalty. Note that scaling occurs implicitly with least-squares-based test statistics and confidence intervals via the standard errors. When the inference procedure assumes $\boldsymbol{\Sigma} = \sigma^2\boldsymbol{I}$, $\text{Var}(\hat{\boldsymbol{\beta}}^{OLS}) = \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ with diagonal elements $\sigma^2/(1-R_j^2)$ where $R_j^2 = \boldsymbol{x}_j^T\boldsymbol{P}_{-j}\boldsymbol{x}_j$ is the percent variation of $\boldsymbol{x}_j$ explained by a linear model of the remaining predictors. The so-called variance inflation factors, $VIF_j = (1 - R_j^2)^{-1}$, should be investigated to diagnose serious issues with the resulting inference. A general rule of thumb is $VIF_j > 5$ or 10 imply low-powered inference due to multicollinearity [James et al., 2013]. For high-dimensional problems, the R_j^2 will be close or equal to 1 and so are no longer useful diagnostic measures. Instead, other model matrix diagnostics should be explored based on the lasso and group lasso estimators, which are more appropriate for high dimensions.

The success of a high-dimensional analysis depends on the degree of sparsity of $\boldsymbol{\beta}^*$ relative to N and the correlation between the columns of $\boldsymbol{X}$. Denote the set of indices corresponding to the nonzero entries of $\boldsymbol{\beta}^*$ by $S = \text{supp}(\boldsymbol{\beta}^*) = \{j : \beta_j^* \neq 0\}$ with complement S^c, and denote the corresponding columns of $\boldsymbol{X}$ as $\boldsymbol{X}_S$ and $\boldsymbol{X}_{S^c}$, respectively. The lasso's irrepresentable/incoherence condition (see Chapter 11 in Hastie et al. [2015] and the references

therein) concerns support recovery and is met if there is some constant $\delta > 0$ where

$$\max_{j \in S^c} ||(\boldsymbol{X}_S^T \boldsymbol{X}_S)^{-1} \boldsymbol{X}_S^T \boldsymbol{x}_j||_1 \leq 1 - \delta \ . \tag{2.2}$$

The ideal scenario has $1 - \delta = 0$, corresponding to orthogonality between $\boldsymbol{X}_S$ and $\boldsymbol{X}_{S^c}$. A similar condition for the group lasso is discussed in Section 2.3.1. This condition is closely related to VIF, except we only regress $\boldsymbol{x}_j$ with $j \in S^c$ onto the predictors in S. Condition 2.2 is necessary but not sufficient for support recovery, focusing only on the lasso's Karush-Kuhn-Tucker (KKT) conditions that involve setting the truly inactive β_j to 0. Hence, violating these conditions should lead to higher probabilities of false positives but tells us little about the true positive rate.

The incoherence condition relies on knowledge of the support, S, and makes no assumptions about the values of $\boldsymbol{\beta}_S$ or the error distribution. Hastie et al. [2015], based on the work by Bickel et al. [2009], provide a deterministic upper bound on the lasso's $\mathcal{L}_2$-error under a restricted eigenvalue (RE) condition and a given error vector, $\boldsymbol{e}$. Let $\widehat{\boldsymbol{v}} = \widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*$ be the estimated error vector, and let $\widehat{\boldsymbol{v}}_S \in \mathbb{R}^{|S|}$ and $\widehat{\boldsymbol{v}}_{S^c}$ be the sub-vectors of $\widehat{\boldsymbol{v}}$ corresponding to S and its complement, S^c, respectively. Then the RE condition holds for S and some $\gamma > 0$ when

$$\frac{\frac{1}{N}\boldsymbol{v}^T \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{v}}{||\boldsymbol{v}||_2^2} \geq \gamma \quad \text{for all nonzero } \boldsymbol{v} \in \mathcal{C}, \tag{2.3}$$

where $\mathcal{C}(S; \alpha) := \{\boldsymbol{v} \in \mathbb{R}^p : ||\boldsymbol{v}_{S^c}||_1 \leq \alpha ||\boldsymbol{v}_S||_1\}$ for some $\alpha \geq 1$. The interpretation is that $\mathcal{C}$ includes error vectors that are mostly supported on S. It is known for the lasso that $\alpha = 3$ when $\lambda \geq 2||\boldsymbol{X}^T \boldsymbol{e}||_\infty / N$. For such λ and known $\boldsymbol{e}$, the lasso estimator satisfies

$$||\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*||_2 \leq \frac{6}{\gamma} \frac{\sqrt{k}}{N} ||\boldsymbol{X}^T \boldsymbol{e}||_\infty \leq \frac{3\sqrt{k}}{\gamma} \lambda, \tag{2.4}$$

for $k = |S|$ [Hastie et al., 2015]. For a fixed $\boldsymbol{X}$, Hastie et al. [2015] assume $iid\ N(0, \sigma^2)$ errors to derive the following union probability bound

$$\mathbb{P}\left[||\boldsymbol{X}^T \boldsymbol{e}||_\infty / N \geq t\right] \leq 2e^{\left\{-\frac{Nt^2}{2\sigma^2} + \log p\right\}}. \tag{2.5}$$

Denoting the right side of (2.5) by P, it follows that with $\lambda = 2t$, $||\widehat{\beta} - \beta^*||_2 \leq ct\sqrt{k}/\gamma$ for some constant $c > 0$ with probability at least $1 - P$. Clearly, understanding the behavior of $||\boldsymbol{X}^T \boldsymbol{e}||_\infty / N$ is vital to understanding robustness. This idea serves as the basis for our covariance robustness diagnostics in Section 2.3.2.

Lounici et al. [2009] also follow the technique by Bickel et al. [2009] to study the group lasso, providing theoretical bounds on a large class of loss functions comparing $\hat{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}^*$. Similar to (2.4), they provide a bound on the group-wise $\mathcal{L}_2$ loss function, $||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*||_{2,1} = \sum_{j=1}^{m} ||\hat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_j^*||_2 \geq ||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*||_2$, but under *iid*, Normally distributed errors. We prefer to work with a more general bound in order to study robustness to different covariance structures and in Section 2.3.2 we derive a deterministic upper bound on $||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*||_{2,1}$ for a known support S. Huang & Zhang [2010] compare the group lasso's performance to the lasso and derive $\mathcal{L}_2$ bounds on the group lasso with relaxed assumptions about the error vector. Under a group sparsity assumption, their theory relies on a group noise condition that assumes there exist constants $a, b \geq 0$ such that with probability $1 - \eta$ for $\eta \in (0, 1)$,

$$||(\boldsymbol{X}_j^T \boldsymbol{X}_j)^{-1/2} \boldsymbol{X}_j^T \boldsymbol{e}||_2 \leq a\sqrt{k_j} + b\sqrt{-\ln \eta}, \tag{2.6}$$

for all $j = 1, ..., m$. They show that for $e_i \sim N(0, \sigma_i^2)$ where $\sigma_i \leq \sigma$, $i = 1, ..., N$, the group noise condition holds with $a = \sigma$ and $b = \sqrt{2}\sigma$. The appropriate constants for arbitrary error distributions are more difficult to derive but if the condition holds for similar-valued constants, the estimation should be robust.

Diagnostic measures based on the above results are more heuristic than VIF since they are not as directly tied to an inferential procedure. Their main advantages lie in their interpretation and speed of calculation. Chapter 6 of [Hastie et al., 2015] reviews inference procedures under the lasso and group lasso, such as Bayesian lasso, bootstrapping, and debiased lasso. The methods are unfortunately either challenging conceptually or computationally slow. Cross-validation [Kohavi, 1995], or CV, is a common practical tool to determine the final tuning parameter value(s) and the final model is fitted using this tuning parameter on the full data. V-fold CV randomly splits the full data into V equally-sized sets, or folds. Each fold is used as a test set with training set based on combining the remaining folds, giving V estimates of squared prediction error for each value of λ considered. We then calculate the average predictor error APE and its standard error for each value of λ. The $minAPE$ rule chooses the λ value with the smallest minimum APE, while the SE rule chooses the largest λ value whose APE lies within one standard error of the minimum APE. To our knowledge, the CV literature has ignored the possibility that for a given λ, there could be significant variation in the estimated coefficients across the V training sets. This would hopefully lead to large APE standard errors but this is not guaranteed especially when the columns of $\boldsymbol{X}$ are highly correlated. Practitioners need a diagnostic that can further highlight the potential for these inconsistencies.

An obvious limitation to motivating diagnostics from the the above results is that they

often require knowledge of the support S. It is computationally infeasible to check whether the conditions hold across all possible supports so we recommend an exploratory approach that randomly samples supports of a given size. We also recommend that the diagnostics be calculated under estimated supports based on some model selection methods, such as the $minAPE$ or SE rule.

2.3 Model Matrix and Cross-Validation Diagnostics

We now describe our proposed diagnostics based on the incoherence conditions and $\mathcal{L}_2$ bounds for the lasso and group lasso. The incoherence conditions produce diagnostics for confidence in support recovery and do not rely on error vector assumptions, while diagnostics based on the $\mathcal{L}_2$ bounds are recommended for studying robustness to different error assumptions. We also introduce diagnostics to be computed during cross-validation to detect large discrepancies in the folds' estimates.

2.3.1 Support Recovery and Correlation Diagnostics

For a given support S, let $IC_S = 1 - \delta > 0$ denote the right-hand side of equation (2.2) corresponding to the lasso. When IC_S approaches or exceeds 1, we should be skeptical of the lasso's estimated support since the predictors in the support are highly correlated with one or more of the predictors outside the support. A similar approach that may be more attractive to practitioners familiar with VIF is the following high-dimensional version. For a support S, define $VIF_S = \max_j VIF_{S,j}$ where $VIF_{S,j} = (1 - R^2_{S,j})^{-1}$ and

$$R^2_{S,j} = \boldsymbol{x}_j^T \boldsymbol{P}_{S/j} \boldsymbol{x}_j \ . \tag{2.7}$$

The value $R^2_{S,j}$ measures the explained percent variation of $\boldsymbol{x}_j$ with respect to S with index j removed, denoted by S/j (if $j \notin S$ then $S/j = S$). We investigate rules of thumb for these values in the following section.

Based on the lasso's primal-dual witness construction technique [Wainwright, 2009] that led to equation (2.2), we derived the following condition for the group lasso

$$\sqrt{\Lambda_{\max}(\boldsymbol{X}_j^T \boldsymbol{X}_S (\boldsymbol{X}_S^T \boldsymbol{X}_S)^{-2} \boldsymbol{X}_S^T \boldsymbol{X}_j)} < \frac{\sqrt{k_j}}{\sqrt{\sum_{j' \in S} k_{j'}}} \ , \tag{2.8}$$

for all $j \in S^c$, where $\Lambda_{\max}(\boldsymbol{A})$ denotes the maximum eigenvalue of square matrix $\boldsymbol{A}$. Details may be found in Appendix A.3. If $k_j = k$ for all j, then the right hand side becomes $1/\sqrt{|S|}$.

The lasso's $\mathcal{L}_2$ bounds in (2.4) imply another model matrix diagnostic based on the restricted eigenvalue, γ, but calculating it is a difficult optimization problem with a nonlinear constraint set. Others have shown that certain randomly generated $\boldsymbol{X}$ have a high probability of attaining the RE property [Raskutti et al., 2010], but this is difficult to verify for a given $\boldsymbol{X}$. An upper bound for γ can be found by simply setting $\boldsymbol{v}_{S^c} = \boldsymbol{0}$ and calculating the minimum eigenvalue of $\boldsymbol{X}_S^T\boldsymbol{X}_S/N$. We believe such a metric has limited utility and prefer to use the $\mathcal{L}_2$ bounds as a way to investigate robustness to different error assumptions, described next.

2.3.2 Covariance Robustness Diagnostics

Based on (2.4), the lasso's robustness to the error assumptions can be heuristically investigated by studying the robustness of the distribution of $\frac{1}{N}||\boldsymbol{X}^T\boldsymbol{e}||_\infty$. The idea is that if these distributions are consistent across different error distributions, so will be the resulting $\mathcal{L}_2$ losses. Comparing these distributions is analytically challenging, so we propose a simulation-based approach to approximate these distributions by simulating error vectors. However, the error vectors we generate should exhibit some degree of similarity that is consistent with the collected data. For example, suppose we want to compare $\boldsymbol{\Sigma} = \sigma^2\boldsymbol{I}$ with $\boldsymbol{\Sigma} = \sigma^2\{(1-\rho)\boldsymbol{I} + \rho\boldsymbol{J}\}$ for $|\rho| \leq 1$. Both corresponding models would have $\text{Var}(y_i) = \sigma^2$ but the expected total corrected sum of squares $\mathbb{E}(\boldsymbol{y}^T\boldsymbol{y}) = \text{tr}((\boldsymbol{I}-\boldsymbol{P}_1)\boldsymbol{\Sigma}) + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}$ (here $(\boldsymbol{I}-\boldsymbol{P}_1)$ denotes the column centering matrix) would be different between the two cases, since $\text{tr}((\boldsymbol{I}-\boldsymbol{P}_1)\boldsymbol{\Sigma})$ equals $(N-1)\sigma^2$ and $(N-1)(1-\rho)\sigma^2$, respectively. Therefore, for the same value of σ^2, we should expect the $\mathcal{L}_2$ loss for the CS case to be smaller than the *iid* case because the resulting $\boldsymbol{y}$ would have smaller total variance. This is also seen by modifying the union probability bound in (2.5) for the CS case

$$\mathbb{P}\left[||\boldsymbol{X}^T\boldsymbol{e}||_\infty/N \geq t\right] \leq 2e^{\left\{-\frac{Nt^2}{2\sigma^2(1-\rho)}+\log p\right\}}, \tag{2.9}$$

which is a smaller probability than (2.5).

Since we are interested in studying robustness for the data at hand, in practice one should generate some estimate, $\hat{\sigma}^2$, under the *iid* assumption and then standardize other competing error distributions as follows. For an error distribution's $\boldsymbol{\Sigma}$ different from $\hat{\sigma}^2\boldsymbol{I}$, let $d(\boldsymbol{\Sigma}) = \text{tr}((\boldsymbol{I}-\boldsymbol{P}_1)\boldsymbol{\Sigma})$ and replace $\boldsymbol{\Sigma}$ with $\boldsymbol{\Sigma}^* = \frac{(N-1)\hat{\sigma}^2}{d(\boldsymbol{\Sigma})}\boldsymbol{\Sigma}$ so that $d(\boldsymbol{\Sigma}^*) = (N-1)\hat{\sigma}^2$, matching the *iid* scenario. Figure 2.2 shows these simulated distributions before and after standardization for the scenarios shown in Figure 2.1 as well as error distributions for a symmetric uniform and Laplace distribution. The standardization has made the distributions based on the Normal errors comparable which is consistent with the observation of robustness

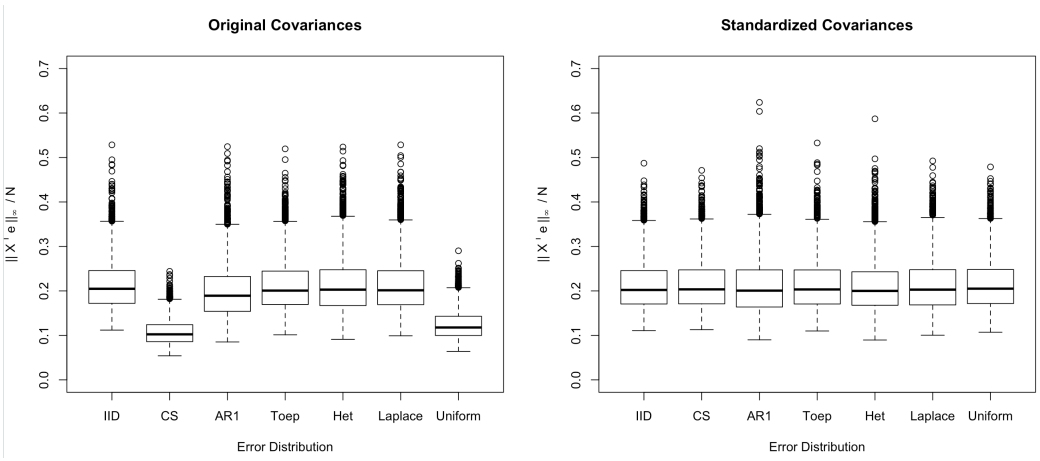

Figure 2.2 Boxplots for randomly generated $||\boldsymbol{X}^T\boldsymbol{e}||_\infty/N$ across different errors distributions for $N = 100$ and $p = 1000$ and IID generated $\boldsymbol{X}$. The left plot shows these distributions without standardizing the covariances and the right shows the results after standardization.

of the lasso in Figure 2.1.

With the appropriate modifications performed, an empirical distribution of $\frac{1}{N}||\boldsymbol{X}^T\boldsymbol{e}||_\infty$ for a collection of error distributions can be easily simulated. The univariate distributions may be easily compared visually or through summary statistics such as probability bounds such as (2.4), medians, or means. We demonstrate and further justify this approach in Sections 2.4 and 2.5.

A similar approach may be taken for the group lasso based on the following deterministic $\mathcal{L}_2$ bound. Assuming $\boldsymbol{X}$ has a positive group lasso restricted eigenvalue:

$$\min\left\{\frac{||\boldsymbol{X}\boldsymbol{\nu}||_2}{\sqrt{N}||\boldsymbol{\nu}_S||_2} : \sum_{j\in S^c}\lambda_j||\boldsymbol{\nu}_j||_2 \leq 3\sum_{j\in S}\lambda_j||\boldsymbol{\nu}_j||_2\right\} \geq \kappa \ , \tag{2.10}$$

and $\lambda_j \geq 2||\boldsymbol{X}_j^T\boldsymbol{e}||_2/N$ for all j, we show in Appendix A.4 that

$$\sum_j ||\hat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_j^*||_2 \leq \frac{32}{N\kappa^2}\sum_{j\in S}\frac{||\boldsymbol{X}_j^T\boldsymbol{e}||_2^2}{\min_j||\boldsymbol{X}_j^T\boldsymbol{e}||_2} \leq \frac{16}{\kappa^2}\sum_{j\in S}\frac{\lambda_j^2}{\lambda_{\min}} \ , \tag{2.11}$$

where $\lambda_{\min} = \min_{j=1,\ldots,m}\lambda_j$. This shares a strong similarity to (2.4) except it involves $||\boldsymbol{X}_j^T\boldsymbol{e}||_2$. While we could look at the m different distributions for each group, we prefer to look at the distribution of $\max_j||\boldsymbol{X}_j^T\boldsymbol{e}||_2/\sqrt{N}$.

Note that the $\mathcal{L}_2$ error bounds in (2.4) and (2.3.2) rely on satisfying the respective restricted eigenvalue conditions with the model resulting from an 'optimal' regularization parameter $\lambda \geq 2||\boldsymbol{X}^T\boldsymbol{e}||_\infty/N$ for the lasso and $\lambda_j \geq 2||\boldsymbol{X}_j^T\boldsymbol{e}||_2/N$ for the group lasso. In

practice, where the true support and error distribution are typically unknown, proving that both of these items are satisfied is not possible. Our approach is a heuristic way of quickly studying the potential effects of different error distributions without needing to specify or suppose the sparsity of $\boldsymbol{\beta}$.

2.3.3 A Cross-Validation Diagnostic

For V-fold cross-validation and a tuning parameter λ, we have $V+1$ estimates for $\boldsymbol{\beta}^*$ that come from the V folds $(\hat{\boldsymbol{\beta}}^1, \ldots, \hat{\boldsymbol{\beta}}^V)$ and the estimate from the full data set, $\hat{\boldsymbol{\beta}}$. These estimates involve overlapping data sets and so should be correlated with each other. However, some data sets may exhibit inconsistent correlations which is problematic when we use cross-validation for tuning parameter selection. In particular, if we recommend a tuning parameter based on the APE of the estimates $\hat{\boldsymbol{\beta}}^1, \ldots, \hat{\boldsymbol{\beta}}^V$ but these estimates are not correlated with $\hat{\boldsymbol{\beta}}$, then the resulting conclusions may be poor. Therefore, we recommend that the squared correlations $\text{Corr}(\hat{\boldsymbol{\beta}}^v, \hat{\boldsymbol{\beta}})^2$ be calculated along with calculating V prediction errors. Investigating these correlations can reveal major inference issues attributable to either outliers or unusual correlation behavior of $\boldsymbol{X}$.

As a short demonstration, we generated a high-dimensional $\boldsymbol{X}$ matrix with $N = 50$ and $p = 1000$, a response vector with iid, $N(0,1)$ errors, and a $\boldsymbol{\beta}$ with 10 randomly generated nonzero effects of magnitude 1 or larger. Figure 2.3 plots the results of a 5-fold CV analysis including, for each λ value, the APE, APE plus one standard error, the average $\text{Corr}(\hat{\boldsymbol{\beta}}^v, \hat{\boldsymbol{\beta}})^2$, and the minimum $\text{Corr}(\hat{\boldsymbol{\beta}}^v, \hat{\boldsymbol{\beta}})^2$. Inspection of only the APE may lead one to believe the V CV models have relatively consistent fits, while the two $\text{Corr}(\hat{\boldsymbol{\beta}}^v, \hat{\boldsymbol{\beta}})^2$ summaries imply the opposite. Indeed, the supports differed significantly between the folds and often missed many of the important effects. This straightforward metric hence should indicate issues with the tuning parameter selection process and any of its resulting inferences.

2.4 Simulation Study

This section demonstrates and justifies the proposed diagnostics across multiple simulated scenarios. The results also lead to rules of thumb for practitioners.

2.4.1 Lasso Diagnostics

For sample sizes $N \in \{50, 100\}$, we consider low- and high-dimensional scenarios with the respective number of parameters $p \in \{N/2, 2N, 10N\}$. We simulated 50 $N \times p$ model matrices $\boldsymbol{X}$ by independently drawing each row from either a zero-mean Normal distribution with

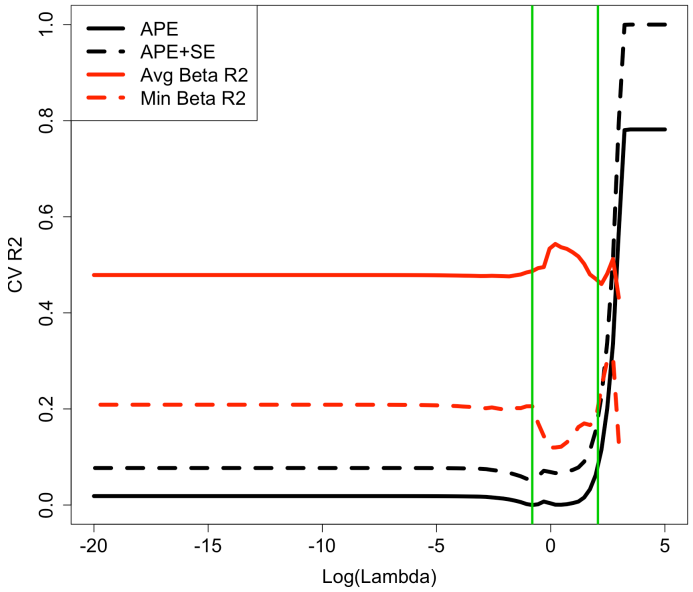

Figure 2.3 Results of 5-fold CV lasso analysis for $N = 50$, $p = 1000$, and $k = 10$ important effects. The left (right) vertical green line is the λ value for the APE (SE) rule. The red solid and dashed lines correspond to the average and minimum $\text{Corr}(\hat{\boldsymbol{\beta}}^v, \hat{\boldsymbol{\beta}})^2$ values.

correlation matrix $\boldsymbol{\Sigma_x}$ or draw ± 1 elements with equal probability. We let $\boldsymbol{\Sigma_x}$ be structured as (1) the identity matrix, (2) CS with $\rho_x = 0.8$, (3) AR(1) with $\rho_x = 0.8$, and (4) one-banded Toeplitz with correlation $\rho_{xT} = 0.5$. We fixed $\sigma^2 = 1$ and centered and scaled each generated $\boldsymbol{X}$.

For each (N, p) combination, we considered two support sizes $k \in \{N/10, N/3\}$. For each $\boldsymbol{X}$, we calculated IC_S and VIF_S across 5000 randomly sampled supports of size k and stored the minimum, 25th percentile, median, mean, 75th percentile, and maximum of these values. We also calculated covariance robustness diagnostics by simulating 5000 error vectors for each $\boldsymbol{X}$ from four error distributions: a standard Laplace distribution and three normal distributions with (1) independent and constant variance, (2) $AR(1)$ correlation with $\rho_e = 0.85$ and constant variance, and (3) independent and heterogeneous variances. The heterogeneous variances were generated from a gamma distribution with shape and rate parameters both equal to 0.5. All covariance matrices were standardized according to the reference iid distribution having $\boldsymbol{\Sigma} = \boldsymbol{I}$.

For each $\boldsymbol{X}$ and error distribution, we simulated 100 responses with $\boldsymbol{\beta}$ vectors having k nonzero effects drawn from a $U[1, 10]$ with random positive and negative signs assigned with probability 0.5. For each simulated data set we performed a 5-fold CV lasso analysis with λ sequence ranging from $e^(-20)$ to $e^(5)$ using the `R` package `cv.glmnet`. We recorded the following information for each analysis:

1. $\mathcal{L}_2$ loss of full data's lasso estimate across λ sequence;

2. Estimated supports under $minAPE$ rule (S_{APE}), SE rule (S_{SE}), and the support having the smallest false discovery rate among those supports with highest true positive rate (S_*);

3. IC_S and VIF_S for S_{APE}, S_{SE}, S_*;

4. True positive rate (TPR) and false discovery rate (FDR) for S_{APE}, S_{SE}, S_*;

5. Minimum $\text{Corr}(\hat{\boldsymbol{\beta}}^v, \hat{\boldsymbol{\beta}})^2$ for λ's corresponding to S_{APE}, S_{SE}, S_* .

For clarity, TPR is defined as the proportion of truly active effects that are included in the estimated support, while FDR is the proportion of nonzero lasso estimates that belong to truly inactive effects.

After running the 100 analyses, we stored the following summarized information for each $\boldsymbol{X}$

1. Mean $\mathcal{L}_2$ loss across λ sequence;

2. Minimum, 25th percentile, median, mean, 75th percentile, and maximum statistics for IC, VIF, TPR, FDR, and minimum $\text{Corr}(\hat{\boldsymbol{\beta}}^v, \hat{\boldsymbol{\beta}})^2$ across supports S_{APE}, S_{SE}, and S_*.

We compared each generated $\boldsymbol{X}$'s mean IC_S value across the 5000 randomly generated supports, denoted $\overline{IC}_k$, and the similarly defined max VIF_S, denoted, VIF_k^{∞}, to the mean FDR. We let $\overline{IC}_{APE}$ and VIF_{APE}^{∞} denote these diagnostics for S_{APE} and use similar notations for the other supports. These support recovery diagnostics were also compared to the mean FDR. We then compared the average TPR with each of the three estimated supports' average minimum $\text{Corr}(\hat{\boldsymbol{\beta}}^v, \hat{\boldsymbol{\beta}})^2$, denoted by $\bar{\rho}_{APE}, \bar{\rho}_{SE}$, and $\bar{\rho}_*$. Finally, we then checked whether consistency/inconsistency of the covariance robustness diagnostics led to similar behavior of the mean $\mathcal{L}_2$ loss across the range of λ.

Of the 12 combinations of (N, p, k), we only report here two scenarios: $(100, 50, 10)$ and $(50, 500, 17)$. Results from the other scenarios exhibited similar behavior and are available upon request. Before we discuss the results, we briefly comment on these two scenarios and overall behavior of the simulations. The scenario $(100, 50, 10)$ seems like it should be little to no challenge to lasso: it is low-dimensional and the signal is fairly sparse with only 20% of the predictors having active effects. The TPR for all three support sets was always 1, but this was often accompanied by a non-negligible FDR. Scenario $(50, 500, 17)$ is high-dimensional and while the signal is sparse relative to $p = 500$, it is not sparse relative to N. Both the

TPR and FDR suffer from this, and we show how issues with TPR are elucidated by small minimum CV correlations. Finally, our simulations indicate that the lasso's expected $\mathcal{L}_2$ loss for a given $\boldsymbol{X}$ is highly consistent for standardized covariances.

Figure 2.4 shows the relationship between $\overline{IC}_k$ and VIF_k^∞ statistics calculated across 5000 randomly generated supports. It is not surprising to see a relationship between the two diagnostics, but the relationship deteriorates as the metric values increase. Figure 2.5 compares the $\overline{IC}_k$ and VIF_k^∞ to the estimated best-case diagnostics $\overline{IC}_*$ and VIF_*^∞. As $\overline{IC}_k$ increases, $\overline{IC}_*$ tends to exceed the reference line that implies perfect agreement.

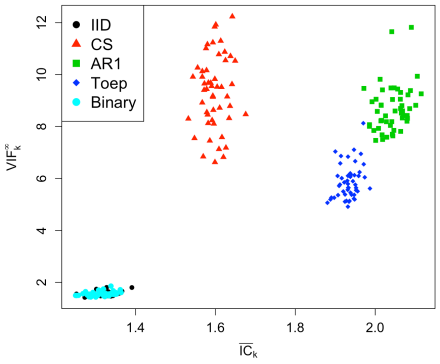

Figure 2.4 $(N, p, k) = (100, 50, 10)$: scatterplot of $\overline{IC}_k$ and VIF_k^∞ statistics calculated across 5000 randomly generated supports. Different types of points refer to the construction method for $\boldsymbol{X}$.

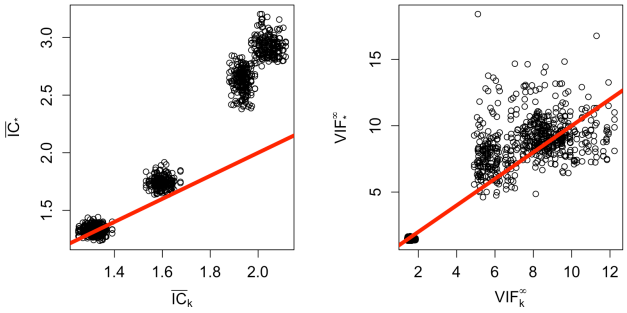

Figure 2.5 $(N, p, k) = (100, 50, 10)$: scatterplot of $\overline{IC}_k$ and VIF_k^∞ for true support size versus the diagnostics $\overline{IC}_*$ and VIF_*^∞ under estimated support S_*.

Comparisons of the estimated supports' diagnostics versus FDR are shown in Figure 2.6. The IC-based diagnostics are more strongly correlated with FDR than the VIF-based

diagnostics, which is not surprising given that the former are more closely tied to the lasso. Still, we see that VIF values of 5 or larger imply truly inactive effects have been included in the estimated supports. According to these results, the same may be said of IC values that exceed 1.5. Even though $TPR = 1$ for all scenarios, we include Figure 2.7 that plots the TPR versus the CV $\bar{\rho}$ statistics. The $\bar{\rho}$ values consistently exceed 0.99 across all supports.

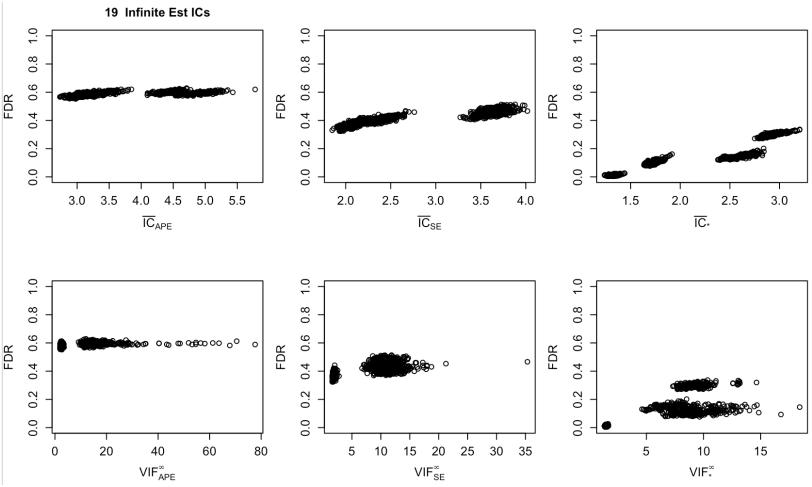

Figure 2.6 $(N, p, k) = (100, 50, 10)$: scatterplots of mean FDR's and support diagnostics for estimated supports S_{APE}, S_{SE}, and S_*.

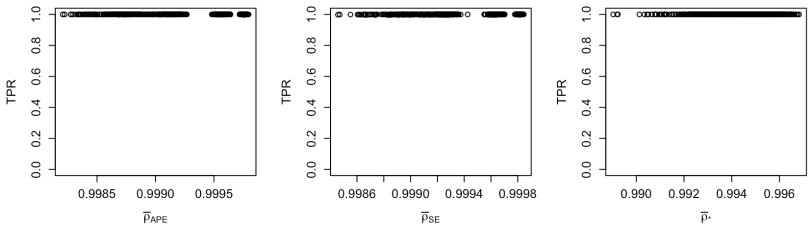

Figure 2.7 $(N, p, k) = (100, 50, 10)$: scatterplots of mean TPR versus $\bar{\rho}$ values for estimated supports S_{APE}, S_{SE}, and S_*. Note the narrow range of values for all three $\bar{\rho}$.

Next we investigated the error assumption robustness. Figure 2.8 shows histograms of the summary statistics of the 5000 randomly generated $||\boldsymbol{X}^T\boldsymbol{e}||_\infty/N$ across all generated $\boldsymbol{X}$. The histograms are remarkably consistent and the corresponding plots of the mean $\mathcal{L}_2$ loss across all generated $\boldsymbol{X}$ and λ also display strong consistency. As an example, Figure 2.9 shows these mean $\mathcal{L}_2$ loss functions for the CS-generated $\boldsymbol{X}$.

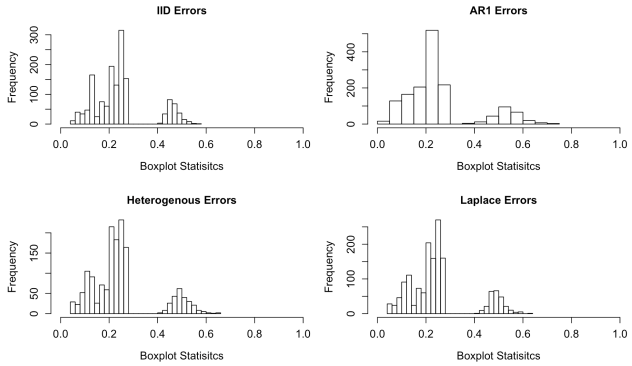

Figure 2.8 $(N, p, k) = (100, 50, 10)$: simultaneous histograms of all summary statistics for 5000 randomly generated $||\boldsymbol{X}^T\boldsymbol{e}||_\infty/N$ for the four considered error distributions.

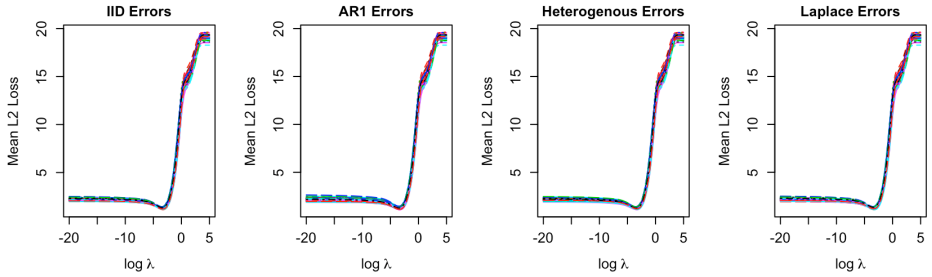

Figure 2.9 $(N, p, k) = (100, 50, 10)$: overlaying mean $\mathcal{L}_2$ loss curves for the 50 CS-constructed $\boldsymbol{X}$ matrices.

As anticipated, the $(50, 500, 17)$ case showed more concerning support recovery diagnostic measures. For brevity, we only recreate some of the figures from the previous scenario, but do report that the $\bar{IC}_*$ ranged from 12.66 to an astronomical 2670. Figures 2.10 and 2.11 investigate the relationship between the (log transformed) support recovery diagnostics to FDR and the $\bar{\rho}$'s to TPR, respectively. Again, the extremely large support recovery diagnostics are consistent with high FDR's. We also see smaller values of the $\bar{\rho}$ values that are accompanied by small TPR's. Based on the results of all 12 scenarios considered, we recommend that $\bar{\rho}$ statistics less than 0.8 should raise concerns about the lasso's TPR.

We were surprised to find that, even in this high-dimensional problem, the error robustness diagnostics were still fairly robust (see Figure 2.12). The most glaring difference between the histograms are the distribution of the maxima, with more extreme values for the $AR(1)$ error structure. Even so, Figure 2.13 gives empirical evidence that the mean $\mathcal{L}_2$ loss is still fairly robust.

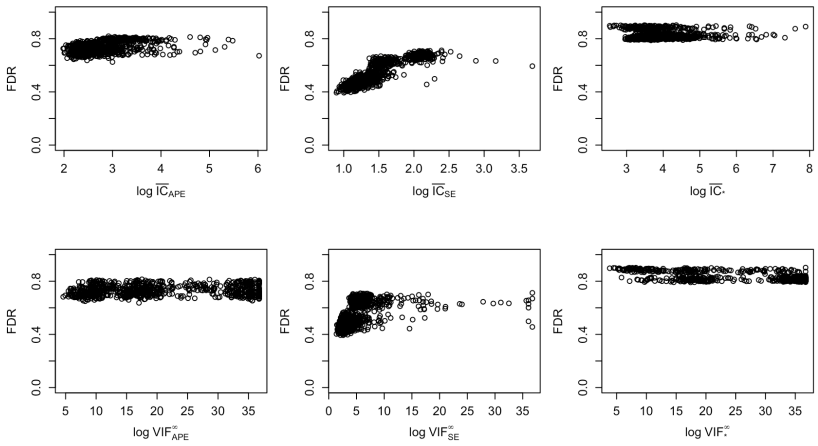

Figure 2.10 $(N, p, k) = (50, 500, 17)$: scatterplots of mean FDR's and log-transformed support diagnostics for estimated supports S_{APE}, S_{SE}, and S_*.

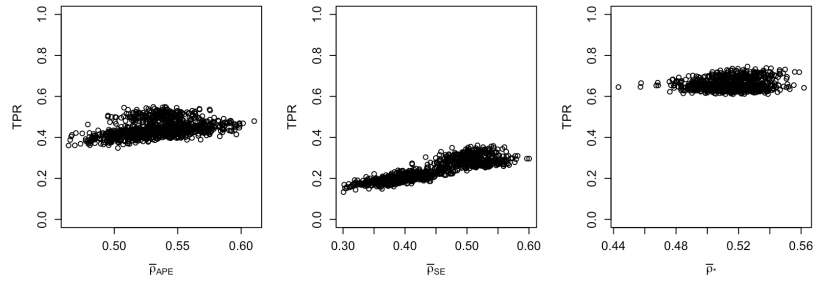

Figure 2.11 $(N, p, k) = (50, 500, 17)$: scatterplots of mean TPR versus $\bar{\rho}$ values for estimated supports S_{APE}, S_{SE}, and S_*.

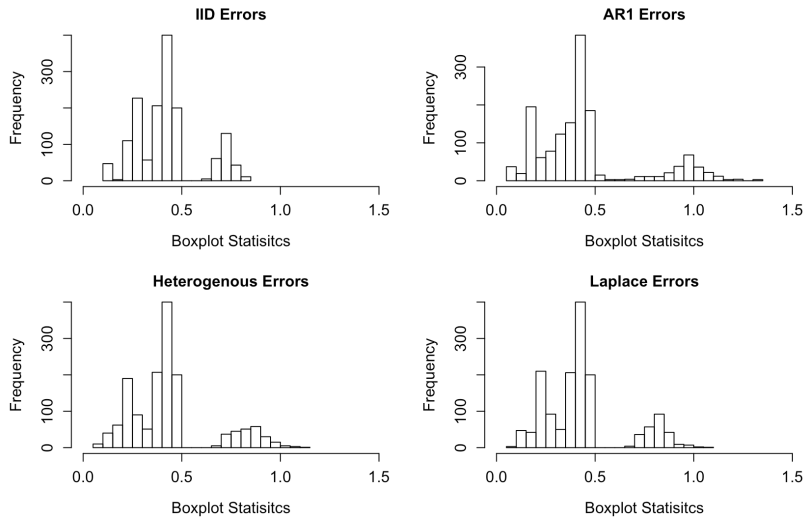

Figure 2.12 $(N, p, k) = (50, 500, 17)$: simultaneous histograms of all summary statistics for 5000 randomly generated $||\boldsymbol{X}^T\boldsymbol{e}||_\infty/N$ for the four considered distributions.

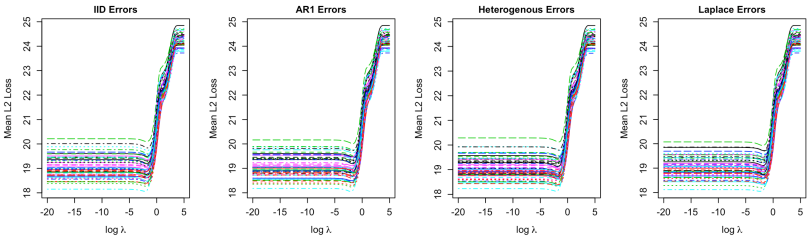

Figure 2.13 $(N, p, k) = (50, 500, 17)$: overlaying mean $\mathcal{L}_2$ loss curves for the 50 CS-constructed $\boldsymbol{X}$ matrices.

2.4.2 Group Lasso Diagnostics

To investigate the diagnostics of the group lasso, we follow Huang & Zhang [2010] and generate $\boldsymbol{X}$ with $m = 128$ groups of size $k_0 = 4$ for a total of $p = 512$ predictors. We fix $g = 16$ to be the number of active groups so there are $k = 64$ nonzero effects. The sample size N was chosen as multiples of k: $N = 64r$ for $r \in \{2, 4, 6\}$. All of these are high-dimensional cases, with $r = 2$ being the most challenging scenario. Each observation in $\boldsymbol{X}$ was a random draw from either the standard Normal distribution or the Uniform distribution on $(-1, 1)$. The simulation study was performed similarly to the lasso simulation study but with support recovery diagnostic $IC_S = \max_{j \in S^c} \sqrt{\Lambda_{\max}(\boldsymbol{X}_j^T\boldsymbol{X}_S(\boldsymbol{X}_S^T\boldsymbol{X}_S)^{-2}\boldsymbol{X}_S^T\boldsymbol{X}_j)}$ and the two error robustness diagnostics $\max_j ||\boldsymbol{X}_j^T\boldsymbol{e}||_2/\sqrt{N}$ and the proportion of times $||(\boldsymbol{X}_j^T\boldsymbol{X}_j)^{-1/2}\boldsymbol{X}_j^T\boldsymbol{e}||_2 \leq$

$a\sqrt{k_j} + b\sqrt{-\ln \eta}$ for all j. We chose $\eta = 0.05$, so the satisfaction proportions for the latter robustness diagnostic should be close to 0.95.

Table 2.1 summarizes the $\overline{IC}_g$ (being the average IC_S value across randomly generated support groups of size 16) for the two different $\boldsymbol{X}$ constructions. The statistics were consistent within and across the two constructions, with none of them reaching the desired value of $1/\sqrt{16} = 0.25$. This suggests most supports of size 16 will have a high FDR, but it does not tell us whether this will hold for the true, unknown support.

N	$\boldsymbol{X}$	Minimum	25th Perc	Median	Mean	75th Perc	Maximum
128	IID	1.536	1.546	1.548	1.548	1.551	1.559
	$U[-1,1]$	1.538	1.546	1.549	1.549	1.551	1.559
256	IID	0.8071	0.8112	0.8125	0.8126	0.8143	0.8169
	$U[-1,1]$	0.8091	0.8111	0.8128	0.8130	0.8142	0.8177
384	IID	0.6156	0.6177	0.6185	0.6186	0.6194	0.6215
	$U[-1,1]$	0.6151	0.6174	0.6186	0.6185	0.6198	0.6217

Table 2.1 Summary statistics of $\overline{IC}_g$ across group lasso scenarios. None of the scenarios reached the desired value of $1/\sqrt{|S|} = 0.25$.

For our simulation study, the estimated supports S_* and S_{APE} often selected most of the 128 groups, preventing calculation of the corresponding IC diagnostic. However, diagnostics under such estimated supports are uninteresting because the selection of such a large number of groups already causes us to question the resulting inferences. As anticipated, the FDR's for these estimated supports were over 0.80. Figure 2.14 shows the comparisons of $\overline{IC}_{SE}$ to FDR across the three N scenarios. The large FDR's for all scenarios is immediately explained by the large corresponding $\overline{IC}_{SE}$ values. For all scenarios, the 5 $\text{Corr}(\hat{\boldsymbol{\beta}}^v, \hat{\boldsymbol{\beta}})^2$ values were all above 0.99 indicating a higher TPR. This is unsurprising, though, given the size of the estimated supports.

Similar to the lasso, the two error robustness diagnostics imply the group lasso is robust to error distributions whose covariances are consistent with the data. Summaries of the empirical distributions of $\max_j ||\boldsymbol{X}_j^T \boldsymbol{e}||_2/\sqrt{N}$ resemble Figure 2.12 so we omit reporting the results to save space. The diagnostic based on the group noise condition tells a slightly different story. Figure 2.15 shows the proportion of times the group noise condition holds for the $N = 128$ scenario. While the heterogeneous variance and Laplace variance scenarios come close to the desired value of 0.95, the $AR(1)$ error distribution satisfies the condition much less often. Upon further investigation, we discovered that the group noise condition establishes the probability $1-\eta$ for a given group, whereas we calculated the joint probability

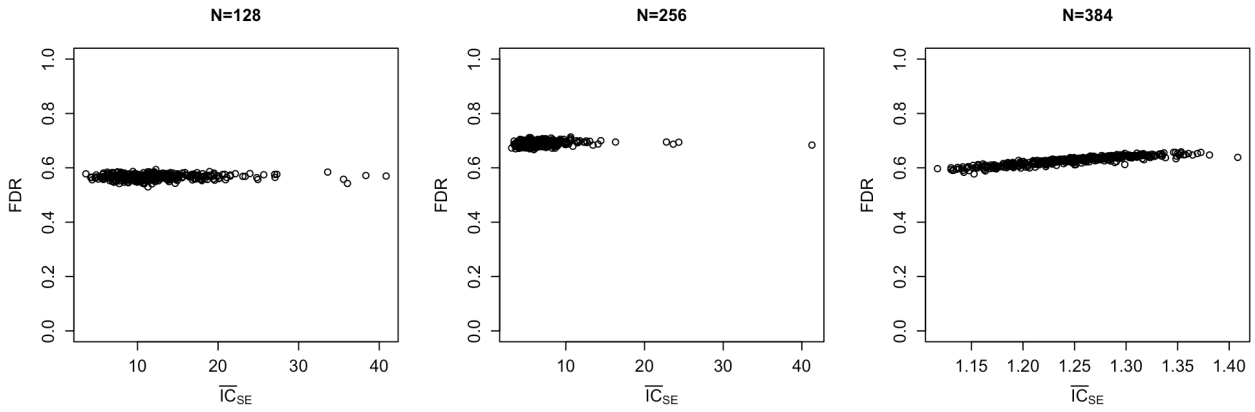

Figure 2.14 Scatterplots of mean FDR's and $\overline{IC}_{SE}$ group lasso support recovery diagnostics for $N = 128, 256$, and 384.

that all 128 groups satisfy the condition simultaneously because the primary result in Huang & Zhang [2010] relies on joint satisfaction of the condition. However, inspection of the mean $\mathcal{L}_2$ losses (similar to Figure 2.13, but omitted here to save space) imply a high degree of consistency. Given the confusion surrounding the group noise condition and that the group Incoherence Condition is more theoretically motivated, we conclude that diagnostics based on the empirical distributions of $\max_j ||\boldsymbol{X}_j^T \boldsymbol{e}||_2/\sqrt{N}$ are better indicators of error robustness.

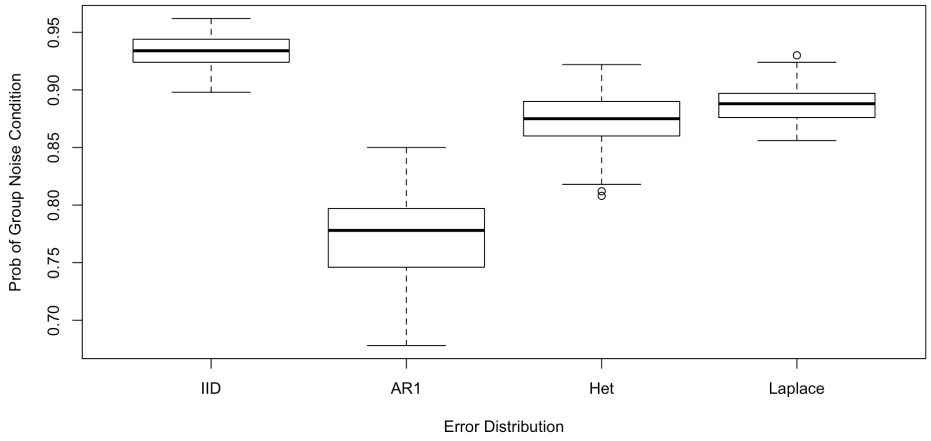

Figure 2.15 Boxplots for proportion of times the group noise condition holds across all 128 groups for the four considered error distributions. The desired level is 0.95.

2.5 Data Analysis

To demonstrate the practicality of our described simulation approach, we use this section to apply our approach to real data. We consider the popular diabetes data set [Efron et al., 2004] for the lasso in Section 2.5.1 and the EMG data described in the Introduction for the group lasso in Section 2.5.2.

2.5.1 Lasso - Diabetes Data

The diabetes data consist of 442 patients' measurements of age, sex, body-mass index, average blood pressure, and six blood serum measurements, as well as all two-factor interactions and quadratic effects for all variables except sex. Figure 2.16 shows the distribution of $\overline{IC}_S$ and VIF_S^∞ across 10000 randomly generated supports of size 10. The $\overline{IC}_S$ values are all above the desired level of 1 and some become quite large, while some $\log(VIF_S^\infty)$ values are below the recommended threshold of $\log(5)$. It seems then that there are some supports that would have desirable support recovery properties, and others that would be poor.

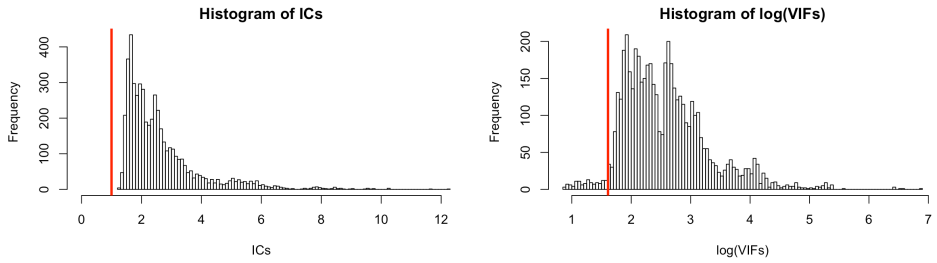

Figure 2.16 Histograms of $\overline{IC}_S$ and log-transformed VIF_S^∞ across $10,000$ randomly generated supports of size 10 for Diabetes data set. Values to the right of the red lines (1 for $\overline{IC}_S$ and $\log(5)$ for $\overline{VIF}_S^\infty$) indicate concerns about support recovery.

Figure 2.17 shows the individual $\overline{IC}_{S,j} = ||(\boldsymbol{X}_S^T\boldsymbol{X}_S)^{-1}\boldsymbol{X}_S^T\boldsymbol{x}_j||_1$ and $VIF_{S,j}^\infty$ values for estimated supports S_{APE} and S_{SE}. The two $\overline{IC}_{S,j}$ distributions are either concentrated around or are below the threshold of 1 and all the $VIF_{S,j}^\infty$ are below 5. These results are encouraging and suggest a low FDR. The 5 $\text{Corr}(\hat{\boldsymbol{\beta}}^v, \hat{\boldsymbol{\beta}})^2$ values are all above 0.96 indicating a higher TPR. These diagnostics instill confidence in the resulting inferences as it pertains to support recovery.

Finally, we investigated the error robustness for this data set across the error distributions considered in Figure 2.2. We calculated $\hat{\sigma}^2 = 2982.02$ based on an unpenalized linear model using support S_{SE}. We generated 1000 random vectors for each distribution and calculated

$||\boldsymbol{X}^T\boldsymbol{e}||_\infty/N$; the results are shown in Figure 2.18. These boxplots give us confidence that our inferences are robust to different error distributions that are consistent with the data.

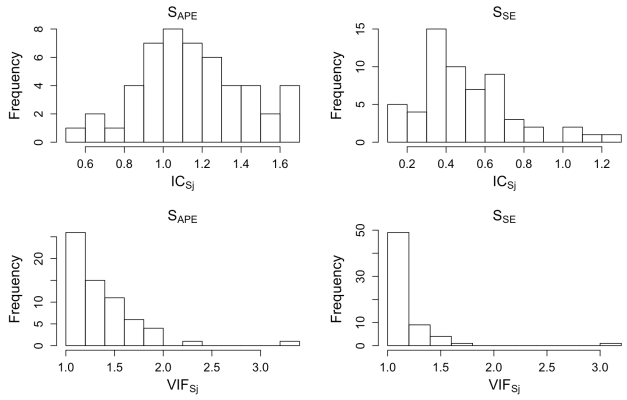

Figure 2.17 Histograms of all individual $\overline{IC}_{S,j}$ and $VIF^{\infty}_{S,j}$ values for estimated supports S_{APE} and S_{SE} for Diabetes data set.

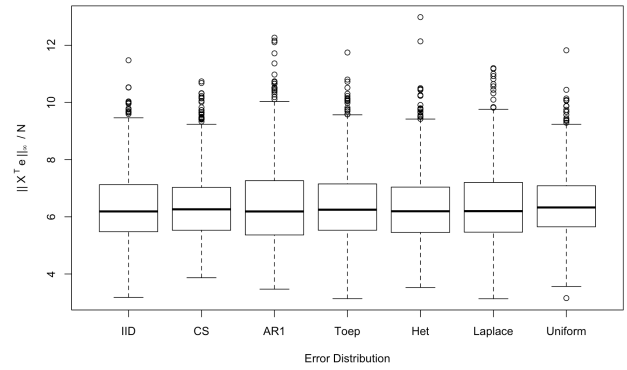

Figure 2.18 Boxplots for 5000 randomly generated $||\boldsymbol{X}^T\boldsymbol{e}||_\infty/N$ across different errors distributions for Diabetes data set.

2.5.2 Group Lasso - EMG Data

We now consider diagnostics for the analysis of the EMG data described in Chapter 1. First we give details on how this analysis becomes a group lasso problem. As discussed in Chapter 1, the EMG data in Stallrich et al. [2020] consists of simultaneous finger/wrist movement data and data from 15 EMG sensors. A fake signal was also included. The EMG signals and

the fake signal were restructured as functional covariates, $X_{ij}(t)$, over the preceding $\delta = 40$ time increments and used to predict hand velocity via the following functional linear model,

$$\mathbb{E}[y_i|X_{i1}, ..., X_{im}, z_i] = \sum_{j=1}^{m} \int_{-\delta}^{0} X_{ij}(t)\gamma_j(t, z_i)dt, \tag{2.12}$$

where y_i is the current velocity, $\gamma_j(t, z)$ is a bivariate coefficient function of time, $t \in [-\delta, 0]$, and position, $z \in \mathcal{Z}$. Each $\gamma_j(t, z)$ was approximated by a tensor product of orthogonal cubic B-spline basis functions, $\{\omega_\ell(\cdot)\}_{\ell=1}^{L}$ and $\{\tau_m(\cdot)\}_{m=1}^{M}$, given as $\gamma_j(t, z) \approx \boldsymbol{\omega}(t)\boldsymbol{B}_j\boldsymbol{\tau}(z)$. Using this notation, (2.12) can be approximated by the linear model

$$\mathbb{E}[y_i|\widetilde{\boldsymbol{X}}_{i1}, ..., \widetilde{\boldsymbol{X}}_{im}, z_i] = \sum_{j=1}^{m} \widetilde{\boldsymbol{X}}_{ij}\boldsymbol{\beta}_j, \tag{2.13}$$

where $\widetilde{\boldsymbol{X}}_{ij} = \left\{ \sum_{r=-\delta}^{0} X_{ij}(t_r)\boldsymbol{\omega}(t_r)^T \right\} \otimes \boldsymbol{\tau}(z_i)$ and $\boldsymbol{\beta}_j$ is the vectorized version of $\boldsymbol{B}_j$. The values $L = M = 10$ were used and, since the rows of each $N \times (\delta + 1)$ matrix $\widetilde{\boldsymbol{X}}_j$ were correlated, the data was thinned by keeping every 20th y_i.

To penalize both the magnitude and smoothness of the coefficient function, $\gamma_j(t, z)$, the general group lasso penalty $\sum_{j=1}^{m}(\boldsymbol{\beta}_j\boldsymbol{Q}_\varphi\boldsymbol{\beta}_j)^{1/2}$ is used with penalty matrix $\boldsymbol{Q}_\varphi = ||\gamma_j||_2 + \varphi_t||\gamma''_{j,t}||_2 + \varphi_z||\gamma''_{j,z}||_2$, where $\varphi_t, \varphi_z > 0$ are smoothness parameters and $\gamma''_{j,t}$ and $\gamma''_{j,z}$ are second partial derivatives of γ_j with respect to t and z, respectively. By taking the Cholesky decomposition of the penalty matrix, $\boldsymbol{Q}_\varphi = \boldsymbol{R}_\varphi\boldsymbol{R}_\varphi^T$, where $\boldsymbol{R}_\varphi$ is a lower triangular matrix, (2.13) is reparameterized and the least squares loss is penalized to obtain the group lasso objective function

$$\sum_{i=1}^{N}(y_i - \sum_{j=1}^{m} \boldsymbol{W}_{ij}^T\widetilde{\boldsymbol{\beta}}_j)^2 + \lambda \sum_{j=1}^{m} ||\widetilde{\boldsymbol{\beta}}_j||_2,$$

for $\boldsymbol{W}_{ij} = \boldsymbol{R}_\varphi^{-1}\widetilde{\boldsymbol{X}}_{ij}$, $\widetilde{\boldsymbol{\beta}}_j = \boldsymbol{R}_\varphi\boldsymbol{\beta}_j$, and $\lambda > 0$.

The sequential, adaptive functional estimation (SAFE) method developed in Stallrich et al. [2020] performs multiple stages of group lasso, updating the set of considered functional covariates and incorporating adaptive weights. Their first stage used all 16 signals and a conventional application of the group lasso. We applied our diagnostics to this initial stage for data set they denoted $FC1$, which Stallrich et al. [2020] noted had a high FDR (based on expert knowledge of the underlying biomechanical system). However, the sample size after thinning leads to a high-dimensional model and prevents calculation of the support recovery diagnostic. We propose instead the diagnostic $\sqrt{\Lambda_{\max}(\boldsymbol{X}_j^T\boldsymbol{X}_S(\boldsymbol{X}_S^T\boldsymbol{X}_S)^{+2}\boldsymbol{X}_S^T\boldsymbol{X}_j)}$,

which replaces the usual inverse with the Moore-Penrose inverse.

For brevity, we only report the support recovery diagnostics under the S_{SE} support, which included four functional predictors. The individual $IC_{S,j}$ were quite large, with minimum and maximum values of 1.78 and 22.96, respectively. Ideally, we would like these all to be less than 1. This calls into question the FDR for S_{SE}, which is exactly why Stallrich et al. [2020] employ SAFE. The 5 $\text{Corr}(\hat{\boldsymbol{\beta}}^v, \hat{\boldsymbol{\beta}})^2$ values were all above 0.86 indicating a reliable TPR.

We next investigated robustness with our two proposed metrics under $\hat{\sigma}^2 = 0.13$ coming from the APE for the chosen tuning parameter set. Figure 2.19 shows the distribution of the $\max_j ||\boldsymbol{X}_j^T\boldsymbol{e}||_2/\sqrt{N}$ across 1000 simulated error vectors. All error distributions appear robust except for the CS scenario. For the group noise condition diagnostic, we found all error distribution scenarios had probabilities of 0.95 or higher except for the CS scenario with a probability of 0.109. The reason for these failings with the CS distribution is we don't center the columns of $\boldsymbol{W}$ since Stallrich et al. [2020] fit a model with no intercept. This becomes clear when analyzing the variance of $\boldsymbol{X}^T\boldsymbol{e}$ when the error covariance matrix has the structure $\boldsymbol{\Sigma} = a\boldsymbol{I} + b\boldsymbol{J}$, where $\boldsymbol{J} = \boldsymbol{1}\boldsymbol{1}^T/N$. If the columns of $\boldsymbol{X}$ are not centered, then $\text{Var}(\boldsymbol{X}^T\boldsymbol{e}) = a\boldsymbol{X}^T\boldsymbol{X} + b\boldsymbol{X}^T\boldsymbol{J}\boldsymbol{X}$. If the columns are centered, say $\boldsymbol{X}_c = (\boldsymbol{I} - \boldsymbol{P}_1)\boldsymbol{X}$, then the second term in the variance expression equals zero and we have $\text{Var}(\boldsymbol{X}_c^T\boldsymbol{e}) = a\boldsymbol{X}_c^T\boldsymbol{X}_c$. Thus, the additional term $b\boldsymbol{X}^T\boldsymbol{J}\boldsymbol{X}$ in the variance expression is the cause of the increased spread in Figure 2.19 and the poor satisfaction of the group noise condition. Regardless, Stallrich et al. [2020] performed a residual analysis on the data and found little evidence of correlation.

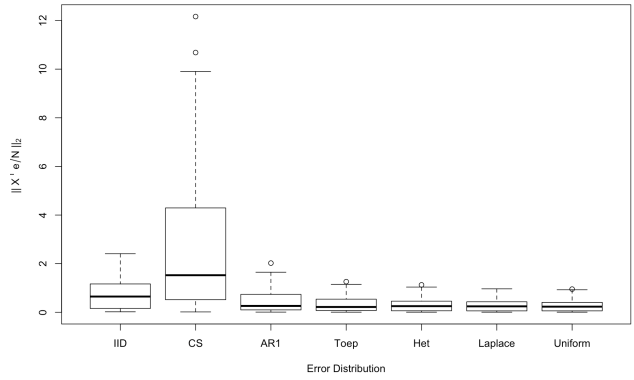

Figure 2.19 Boxplots for 1000 randomly generated $\max_j ||\boldsymbol{X}_j^T\boldsymbol{e}||_2/\sqrt{N}$ across different error distributions for the EMG data set.

2.6 Discussion

This paper takes an important first step in advocating and developing support recovery and error robustness diagnostic tools for the lasso and group lasso. Our proposed diagnostics are based on the fundamental results in the lasso and group lasso literature. They are interpretable, straightforward to calculate, and preferable over a more in-depth and computationally slow simulation study, which requires specification of the unknown model parameters. Our proposed cross-validation diagnostic is also straightforward to include in all existing cross-validation approaches with negligible added computational cost. Our recommended rules of thumb for the support recovery and cross-validation diagnostics were justified via an extensive simulation study. The main conclusions are that the IC-based support recovery diagnostics are strong indicators of FDR issues, while the cross-validation diagnostic should be used to asses issues with TPR. We have also provided strong evidence that the $\mathcal{L}_2$ loss behavior of the lasso and group lasso is robust to the error distribution, a remarkable property of interest to practitioners.

We acknowledge that our proposed diagnostics could potentially be improved. For example, Ravikumar et al. [2009] and [Lounici et al., 2009] provide other model matrix conditions that promote desirable support recovery properties. We are also interested in extending the diagnostics for the adaptive lasso [Zou, 2006], relaxed lasso [Meinshausen, 2007], and SAFE [Stallrich et al., 2020] procedures that have superior support recovery properties to the lasso and group lasso.

Chapter 3

On Functional Linear Regression with Smooth-Sparse Penalties

3.1 Introduction

When the response or explanatory variables are smooth, functional processes measured over time, as in the EMG application, a standard linear model is insufficient. In such settings, a functional linear model can be used to capture the smooth, continuous nature of the functional variables while linearly relating the response and predictors. The scenario of interest here is scalar-on-function linear regression, where the response is scalar, there are multiple functional predictors, and the corresponding effects vary over time. The model is written as

$$y_i = \alpha + \sum_{j=1}^{m} \int_{\mathcal{T}} X_{ij}(t)\gamma_j(t)dt + \varepsilon_i, \tag{3.1}$$

with functional predictors $X_j(t)$ measured on a dense grid of points $t \in \mathcal{T}$. Notice that (3.1) is a slightly simpler model than (1.1) proposed by Stallrich et al. [2020] to model the EMG signals in that γ_j does not depend on a scalar covariate in addition to time. Much of this chapter focuses on the simpler model for ease of presentation, but all of the methods described herein can be extended to accommodate bivariate coefficient functions.

An approach frequently used to fit model (3.1) approximates the functional effects with a linear combination of basis functions. Specifically, for a finite basis $\{\omega_\ell(t)\}_{\ell=1}^{L}$, let $\gamma_j(t) \approx \sum_{\ell=1}^{L} \beta_{j\ell}\omega_\ell(t)$ for $j = 1, ..., m$. By doing so, the problem simplifies to estimating a group of scalar coefficients for each functional effect rather than directly estimating the infinite-dimensional functions. As is standard in functional linear regression, $\gamma(t)$ is assumed to be a smooth function of time. To impose this assumption, one option is to strictly control the

number L of basis functions used in the approximation of $\gamma(t)$. This value must be large enough to capture the important features of $\gamma(t)$, but small enough so that $\gamma(t)$ retains interpretability. In practice, however, this balance is difficult to achieve. As a result, L often must take a low value, which forces $\gamma(t)$ to lie in a low-dimensional space and risks missing important features. Cardot et al. [2003] introduced a more flexible approach where L assumes a large value to provide a rich set of basis functions, and the roughness of $\gamma(t)$ is penalized by an appropriate derivative penalty. The degree of penalization is controlled by a tuning parameter $\varphi > 0$, such that larger values of φ encourage smoother, more linear effects and smaller values allow for more wiggly effects. The 'optimal' value of φ is commonly determined via cross-validation.

In high-dimensional settings where m is large, one may also assume that fewer than m functional predictors are active, that is, $\gamma_j(t) = 0$ for all $t \in \mathcal{T}$ for some $j \in \{1, ..., m\}$. Many functional variable selection methods have been proposed that utilize strategies such as hypothesis testing [Collazos et al., 2016; Su et al., 2017], random subspaces [Smaga & Matsui, 2018], and penalized regression [Fan et al., 2015; Gertheiss et al., 2013; Liu et al., 2014; Matsui & Konishi, 2011; Pannu & Billor, 2017; Stallrich et al., 2020; Zhu & Cox, 2009]. In alignment with the SAFE procedure [Stallrich et al., 2020] described in Chapter 1, we are interested in functional linear regression methods with a least squares loss function that penalize for smoothness and profile-wise sparsity. While Zhu & Cox [2009] and Fan et al. [2015] use a sparsity penalty, they impose smoothness by controlling the number of basis functions. Pannu & Billor [2017] use a least absolute deviation loss function with a group lasso smooth-sparse penalty, whereas Matsui & Konishi [2011] use the group SCAD (smoothly-clipped absolute deviation) penalty to induce sparsity. Gertheiss et al. [2013] developed the method that inspired SAFE, where one round of adaptive weighting with a joint smooth-sparse penalty is employed for the univariate functional model in (3.1). Liu et al. [2014] impose smoothness independently of profile-wise sparsity through the fused lasso [Tibshirani et al., 2005] and group lasso penalties [Yuan & Lin, 2006], respectively, but they also impose local sparsity on the effects such that γ_j can be zero on subregions of $\mathcal{T}$, which is not desired for EMG data analysis.

An issue with the joint smooth-sparse penalty is the conflation of tuning parameters. Regarding the penalty from Gertheiss et al. [2013] given as $P_{\lambda,\varphi}(\gamma_j) = \lambda(||\gamma_j||^2 + \varphi||\gamma_j''||^2)^{1/2}$, the ideal interpretations of the tuning parameters are that λ controls the level of model sparsity and φ controls the level of smoothness of γ_j. However, the placement of λ outside of the square root gives λ influence over both sparsity and smoothness. This is seen clearly

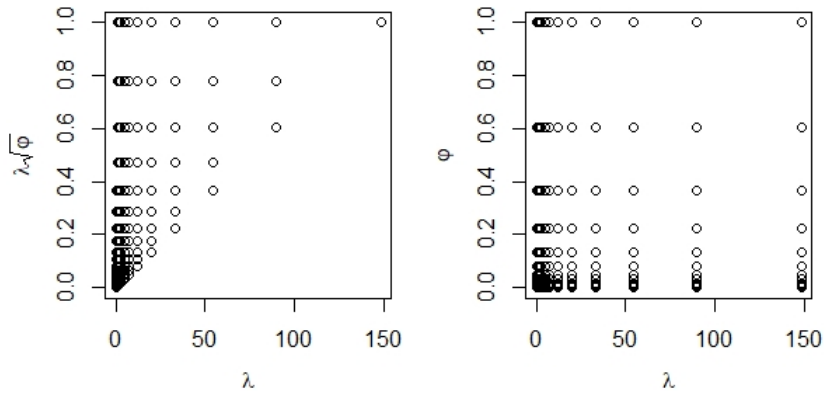

Figure 3.1 Illustration of tuning parameter search space under the joint smooth-sparse penalty (left) and a penalty with separate smoothness and sparsity penalties (right).

by distributing λ within the square root as

$$P_{\lambda,\varphi}(\gamma_j) = (\lambda^2||\gamma_j||^2 + \lambda^2\varphi||\gamma_j''||)^{1/2}. \tag{3.2}$$

For a given value of φ, small values of λ^2 will lead to a dense model with rough coefficient functions and large values of λ^2 will lead to a sparse model with smooth coefficients. Without careful specification of φ values, the model selection process may overlook whole classes of models, i.e., sparse models with rough coefficients and dense models with smooth coefficients. Figure 3.1 visualizes this issue, in which the left panel represents the models considered under the joint penalty and the right panel represents models considered under a penalty with independent tuning parameters. This phenomenon is amplified in the SAFE criterion (1.2) because λ influences smoothness in the directions of both t and z.

In theory, this problem can be solved by reparameterizing the smoothing parameters as $\varphi^* = \varphi/\lambda^2$. However, as will be described in the following section, functional linear regression with the smooth-sparse penalty of Gertheiss et al. [2013] is solved using the group lasso after certain linear approximations and a Cholesky decomposition of the penalty to reparameterize the model matrix and coefficient vector. Replacing φ with φ^* would require a Cholesky decomposition for each distinct φ^*, i.e. for each (φ, λ) set. In turn, this would require separate, independent executions of the group lasso algorithm for each distinct set of tuning parameters. Group lasso algorithms built on coordinate descent would hence the computational efficiency gained by using "warm starts" across a sequence of λ values [Friedman et al., 2007], where the solution for a given λ is used as the initial value for the next λ in the sequence. Thus, tuning parameter conflation is a practical problem.

Empirically, both Gertheiss et al. [2013] and Stallrich et al. [2020] appear to work around

this issue through adaptive weighting. The adaptive weights not only allow for different levels of smoothing for each coefficient function, but they may also resolve the mixing of tuning parameters after a few iterations. However, each estimation stage adds computational cost, so fewer stages would be better. Further, it was previously mentioned that the specification of possible φ values must be done with care. Unlike λ, whose possible range of values has an explicit maximum at which all coefficients are set to zero, there is little guidance for which values of φ should be considered. Having λ mix with φ only complicates this problem, and simply increasing the number of φ values considered greatly increases computation. For these reasons, a penalty that independently enforces smoothness and sparsity should be investigated.

In the context of high-dimensional additive models, Meier et al. [2009] proposed four smooth-sparse penalties, with three comprised of at least partially independent smoothness and sparsity components and the fourth being the foundation for the adaptive penalty in Gertheiss et al. [2013]. Written in the context of functional linear models with adaptive weights $w_j = ||\gamma_j||$ and $w_{j,1} = ||\gamma_j''||$, the penalties are as follows:

$$P_{\varphi,\lambda}(\gamma_j) = \varphi w_{j,1}||\gamma_j''|| + \lambda w_j||\gamma_j|| \tag{3.3}$$

$$P_{\varphi,\lambda}(\gamma_j) = \varphi w_{j,1}||\gamma_j''||^2 + \lambda w_j||\gamma_j|| \tag{3.4}$$

$$P_{\varphi,\lambda}(\gamma_j) = \varphi_1 w_{j,1}||\gamma_j''||^2 + \lambda\sqrt{w_j||\gamma_j||^2 + \varphi_2 w_{j,1}||\gamma_j''||} \tag{3.5}$$

$$P_{\varphi,\lambda}(\gamma_j) = \lambda\sqrt{w_j||\gamma_j||^2 + \varphi w_{j,1}||\gamma_j''||}. \tag{3.6}$$

As noted by Meier et al. [2009], (3.3) enjoys nice theoretical properties but is computationally expensive because it requires solving a second order cone programming problem. Thus, this penalty is of no interest for applications that require speedy computation. Of the penalty in (3.4), Meier et al. [2009] say it 'appears' to have theoretical drawbacks and the smoothing penalty needs to be within the square root, but no expansion is given on this point. They also develop theoretical results for the penalty in (3.5) and claim their simulations show $\varphi_1 = 0$ provides slightly better results than $\varphi_1 = \varphi_2$, but neglect to give supporting results in the paper. Given the lack of evidence supporting one penalty over another between (3.4)-(3.6), the functional data literature needs a practical implementation of the penalties to compare their performances.

The goal of this chapter is to conduct this comparison of penalties (3.4)-(3.6). As Meier et al. [2009] noted, it would be more ideal for the smoothing component of (3.4) to be under the square root as in (3.3) to keep smoothing and sparsity on the same scale. However, such a penalty results in a second order cone programming problem because there are two non-

differentiable penalties. Solving such a problem would require a new, more computationally intense, algorithm. A similar statement can be made regarding the smoothing-only component in (3.5). Hence, we opt to use the squared smoothing norm. Note that using the squared norm, though, places more emphasis on smoothing than sparsity. To counteract this so the three penalties can be compared more equally, and to provide greater computational stability, we use smaller smoothing parameters for the separate smoothing penalties. Specifically, we take the square root of φ and also of the smoothing weights $w_{j,1}$.

Of course, a reduction of stages is not the only way to reduce overall computation cost. Another strategy is to use a faster algorithm. As will be discussed in the next section, mathematical and computational issues arose when fitting a penalized regression with (3.4) and (3.5) under the group-wise majorization descent algorithm implemented in the `R` package `gglasso`. As a result, we modify and use the more efficient GLODE algorithm to compare the three penalties.

The remainder of this chapter proceeds as follows. Section 3.2 details the linear approximation of a penalized functional regression model to establish the need for the group lasso and provides background on solving the group lasso optimization problem, with mathematical descriptions of two existing group lasso algorithms. These algorithms are modified in Section 3.3 to accommodate the separate smoothness penalty in (3.4) and (3.5) with discussion of their respective limitations. All three penalties are compared through simulation in Section 3.4 and application to the EMG data in Section 3.5. Section 3.6 concludes the chapter with a discussion of observations and insights, as well as avenues for future research.

3.2 Group Lasso Solution Algorithms

First, consider the penalized functional linear regression with joint smooth-sparse penalty,

$$\underset{\{\gamma_j(t);j=1,\ldots,m\}}{\arg\min} \sum_{i=1}^{N}\left(y_i - \sum_{j=1}^{m}\int_{\mathcal{T}} X_{ij}(t)\gamma_j(t)dt\right)^2 + \lambda\sum_{j=1}^{m}(||\gamma_j||^2 + \varphi||\gamma_j''||^2)^{1/2}, \tag{3.7}$$

where the $X_{ij}(t)$ are assumed to be measured on a dense grid of points $t \in \mathcal{T}$. Approximating the functional coefficients as $\gamma_j(t) \approx \sum_{\ell=1}^{L}\beta_{j\ell}\omega_\ell(t)$ and using a Riemann sum approximation for the integral yields, for $j = 1, ..., m$,

$$\int_{\mathcal{T}} X_{ij}(t)\gamma_j(t)dt = \sum_{\ell=1}^{L}\left\{\Delta_t\sum_{r=1}^{n} X_{ij}(t_r)\omega_\ell(t_r)\right\}\beta_{j\ell} = \widetilde{\boldsymbol{X}}_{ij}^T\boldsymbol{\beta}_j, \tag{3.8}$$

with $\boldsymbol{\beta}_j = (\beta_{j1}, ..., \beta_{jL})^T$, $\widetilde{\boldsymbol{X}}_{ij} = (\widetilde{X}_{ij1}, ..., \widetilde{X}_{ijq})^T$, $\widetilde{X}_{ij\ell} = \Delta_t \sum_r X_{ij}(t_r)\omega_\ell(t_r)$, and $\Delta_t = t_r - t_{r-1}$ denotes the distance between two consecutive time points on an equally-spaced grid. The coefficient norms $||\gamma||$ and $||\gamma''||$ are rewritten in terms of the basis functions by defining the $L \times L$ matrices $\boldsymbol{\Omega}$ and $\boldsymbol{\Omega}_t$ with elements $(\boldsymbol{\Omega})_{\ell,\ell'} = \int_{\mathcal{T}} \omega_\ell(t)\omega_{\ell'}(t)dt$ and $(\boldsymbol{\Omega}_t)_{\ell,\ell'} = \int_{\mathcal{T}} \omega''_\ell(t)\omega''_{\ell'}(t)dt$, respectively, for $\ell, \ell' = 1, ..., L$. If the chosen basis functions are also orthonormal, which we assume here, then $\boldsymbol{\Omega} = \boldsymbol{I}$. Thus, the penalized functional regression can be re-expressed as a general group lasso problem [Yuan & Lin, 2006],

$$\sum_{i=1}^{N} \left(y_i - \alpha - \sum_{j=1}^{m} \widetilde{\boldsymbol{X}}_{ij}^T \boldsymbol{\beta}_j \right)^2 + \lambda \sum_{j=1}^{m} \{\boldsymbol{\beta}_j^T (\boldsymbol{I} + \varphi \boldsymbol{\Omega}_t) \boldsymbol{\beta}_j\}^{1/2}, \tag{3.9}$$

where estimation of each functional coefficient γ_j requires estimating the group of scalar coefficients $\boldsymbol{\beta}_j$.

Further, let $\boldsymbol{Q}_\varphi = \boldsymbol{I} + \varphi\boldsymbol{\Omega}_t$ and let $\boldsymbol{R}_\varphi$ be the square lower triangular matrix such that $\boldsymbol{Q}_\varphi = \boldsymbol{R}_\varphi \boldsymbol{R}_\varphi^T$, i.e., $\boldsymbol{R}_\varphi$ is the Cholesky factor of $\boldsymbol{Q}_\varphi$. Define $\widetilde{\boldsymbol{\beta}}_j = \boldsymbol{R}_\varphi^T \boldsymbol{\beta}_j$ and $\boldsymbol{W}_{ij} = \boldsymbol{R}_\varphi^{-1} \widetilde{\boldsymbol{X}}_{ij}$. Then the above group lasso can be reparameterized as

$$\sum_{i=1}^{N} \left(y_i - \alpha - \sum_{j=1}^{m} \boldsymbol{W}_{ij}^T \widetilde{\boldsymbol{\beta}}_j \right)^2 + \lambda \sum_{j=1}^{m} ||\widetilde{\boldsymbol{\beta}}_j||. \tag{3.10}$$

This version of the optimization problem can be solved for a particular λ and φ. To obtain the estimated functional effects, undo the Cholesky reparameterization by $\widehat{\boldsymbol{\beta}}_j = (\boldsymbol{R}_\varphi^T)^{-1} \widehat{\widetilde{\boldsymbol{\beta}}}_j$ and let $\widehat{\gamma}_j(t) = \sum_{\ell=1}^{L} \widehat{\beta}_{j\ell}\omega_\ell(t)$. With the above approximations, the optimization task transitions from estimating the unknown functions γ_j to sparsely estimating the groups of scalar coefficients $\boldsymbol{\beta}_j$, $j = 1, , m$. As such, the optimization problem becomes a standard group lasso problems. For ease of exhibition, the notation for a multivariate linear model with grouped covariates from Chapter 2 will be adopted for all further group lasso discussion.

To optimize a differentiable multivariate expression, one starts by taking the partial derivative with respect to the kth variable and equates the derivative to zero. The group lasso is not everywhere differentiable, through, so the following Karush-Kuhn Tucker (KKT) conditions are obtained,

$$\boldsymbol{X}_k^T (\boldsymbol{y} - \sum_{j=1}^{m} \boldsymbol{X}_j \boldsymbol{\beta}_j)/N - \lambda \frac{\boldsymbol{\beta}_k}{||\boldsymbol{\beta}_k||_2} = 0 \quad \text{if } j \in S, \tag{3.11}$$

$$||\boldsymbol{X}_k^T (\boldsymbol{y} - \sum_{j \neq k} \boldsymbol{X}_j \boldsymbol{\beta}_j)/N|| \leq \lambda \quad \text{otherwise}, \tag{3.12}$$

where $S = \{j : \boldsymbol{\beta}_j \neq 0\}$. If the matrices are group-wise orthogonal such that $\boldsymbol{X}_j^T \boldsymbol{X}_j = \boldsymbol{I}$ for $j = 1, ..., m$, then a closed-form expression can be obtained as in Yuan & Lin [2006] as

$$\widehat{\boldsymbol{\beta}}_k = \left(1 - \frac{\lambda}{||\boldsymbol{X}_k^T \boldsymbol{y}/N||_2}\right)_+ ||\boldsymbol{X}_k^T \boldsymbol{y}/N||_2. \tag{3.13}$$

Yuan & Lin [2006] and Meier et al. [2009, `grplasso`] developed block-wise descent algorithms for group lasso penalized least squares and logistic regression, respectively.

Although this group-wise orthogonal condition can be forced via Gram-Schmidt orthogonalization, the condition is easily violated by perturbations in the data or by removing a chunk of data as for cross-validation. Thus, Yang & Zou [2014] developed a group-wise majorization descent (GMD) algorithm that does not rely on group-wise orthogonality, is more computationally efficient than `grplasso`, and is implemented in the `R` package `gglasso` [Yang & Zou, 2017]. Further, Yau & Hui [2017] extended the least angle regression method of Efron et al. [2004, LARS] to the group lasso and demonstrated even better efficiency over `gglasso`. For completion, we build up both algorithms in the following sections and discuss their application to our problem in Section 3.3.

3.2.1 Group-wise Majorization Descent

The GMD algorithm uses a majorize-minimize (MM) approach to solve the group lasso. This approach consists of defining a majorization function that is uniformly bounded below by the original optimization function, then minimizing the majorization function to obtain the next coordinate in the descent. If the original optimization criterion is a smooth function, one option for the majorization function is a quadratic majorizer. By definition, a quadratic function g majorizes a function f at $\boldsymbol{y}$ on $\mathbb{R}^n$ if $g(\boldsymbol{y}) = f(\boldsymbol{y})$ and $g(\boldsymbol{x}) \geq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^n$. By writing this in the form

$$g(\boldsymbol{x}) = f(\boldsymbol{y}) + (\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{b} + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{y}), \tag{3.14}$$

it is clearly seen that $g(\boldsymbol{y}) = f(\boldsymbol{y})$. If f is twice differentiable, then $\boldsymbol{b} = \mathcal{D}f(\boldsymbol{y})$ and $\boldsymbol{A} - \mathcal{D}^2 f(\boldsymbol{y})$ is a nonnegative definite (n.n.d.) matrix, denoted $\boldsymbol{A} \gtrsim \mathcal{D}^2 f(\boldsymbol{y})$, where the operator $\mathcal{D}$ is the linear differential operator.

Not every function has a quadratic majorizer. However, for a loss function Φ with empirical loss $L(\boldsymbol{\beta}|\boldsymbol{D}) = \frac{1}{N}\sum_{i=1}^N \Phi(y_i, \boldsymbol{\beta}^T \boldsymbol{x}_i)$ and observed data $\boldsymbol{D}\{\boldsymbol{y}, \boldsymbol{X}\}$, Yang & Zou [2014] give the definition of the quadratic majorization (QM) condition as follows:

Definition 1. The loss function Φ is said to satisfy the QM condition if and only if the

following two assumptions hold:

(i) $L(\boldsymbol{\beta}|\boldsymbol{D})$ is differentiable as a function of $\boldsymbol{\beta}$, i.e., $\nabla L(\boldsymbol{\beta}|\boldsymbol{D})$ exists everywhere.

(ii) There exists a $p \times p$ matrix $\boldsymbol{H}$, which may only depend on the data $\boldsymbol{D}$, such that for all $\boldsymbol{\beta}, \boldsymbol{\beta}'$,

$$L(\boldsymbol{\beta}|\boldsymbol{D}) \leq L(\boldsymbol{\beta}'|\boldsymbol{D}) + (\boldsymbol{\beta} - \boldsymbol{\beta}')^T \nabla L(\boldsymbol{\beta}'|\boldsymbol{D}) + \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}')^T \boldsymbol{H}(\boldsymbol{\beta} - \boldsymbol{\beta}'). \tag{3.15}$$

Yang & Zou [2014] show that the least squares loss function satisfies the QM condition with $\boldsymbol{H} = \boldsymbol{X}^T\boldsymbol{X}/N$. To formalize the GMD algorithm, then, denote the current solution of $\boldsymbol{\beta}$ by $\boldsymbol{\beta}^t$. To determine the update of $\boldsymbol{\beta}_k^{t+1}$, note that $\boldsymbol{\beta} - \boldsymbol{\beta}^t = (\underbrace{0, ..., 0}_{k-1}, \boldsymbol{\beta}_k - \boldsymbol{\beta}_k^t, \underbrace{0, ..., 0}_{K-k})$, let $U(\boldsymbol{\beta}') = \nabla L(\boldsymbol{\beta}'|\boldsymbol{D})$ and let U_k denote the kth subvector of $U(\boldsymbol{\beta}')$. Let $\gamma_k = (1 + \epsilon^*)\eta_k$, where $\epsilon^* = 10^{-6}$ and η_k is the largest eigenvalue of $\boldsymbol{H}_k$, the submatrix of $\boldsymbol{H}$ corresponding to group k. The quadratic majorizer can then be relaxed by replacing $\boldsymbol{H}$ with γ_k, so that

$$L(\boldsymbol{\beta}|\boldsymbol{D}) \leq L(\boldsymbol{\beta}^t|\boldsymbol{D}) + (\boldsymbol{\beta}_k - \boldsymbol{\beta}_k^t)^T U_k + \frac{1}{2}\gamma_k ||\boldsymbol{\beta}_k - \boldsymbol{\beta}_k^t||_2^2. \tag{3.16}$$

The group lasso penalty is added to the right side of (3.16) to obtain the following optimization problem,

$$\arg\min_{\boldsymbol{\beta}_k} L(\boldsymbol{\beta}^t|\boldsymbol{D}) + (\boldsymbol{\beta}_k - \boldsymbol{\beta}_k^t)^T U_k + \frac{1}{2}\gamma_k ||\boldsymbol{\beta}_k - \boldsymbol{\beta}_k^t||_2^2 + \lambda ||\boldsymbol{\beta}_k||_2, \tag{3.17}$$

that has the closed-form update expression,

$$\boldsymbol{\beta}_k^{t+1} = \frac{1}{\gamma_k}\left(U_k + \boldsymbol{\beta}_k^t\right)\left(1 - \frac{\lambda}{||U_k + \gamma_k \boldsymbol{\beta}_k^t||_2}\right)_+. \tag{3.18}$$

The derivation of the update expression can be found in Appendix B. The full GMD algorithm operates by cyclically updating the estimate of $\boldsymbol{\beta}_k$ for $k = 1, ..., K$ until convergence.

3.2.2 GLODE

An alternative method for solving the group lasso problem is an extension of the least angle regression [Efron et al., 2004, LARS]. This algorithm, titled GLODE [Yau & Hui, 2017], uses a system of ordinary differential equations to describe the changes in the group lasso solution paths as the regularization parameter changes. As such, the full group lasso solution path can be obtained across a dense grid of regularization parameter values, denoted $\{\widehat{\boldsymbol{\beta}}(\lambda)\}_{\lambda \in [0,\infty)}$.

Through extensive simulations, Yau & Hui [2017] demonstrated a reduction in computation time as compared to `gglasso` while obtaining comparable prediction error. The GLODE algorithm proceeds as follows.

For a given sparsity parameter $\lambda > 0$ and weight vector $\boldsymbol{w} = (w_1, ..., w_m)^T$, where $w_j \geq 0$ and $w_j = 0$ denotes that $\boldsymbol{\beta}_j$ is not penalized, define the following functions,

$$q(\boldsymbol{\beta}) = \frac{1}{2}||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||^2, \quad \bar{h}(\boldsymbol{\beta}) = \sum_{j=1}^{m} w_j||\boldsymbol{\beta}_j||, \quad f_\lambda(\boldsymbol{\beta}) = q(\boldsymbol{\beta}) + \lambda\bar{h}(\boldsymbol{\beta}). \tag{3.19}$$

Then the group lasso solution at a particular λ can be rewritten as $\widehat{\boldsymbol{\beta}}(\lambda) = \arg\min_{\boldsymbol{\beta}}\{f_\lambda(\boldsymbol{\beta})\} = \arg\min_{\boldsymbol{\beta}}\{q(\boldsymbol{\beta}) + \lambda\bar{h}(\boldsymbol{\beta})\}$. Rather than cycle through each $\boldsymbol{\beta}_j$ and iteratively update each coordinate as `grplasso` and `gglasso` do, GLODE calculates the change in $\boldsymbol{\beta}_j$ as λ changes. Specifically, at a particular λ, the estimated active set $\mathcal{J}$ will contain all j such that $\widehat{\boldsymbol{\beta}} \neq 0$. For a change in λ of $\Delta\lambda$, the difference in the estimates is denoted by $\Delta\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}(\lambda+\Delta\lambda) - \widehat{\boldsymbol{\beta}}(\lambda)$ and satisfies

$$\Delta\widehat{\boldsymbol{\beta}} = \arg\min_{\Delta\boldsymbol{\beta}}\{f_{\lambda+\Delta\lambda}(\widehat{\boldsymbol{\beta}}(\lambda) + \Delta\widehat{\boldsymbol{\beta}}) - f_\lambda(\widehat{\boldsymbol{\beta}}(\lambda)\} \tag{3.20}$$

subject to $\Delta\widehat{\boldsymbol{\beta}}_j = 0$ for all $j \notin \mathcal{J}$. From λ to $\lambda + \Delta\lambda$, $\widehat{\boldsymbol{\beta}}_j = 0$ and $\Delta\widehat{\boldsymbol{\beta}} = 0$ for all $j \notin \mathcal{J}$. So we only need to focus on $\Delta\widehat{\boldsymbol{\beta}}_j$ for $j \in \mathcal{J}$. Define $\boldsymbol{X}_\mathcal{J} = [\cdots \boldsymbol{X}_k \cdots]$, $\widehat{\boldsymbol{\beta}}_\mathcal{J} = [\cdots \widehat{\boldsymbol{\beta}}_k^T \cdots]^T$, and $\Delta\widehat{\boldsymbol{\beta}}_\mathcal{J} = [\cdots \Delta\widehat{\boldsymbol{\beta}}_k \cdots]^T$ for $k \in \mathcal{J}$. Then (3.20) can be rewritten as

$$\begin{aligned}\Delta\widehat{\boldsymbol{\beta}}_\mathcal{J} = \arg\min_{\Delta\boldsymbol{\beta}} \Big\{ &-\boldsymbol{y}^T\boldsymbol{X}_\mathcal{J}\Delta\widehat{\boldsymbol{\beta}}_\mathcal{J} + \widehat{\boldsymbol{\beta}}_\mathcal{J}^T(\lambda)\boldsymbol{X}_\mathcal{J}^T\boldsymbol{X}_\mathcal{J}\Delta\widehat{\boldsymbol{\beta}}_\mathcal{J} + \frac{1}{2}\Delta\widehat{\boldsymbol{\beta}}_\mathcal{J}^T\boldsymbol{X}_\mathcal{J}^T\boldsymbol{X}_\mathcal{J}\Delta\widehat{\boldsymbol{\beta}}_\mathcal{J} + \\ &(\lambda + \Delta\lambda)\left[h(\widehat{\boldsymbol{\beta}}_\mathcal{J}(\lambda) + \Delta\widehat{\boldsymbol{\beta}}_\mathcal{J}) - h(\widehat{\boldsymbol{\beta}}_\mathcal{J}(\lambda))\right] + \Delta\lambda h(\widehat{\boldsymbol{\beta}}_\mathcal{J}(\lambda))\Big\},\end{aligned} \tag{3.21}$$

where $h(\boldsymbol{\beta}) = \sum_{j\in\mathcal{J}} w_j||\boldsymbol{\beta}_j||$. Apply a second-order Taylor expansion on $h(\widehat{\boldsymbol{\beta}}_\mathcal{J}(\lambda) + \Delta\widehat{\boldsymbol{\beta}}_\mathcal{J}) - h(\widehat{\boldsymbol{\beta}}_\mathcal{J}(\lambda))$ with respect to $\Delta\widehat{\boldsymbol{\beta}}_\mathcal{J}$ and differentiate with respect to $\Delta\widehat{\boldsymbol{\beta}}_\mathcal{J}$ to obtain

$$-\boldsymbol{X}_\mathcal{J}^T\boldsymbol{y} + \boldsymbol{X}_\mathcal{J}^T\boldsymbol{X}_\mathcal{J}\widehat{\boldsymbol{\beta}}_\mathcal{J}(\lambda) + \lambda\nabla h + \Delta\lambda\nabla h + \left[\boldsymbol{X}_\mathcal{J}^T\boldsymbol{X}_\mathcal{J} + (\lambda + \Delta\lambda)\nabla^2 h\right]\Delta\widehat{\boldsymbol{\beta}}_\mathcal{J} = 0. \tag{3.22}$$

The first KKT condition implies $-\boldsymbol{X}_\mathcal{J}^T\boldsymbol{Y} + \boldsymbol{X}_\mathcal{J}^T\boldsymbol{X}_\mathcal{J}\widehat{\boldsymbol{\beta}}_\mathcal{J}(\lambda) + \lambda\nabla h = 0$, so manipulation of the remaining parts of the derivative yields $\frac{\Delta\widehat{\boldsymbol{\beta}}_\mathcal{J}}{\Delta\lambda} = -\left[\boldsymbol{X}_\mathcal{J}^T\boldsymbol{X}_\mathcal{J} + (\lambda + \Delta\lambda)\nabla^2 h\right]^{-1}\nabla h$. Taking $\Delta\lambda \to 0$ we have the ordinary differential equation

$$\frac{d\widehat{\boldsymbol{\beta}}_\mathcal{J}}{d\lambda} = -\left[\boldsymbol{X}_\mathcal{J}^T\boldsymbol{X}_\mathcal{J} + (\lambda + \Delta\lambda)\nabla^2 h\right]^{-1}\nabla h. \tag{3.23}$$

Yau & Hui [2017] show that as long as $\boldsymbol{X}_\mathcal{J}^T\boldsymbol{X}_\mathcal{J}$ is positive definite, the matrix sum in (3.23)

is invertible.

The ODE in (3.23) must be solved for each segment of λ between two critical points, i.e., between two values of λ where the composition of $\mathcal{J}$ changes. The critical points are found by monitoring two specific quantities. First, for a currently inactive group j, the group will become active when the concurrent correlation $||\boldsymbol{X}_j^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})||$ reaches λw_j in magnitude. The corresponding value of λ will be denoted a critical point and j will enter $\mathcal{J}$. Second, a currently active group $j \in \mathcal{J}$ will become inactive when $||\widehat{\boldsymbol{\beta}}_j|| = 0$. The corresponding value of λ will be denoted a critical point and j will exit $\mathcal{J}$. A caveat to this, though, is $\nabla^2 h$ and ∇h are not well-defined when λ is a critical point. To handle these issues, the following adjustments are made. First, when an inactive group becomes active, $\widehat{\boldsymbol{\beta}}_j$ is set to $\delta \boldsymbol{X}_j^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})/||\boldsymbol{X}_j^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})||$ where δ is some numerical error constant. Second, for an active group to become inactive, the criterion $||\widehat{\boldsymbol{\beta}}_j|| < \zeta$ must be satisfied for some numerical error constant ζ. If satisfied, then set $\widehat{\boldsymbol{\beta}}_j = 0$.

3.3 Alternate Penalties and Algorithm Modifications

Following a Cholesky reparameterization as described at the beginning of Section 3.2, both `gglasso` and GLODE can be used to solve the functional linear regression with penalty (3.6). However, the introduction of a separate smoothing penalty, as in (3.4) and (3.5), requires slight modifications of both algorithms. For simplicity, consider solving the functional linear regression with orthonormal basis functions and penalty (3.4), given in matrix form as

$$\arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2}||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||^2 + \frac{\varphi}{2} \sum_{j=1}^{m} \boldsymbol{\beta}_j^T \boldsymbol{\Omega}_{\boldsymbol{t}} \boldsymbol{\beta}_j + \lambda \sum_{j=1}^{m} w_j ||\boldsymbol{\beta}_j|| \right\}. \tag{3.24}$$

The modifications for both algorithms rely on the differentiability of the smoothing penalty, as described in the following sections. Note that solving this problem does not require a Cholesky reparameterization, but solving the problem with penalty (3.5) does.

3.3.1 `gglasso` Modifications

Recall the notation from Section 3.2.1. With regard to the optimization criterion in (3.24), define $L^*(\boldsymbol{\beta}|\boldsymbol{D})$ as the sum of the least squares loss and the smoothing penalty. Since the

smoothing penalty is differentiable everywhere, the gradient and Hessian are given as

$$\begin{aligned} \frac{\delta L^*}{\delta \boldsymbol{\beta}_k} &= -\frac{1}{N}\boldsymbol{X}_k^T(\boldsymbol{y} - \sum_{j=1}^{m} \boldsymbol{X}_j\boldsymbol{\beta}_j) + \varphi\boldsymbol{\Omega}_t\boldsymbol{\beta}_k \\ \frac{\delta^2 L^*}{\delta \boldsymbol{\beta}_k \boldsymbol{\beta}_K^T} &= \frac{1}{N}\boldsymbol{X}_k^T\boldsymbol{X}_k + \varphi\boldsymbol{\Omega}_t. \end{aligned} \tag{3.25}$$

Since $\frac{1}{N}\boldsymbol{X}_k^T\boldsymbol{X}_k$ and $\boldsymbol{\Omega}_t$ are both nonnegative definite and $\varphi > 0$, we have $\frac{\delta^2 L^*}{\delta \boldsymbol{\beta}_k \boldsymbol{\beta}_K^T} \gtrsim 0$. Thus, let $\boldsymbol{H}_k^* = \frac{1}{N}\boldsymbol{X}_k^T\boldsymbol{X}_k + \varphi\boldsymbol{\Omega}_t$, let $\gamma_k^* = (1+\varepsilon^*)\eta_k^*$ with η_k^* denoting the largest eigenvalue of $\boldsymbol{H}_k^*$, let $U_k^* = \frac{\delta L^*}{\delta \boldsymbol{\beta}_k}$, and follow the computational steps from Section 3.2.1 to compute the update $\boldsymbol{\beta}^{t+1}$. A simple alteration to the `gglasso` source code allows for implementation of this modified algorithm.

However, using this algorithm in preliminary simulations quickly revealed a key limitation. Notice that `gglasso` is a block-gradient descent algorithm with step size $1/\gamma_k$ that depends on the maximum eigenvalue η_k of H_k. As η_k increases, the step size decreases. In the modified version of the algorithm, larger values of the smoothing parameter φ will yield increased values of η_k^* and smaller step sizes. When φ is too large, the step size shrinks dramatically and leads to greatly increased computational cost. Since one of the goals for using a separate smooth-sparse penalty is decreased computation time, we turn to modifying the GLODE algorithm.

3.3.2 GLODE Modifications

Now, recall the notation from Section 3.2.2. Since the smoothing penalty is everywhere differentiable, redefine $q(\boldsymbol{\beta})$ as the sum of the least squares loss and the smoothing penalty for a fixed value of φ. Again, rewrite the group lasso problem at a particular λ as $\widehat{\boldsymbol{\beta}}(\lambda) = \arg\min_{\boldsymbol{\beta}}\{q(\boldsymbol{\beta}) + \lambda\bar{h}(\boldsymbol{\beta})\}$ and seek to minimize $\Delta\widehat{\boldsymbol{\beta}}$. Differentiation of the group lasso problem with respect to $\boldsymbol{\beta}_j$ yields the KKT conditions,

$$\boldsymbol{X}_j^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) - \varphi\boldsymbol{\Omega}_t\boldsymbol{\beta}_j = \lambda w_j \boldsymbol{\beta}_j / ||\boldsymbol{\beta}_j|| \qquad \text{if } j \in S \tag{3.26}$$

$$\boldsymbol{X}_j^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = \lambda w_j \delta_j \qquad \text{otherwise,} \tag{3.27}$$

where $||\delta_j|| \leq 1$ and $S = \{j : \boldsymbol{\beta}_j \neq 0\}$, and substitution in $\Delta\widehat{\boldsymbol{\beta}}_{\mathcal{J}}$ with application of a second-order Taylor expansion on $h(\widehat{\boldsymbol{\beta}}_{\mathcal{J}}(\lambda) + \Delta\widehat{\boldsymbol{\beta}}_{\mathcal{J}}) - h(\widehat{\boldsymbol{\beta}}_{\mathcal{J}}(\lambda))$ with respect to $\Delta\widehat{\boldsymbol{\beta}}_{\mathcal{J}}$ yields

$$\Delta\widehat{\boldsymbol{\beta}}_{\mathcal{J}} = \arg\min_{\Delta\boldsymbol{\beta}} \Big\{ \Big[-\boldsymbol{y}^T \boldsymbol{X}_{\mathcal{J}} + \widehat{\boldsymbol{\beta}}_{\mathcal{J}}^T(\lambda)\boldsymbol{X}_{\mathcal{J}}^T\boldsymbol{X}_{\mathcal{J}} + \varphi\widehat{\boldsymbol{\beta}}_{\mathcal{J}}^T(\lambda)(\boldsymbol{I}_{|\mathcal{J}|} \otimes \boldsymbol{\Omega}_t) + (\lambda + \Delta\lambda)(\nabla h)^T \Big] \Delta\widehat{\boldsymbol{\beta}}_{\mathcal{J}} + \frac{1}{2}\Delta\widehat{\boldsymbol{\beta}}_{\mathcal{J}}^T \left[\boldsymbol{X}_{\mathcal{J}}^T\boldsymbol{X}_{\mathcal{J}} + \varphi(\boldsymbol{I}_{|\mathcal{J}|} \otimes \boldsymbol{\Omega}_t) + (\lambda + \Delta\lambda)(\nabla^2 h) \right] \Delta\widehat{\boldsymbol{\beta}}_{\mathcal{J}} + \Delta\lambda h(\widehat{\boldsymbol{\beta}}_{\mathcal{J}}(\lambda)) \Big\}. \tag{3.28}$$

Differentiation with respect to $\Delta\widehat{\boldsymbol{\beta}}_{\mathcal{J}}$ yields

$$\begin{aligned} -\boldsymbol{X}_{\mathcal{J}}^T\boldsymbol{y} + \boldsymbol{X}_{\mathcal{J}}^T\boldsymbol{X}_{\mathcal{J}}\widehat{\boldsymbol{\beta}}_{\mathcal{J}}(\lambda) + \varphi(\boldsymbol{I}_{|\mathcal{J}|} \otimes \boldsymbol{\Omega}_t)\widehat{\boldsymbol{\beta}}_{\mathcal{J}}(\lambda) + \lambda\nabla h + \Delta\lambda\nabla h + \\ \left[\boldsymbol{X}_{\mathcal{J}}^T\boldsymbol{X}_{\mathcal{J}} + \varphi(\boldsymbol{I}_{|\mathcal{J}|} \otimes \boldsymbol{\Omega}_t) + (\lambda + \Delta\lambda)\nabla^2 h \right] \Delta\widehat{\boldsymbol{\beta}}_{\mathcal{J}} = 0, \end{aligned} \tag{3.29}$$

where the first KKT condition implies $-\boldsymbol{X}_{\mathcal{J}}^T\boldsymbol{y} + \boldsymbol{X}_{\mathcal{J}}^T\boldsymbol{X}_{\mathcal{J}}\widehat{\boldsymbol{\beta}}_{\mathcal{J}}(\lambda) + \varphi(\boldsymbol{I}_{|\mathcal{J}|} \otimes \boldsymbol{\Omega}_t)\widehat{\boldsymbol{\beta}}_{\mathcal{J}}(\lambda) + \lambda\nabla h = 0$. Thus, manipulation of the remaining parts of the derivative yields

$$\frac{\Delta\widehat{\boldsymbol{\beta}}_{\mathcal{J}}}{\Delta\lambda} = -\left[\boldsymbol{X}_{\mathcal{J}}^T\boldsymbol{X}_{\mathcal{J}} + \varphi(\boldsymbol{I}_{|\mathcal{J}|} \otimes \boldsymbol{\Omega}_t) + (\lambda + \Delta\lambda)\nabla^2 h \right]^{-1} \nabla h, \tag{3.30}$$

and taking $\Delta\lambda \to 0$ produces the ordinary differential equation

$$\frac{d\widehat{\boldsymbol{\beta}}_{\mathcal{J}}}{d\lambda} = -\left[\boldsymbol{X}_{\mathcal{J}}^T\boldsymbol{X}_{\mathcal{J}} + \varphi(\boldsymbol{I}_{|\mathcal{J}|} \otimes \boldsymbol{\Omega}_t) + (\lambda + \Delta\lambda)\nabla^2 h \right]^{-1} \nabla h. \tag{3.31}$$

The active set is still monitored as λ changes, where a currently inactive group j will become active when the *smoothed* concurrent correlation $||\boldsymbol{X}_j^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) - \varphi\boldsymbol{\Omega}_t\boldsymbol{\beta}_j||$ reaches λw_j. In this case, $\widehat{\boldsymbol{\beta}}_j$ is set to $\{\delta\boldsymbol{X}_j^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) - \boldsymbol{\Omega}_t\boldsymbol{\beta}_k\}/||\boldsymbol{X}_j^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) - \boldsymbol{\Omega}_t\boldsymbol{\beta}_k||$. No modifications are needed for when an active group transitioning to inactive status.

To incorporate adaptive weights for smoothing, $\boldsymbol{\Omega}_t$ in the jth block of the block diagonal matrix $(\boldsymbol{I}_{|\mathcal{J}|} \otimes \boldsymbol{\Omega}_t)$ is replaced by some matrix $\boldsymbol{\Omega}_j$. This matrix is $\boldsymbol{\Omega}_j = w_{j,t}\boldsymbol{\Omega}_t$ for the univariate effects model in (3.1) and $\boldsymbol{\Omega}_j = w_{j,t}(\boldsymbol{\Omega}_t \otimes \boldsymbol{I}) + w_{j,z}(\boldsymbol{I} \otimes \boldsymbol{\Omega}_z)$ for the bivariate effects model in (1.1). The adaptive weights for sparsity are incorporated in the first and second sub-differentials of the group lasso penalty when fitting the regression with penalty (3.4). When penalty (3.5) is used, the sparsity weight is incorporated within the square root as in shown in (1.2).

3.3.3 Tuning Parameter Selection

Typically in V-fold cross-validation, the data is randomly partitioned into V folds, the models are trained with the vth fold withheld and then validated on the vth fold. The errors from validation are averaged, and the set of tuning parameters that achieves the minimum validation error is selected. For time-ordered data like the EMG signals, it is recommended to partition the data into V equally-sized sequential chunks rather than random partitions, a strategy known as V-fold block cross-validation [Roberts et al., 2017]. Additionally, the 1-Standard Error (1-SE) Rule has gained popularity [Hastie et al., 2009; Krstajic et al., 2014; Yang & Zou, 2014]. The rule starts by calculating the standard error of the V validation errors for each tuning parameter(s). Then, for the minimum average error, we add its corresponding standard error to create a threshold denoted as $\min APE + SE$. All tuning parameter settings and their corresponding estimates are collected whose average prediction error falls below the threshold. In the case of a single tuning parameter, we select among this collection the largest tuning parameter as it corresponds to the most parsimonious model, as measured by the penalty of choice. This is equivalent to ranking models according to the penalty value of their estimates. That is, we may use the penalty as a measure of model complexity, and we prefer the model that is least complex but still has good prediction performance relative to the model with the minimum average prediction error. Stallrich et al. [2020] extend this idea for the case of multiple tuning parameters, such as in smooth-sparse functional regression, that uses the penalty function with all tuning parameters set to 1 as a measure of model complexity. The tuning parameter set that minimizes this measure is selected.

Under the extended 1-SE Rule, we first measured model complexity with the smooth-sparse penalty that was used for estimation. However, the squared smoothness norm in the separate and combined penalties unintentionally places greater emphasis on smoothing because its norm is squared. The purpose of squaring the norm in the penalty was primarily to speed up computations for estimation. Using the same penalties to measure model complexity then would favor models with smoother $\hat{\gamma}_j$ more than sparse models. For proper comparison, we use the joint penalty to measure model complexity regardless of the penalty used during optimization.

When multiple stages are performed with updated adaptive weights, it is possible for the current stage's estimates to yield worse overall predictions and the estimated models' predictions would deteriorate in each subsequent stage. This can happen when the tuning parameter space is not well explored. To avoid this, we retain the previous stage's $\min APE + SE$ and continue using this threshold unless it is larger than the current stage's $\min APE +$

SE. If the current stage's APE's all are larger than the previous stage's min $APE + SE$, we recommend adjusting the density of the tuning parameter grids for this stage and rerunning the estimation until the issue no longer occurs. Alternatively, one could terminate the SAFE procedure and conclude with the previous stage's best model, according to the 1-SE rule.

Note to committee: in my previous simulation studies, I had a faulty decision rule where if none of the next stage's APE values were below the current min $APE + SE$ threshold, the model would be set to the null model (no selected covariates). This was an oversight in my coding and has since been corrected. Now when this occurs, we simply retain the best model from the previous stage (according to the 1-SE rule) for the current and all future stages.

3.4 Simulation Study

For the scenarios described below, 5-stage SAFE is conducted with each of the three penalties. Performance is evaluated in terms of computation time, selection, and prediction across 100 randomly generated data sets using the following metrics:

- **Model size, $|\widehat{\mathcal{J}}|$:** the number of groups in the estimated active set;
- **True positive rate (TPR):** $TP/|S|$, where $TP = |\mathcal{J} \cap S|$;
- **False positive rate (FPR):** $FP/|S^c|$, where S^c is the complement of S and $FP = |\mathcal{J} \cap S^c|$;
- **Average prediction error (APE):** $\sum_{i=1}^{N_0}(y_{0i} - \widehat{y}_{0i})^2/N_0$ for N_0 observations y_{0i} in the randomly generated validation data;

In each stage, the optimal tuning parameters are selected with 5-fold CV using the extended 1-SE Rule described in the previous section. As stated earlier, it is possible for all tuning parameter sets in future stages to yield models with APE greater than the min $APE + SE$ threshold. Increasing the tuning parameter grid density for all five stages is infeasible for this simulation study because dense models would require much more computation. Thus, we carry the most recent best model forward from before SAFE terminated for reporting the selection and prediction metrics, report the number of models actually fit during each stage, and report stage-wise mean computation time accordingly.

3.4.1 Data Generation

Consider the model in (3.1). Similar to Tutz & Gertheiss [2010], m functional predictors are generated. Specifically, for 100 equally-spaced time points in $\mathcal{T} = [0, 1]$, the ith observation

of the jth predictor is given by

$$X_{ij}(t) = \{\widehat{\sigma}(t)\}^{-1} \left(\sum_{q=1}^{5} [a_{ijq} \sin\{2\pi t(5 - a_{ijq})\} - m_{ijq}] - \widehat{\mu}(t) \right), \tag{3.32}$$

where $a_{ijq} \overset{iid}{\sim} U(0,5)$ and $m_{ijq} \overset{iid}{\sim} U(0,2\pi)$ for $i = 1, ..., N$ and $j = 1, ..., m$. The quantity $\widehat{\mu}(t)$ is the empirical mean of the summation term at t and $\widehat{\sigma}(t)$ is a point-wise scaling factor such that $\text{Var}\{X_{ij}(t)\} = 0.1$ for all $t \in \mathcal{T}$, where both of these quantities are estimated from 1000 random generations of the above summation. The response is generated as

$$y_i = \sum_{j \in S} \int_{\mathcal{T}} X_{ij}(t)\gamma_j(t)dt + \varepsilon_i = \mu_i + \varepsilon_i. \tag{3.33}$$

The noise $\varepsilon_i \overset{iid}{\sim} N(0, \sigma_\varepsilon^2)$ with σ_ε^2 determined by the signal-to-noise ratio, SNR$= \text{Var}(\mu_i)/\text{Var}(\varepsilon_i)$, with SNR$\in \{1, 10\}$. The response is then centered prior to analysis.

A total of $N = 500$ observations and $m = 16$ functional predictors are generated, where the active set of coefficients varies in size, $|S| \in \{2, 8, 12\}$. The functional forms of the active coefficients are given in Table 3.1. For each $|S|$, the nonzero coefficients are γ_j, $j = 1, ..., |S|$, with one exception. For the sparse univariate effects model with $|S| = 2$, both a smooth effects and rough effects model are considered. The smooth effects model has active signals $j = 1, 2$, whereas the rough effects model has active signals $j = 10, 11$. Finally, the functional effect approximation detailed in Section 3.2 utilizes 10 orthonormal cubic B-spline basis functions, and we approximate $\int_{\mathcal{T}} X_{ij}(t)\gamma_j(t)dt$ with a Riemann sum since the $X_{ij}(t)$ are generated on a dense grid of time points. The smoothing parameter is selected from the set of values $\log \varphi \in \{-15, -14, ..., 2, 3\}$ and the sparsity parameter changes in increments of $\Delta\lambda = 0.05$.

j	$\gamma_j(t)$	j	$\gamma_j(t)$
1	$1 + \sqrt{2}\cos(\pi t)$	7	$\sin^2(2t)\cos(3t) + \sin^2(t + 4)$
2	$1 + t + 0.5t^2$	8	$1 + \sqrt{2}\cos(0.3\pi t)$
3	$\sin^2(7t)\cos(t) + \sin^2(t - 4)$	9	$1 + e^{t^2}\sin(10t)$
4	$\sin(2t)\cos^2(10t) + \sin(t - 0.5)$	10	$1 + e^{-t}\sin(-10t)$
5	$1 + \sqrt{2}\sin(\pi t/2)$	11	$1 + e^{t^2}\cos(10t)$
6	$\sin(5t)\cos^2(3t) + \sin(t + 1)$	12	$1 + e^{-t}\cos(10t)$

Table 3.1 Functional forms of active coefficients.

3.4.2 Results

The results for the four scenarios are given in Tables 3.2-3.5. There are common trends that persist through all four of the considered models, but for the purpose of discussion consider the scenario in Table 3.2 with $|S| = 2$ smooth coefficient functions. The first noticeable trend is that, for a given SNR, the combined penalty (P^{combo}) is consistently competitive with, but rarely outperforms, the joint penalty (P^{joint}) in terms of selection. For both SNR's in Table 3.2, models have a perfect $TPR = 1$, and when $SNR = 1$ both P^{joint} and P^{combo} yield equally-sized Stage 5 models, on average ($Size = 2.05$). However, when $SNR = 10$, P^{joint} yields a sparser mean final model ($Size = 3.97$) than the separate penalty (P^{sep}, $Size = 4.78$) and P^{combo} ($Size = 5.01$). In such cases P^{combo} yields relatively higher FPR's, but this is often paired with having to carry forward more 'last best models' from earlier stages, as represented by lower stage-wise $Model$ counts, particularly when $SNR = 10$. Note that the elevated FPR's in Table 3.5 are not as concerning because there are only 4 possible false positives when $|S| = 12$. As previously mentioned, consistently good model fits throughout the CV process will yield very small CV SE's, resulting in a tight $\min APE + SE$ bound that is difficult to satisfy in later stages for a fixed tuning parameter grid. It is likely that a denser tuning parameter grid would enable better models to be estimated at later stages, yielding more comparable results between the joint and combined penalties.

Conversely, this aspect of the tuning parameter selection protocol is not as significant of an issue for $SNR = 1$ in sparse models. For instance in Table 3.2, SAFE terminates for $10 - 15$ models in Stage 3 and for up to 47 models in Stage 5 when $SNR = 10$, but SAFE terminates for no more than 12 models in Stage 5 when $SNR = 1$. However, keeping SNR fixed while increasing $|S|$ has the potential to hinder selection performance, regardless of the actual value chosen for SNR. The reason for this lies in its definition. As $|S|$ increases, the numerator of the SNR, that is, the combined signal strength, also increases. In order to maintain a constant SNR, the strength of the random error must also increase, which reduces the relative strength of an individual predictor to the overall noise.

Although P^{combo} never definitively outperforms P^{joint} with respect to selection, a key benefit of P^{combo} is observed in the computation time. P^{joint} consistently requires more time in the first stage (82.74 vs 46.55, $SNR = 1$), and continues to do so in later stages relative to P^{combo} even when similar levels of model sparsity are obtained by both methods. For example, on average, P^{joint} requires 82.74 seconds compared to 46.55 seconds for P^{combo} when $SNR = 1$ in Table 3.2. As previously stated, increasing the number of considered tuning parameters is likely to improve the selection performance of P^{combo}. However, doing so for P^{joint} (for an equal comparison) would further inflate the computation time and lend

more favor toward P^{combo}. In fact, we noted in Section 3.1 that an expected deficiency of P^{joint} was the ability to perform accurate selection and estimation of sparse rough models. It appears that the minimum φ values were small enough to overcome this, but combinations of small φ and small λ require more computation because many coefficient estimates will be non-zero. Interestingly, though, P^{sep} slightly outperforms P^{combo} for sparse rough models (2.03 vs 2.21), as shown in Table 3.3.

Remark 3.4.1. Due to presentation format, the APE values appear identical across penalties and stages in many cases. The values are consistent up to three decimal places, but do exhibit variation when more digits are printed. Additionally, APE increases slightly across stages for the denser models with $|S| \in \{8, 12\}$ and $SNR = 1$, which contradicts expectations. However, this is a potential result under the tuning parameter selection protocol due to the flexibility of the 1-SE Rule. The minimum CV APE must be less than the min APE+SE threshold, not less than the previous stage's min APE. Thus, small increases in overall APE can occur.

3.5 Application to EMG Data

The performances of the three penalties are further explored by applying a 2-stage SAFE procedure to the six finger movement data sets that were described in Chapter 1. The model described in Section 2.5.2 is used here with the necessary approximations. Recall the model is

$$\mathbb{E}[y_i|X_{i1}, ..., X_{im}, z_i] = \sum_{j=1}^{m} \int_{-\delta}^{0} X_{ij}(t)\gamma_j(t, z_i)dt, \tag{3.34}$$

with current velocity y_i, bivariate coefficient functions $\gamma_j(t, z)$ of time, $t \in [-\delta, 0]$, and position, $z \in \mathcal{Z}$. Each $\gamma_j(t, z)$ is approximated by a tensor product of orthogonal cubic B-spline basis functions, $\{\omega_\ell(\cdot)\}_{\ell=1}^{L}$ and $\{\tau_m(\cdot)\}_{m=1}^{M}$ with $L = M = 10$, as $\gamma_j(t, z) \approx \boldsymbol{\omega}^T(t)\boldsymbol{B}_j\boldsymbol{\tau}(z)$. The above functional model is then approximated by the linear model

$$\mathbb{E}[y_i|\widetilde{\boldsymbol{X}}_{i1}, ..., \widetilde{\boldsymbol{X}}_{im}, z_i] = \sum_{j=1}^{m} \widetilde{\boldsymbol{X}}_{ij}\boldsymbol{\beta}_j, \tag{3.35}$$

where $\widetilde{\boldsymbol{X}}_{ij} = \left\{ \sum_{r=-\delta}^{0} X_{ij}(t_r)\boldsymbol{\omega}(t_r)^T \right\} \otimes \boldsymbol{\tau}(z_i)$ and $\boldsymbol{\beta}_j$ is the vectorized version of $\boldsymbol{B}_j$. To increase the rank of the Gram matrix $\boldsymbol{X}^T\boldsymbol{X}$ for the GLODE algorithm, the data were only thinned by a factor of 8 rather than 20 as in Stallrich et al. [2020]. Finally, with $\varphi_t, \varphi_z > 0$ and the

Penalty	Metric	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
		$SNR = 10$				
Joint	*Size*	5.98 (0.373)	4.49 (0.33)	4.16 (0.332)	3.97 (0.329)	3.97 (0.329)
	TPR	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
	FPR	0.28 (0.027)	0.16 (0.022)	0.13 (0.021)	0.11 (0.021)	0.11 (0.021)
	APE	0.003 (0)	0.003 (0)	0.003 (0)	0.003 (0)	0.003 (0)
	Time	72.17 (0.323)	12.11 (1.839)	5.31 (1.068)	4.08 (1.075)	1.91 (0.208)
	Models	100	100	90	74	53
Sep	*Size*	11.38 (0.529)	6.81 (0.574)	5.37 (0.515)	4.89 (0.496)	4.78 (0.49)
	TPR	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
	FPR	0.67 (0.038)	0.33 (0.041)	0.21 (0.036)	0.16 (0.033)	0.13 (0.032)
	APE	0.003 (0)	0.003 (0)	0.003 (0)	0.003 (0)	0.003 (0)
	Time	48.34 (0.223)	32.62 (2.517)	12.34 (2.106)	4.87 (1.124)	3.27 (0.86)
	Models	100	100	87	72	57
Combo	*Size*	10.12 (0.559)	6.86 (0.551)	5.65 (0.509)	5.11 (0.483)	5.01 (0.476)
	TPR	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
	FPR	0.58 (0.04)	0.33 (0.04)	0.23 (0.037)	0.18 (0.033)	0.17 (0.033)
	APE	0.003 (0)	0.003 (0)	0.003 (0)	0.003 (0)	0.003 (0)
	Time	41.04 (0.135)	28.54 (2.52)	12.75 (2.315)	6.69 (1.907)	3.3 (1.31)
	Models	100	100	85	74	65
		$SNR = 1$				
Joint	*Size*	2.78 (0.147)	2.11 (0.051)	2.06 (0.028)	2.06 (0.028)	2.05 (0.026)
	TPR	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
	FPR	0.06 (0.01)	0.01 (0.004)	0 (0.002)	0 (0.002)	0 (0.002)
	APE	0.029 (0.0001)	0.028 (0.0001)	0.028 (0.0001)	0.028 (0.0001)	0.028 (0.0001)
	Time	82.74 (0.239)	2.81 (0.362)	1.72 (0.083)	1.67 (0.036)	1.64 (0.027)
	Models	100	100	99	97	93
Sep	*Size*	3.29 (0.26)	2.15 (0.09)	2.1 (0.076)	2.1 (0.076)	2.1 (0.076)
	TPR	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
	FPR	0.09 (0.019)	0.01 (0.006)	0.01 (0.005)	0.01 (0.005)	0.01 (0.005)
	APE	0.029 (0.0001)	0.028 (0.0001)	0.028 (0.0001)	0.028 (0.0001)	0.028 (0.0001)
	Time	60.98 (0.433)	4.12 (1.104)	1.44 (0.17)	1.37 (0.164)	1.21 (0.009)
	Models	100	100	99	92	88
Combo	*Size*	2.75 (0.115)	2.09 (0.043)	2.05 (0.026)	2.05 (0.026)	2.05 (0.026)
	TPR	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
	FPR	0.05 (0.008)	0.01 (0.003)	0 (0.002)	0 (0.002)	0 (0.002)
	APE	0.029 (0.0001)	0.028 (0.0001)	0.028 (0.0001)	0.028 (0.0001)	0.028 (0.0001)
	Time	46.55 (0.179)	1.53 (0.116)	1.03 (0.027)	1 (0.015)	0.99 (0.014)
	Models	100	100	100	99	97

Table 3.2 SAFE results given as mean (SE) for the scenario with $|S| = 2$ and smoother coefficient functions. Computation time is given in seconds.

Penalty	Metric	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
		$SNR = 10$				
	Size	9.77 (0.391)	6.8 (0.426)	5.89 (0.419)	5.67 (0.418)	5.53 (0.416)
	TPR	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)
Joint	*FPR*	0.56 (0.028)	0.34 (0.03)	0.26 (0.029)	0.25 (0.029)	0.24 (0.029)
	APE	0.001 (0.0001)	0.001 (0.0001)	0.001 (0.0001)	0.001 (0.0001)	0.001 (0.0001)
	Time	65.15 (0.766)	24.03 (2.008)	11.19 (1.293)	7.16 (0.996)	4.35 (0.863)
	Models	100	99	95	82	64
	Size	11.48 (0.482)	9.03 (0.544)	6.48 (0.467)	5.7 (0.432)	5.58 (0.43)
	TPR	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)
Sep	*FPR*	0.68 (0.034)	0.5 (0.039)	0.32 (0.033)	0.26 (0.031)	0.25 (0.031)
	APE	0.001 (0.0001)	0.001 (0.0001)	0.001 (0.0001)	0.001 (0.0001)	0.001 (0.0001)
	Time	42.98 (0.544)	26.05 (1.889)	18.22 (1.938)	7.92 (1.269)	4.18 (0.702)
	Models	100	99	97	87	79
	Size	11.26 (0.476)	8.58 (0.547)	7.83 (0.547)	7.06 (0.542)	6.71 (0.532)
	TPR	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)
Combo	*FPR*	0.66 (0.034)	0.47 (0.039)	0.41 (0.039)	0.36 (0.039)	0.33 (0.038)
	APE	0.001 (0.0001)	0.001 (0.0001)	0.001 (0.0001)	0.001 (0.0001)	0.001 (0.0001)
	Time	37.89 (0.444)	29.26 (2.031)	15.08 (1.809)	11.24 (1.862)	7.79 (1.8)
	Models	100	99	93	77	65
		$SNR = 1$				
	Size	4.3 (0.224)	2.17 (0.055)	2.04 (0.035)	2.04 (0.035)	2.03 (0.03)
	TPR	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)
Joint	*FPR*	0.17 (0.016)	0.01 (0.004)	0 (0.002)	0 (0.002)	0 (0.002)
	APE	0.013 (0.0001)	0.013 (0.0001)	0.013 (0.0001)	0.013 (0.0001)	0.013 (0.0001)
	Time	77.76 (0.408)	4.95 (0.645)	1.24 (0.053)	1.11 (0.025)	1.11 (0.027)
	Models	100	99	99	98	94
	Size	4.47 (0.252)	2.14 (0.053)	2.04 (0.032)	2.03 (0.03)	2.03 (0.03)
	TPR	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)
Sep	*FPR*	0.18 (0.018)	0.01 (0.003)	0 (0.002)	0 (0.002)	0 (0.002)
	APE	0.013 (0.0001)	0.013 (0.0001)	0.013 (0.0001)	0.013 (0.0001)	0.013 (1e-04)
	Time	54.53 (0.417)	4.11 (0.747)	0.78 (0.036)	0.71 (0.014)	0.7 (0.013)
	Models	100	99	99	98	97
	Size	4.12 (0.225)	2.47 (0.111)	2.29 (0.077)	2.24 (0.07)	2.21 (0.064)
	TPR	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)
Combo	*FPR*	0.15 (0.016)	0.03 (0.008)	0.02 (0.005)	0.02 (0.005)	0.02 (0.004)
	APE	0.013 (0.0001)	0.013 (0.0001)	0.013 (0.0001)	0.013 (0.0001)	0.013 (0.0001)
	Time	43.53 (0.184)	3.29 (0.493)	0.99 (0.127)	0.77 (0.041)	0.73 (0.035)
	Models	100	99	99	99	98

Table 3.3 SAFE results given as mean (SE) for the scenario with $|S| = 2$ and rougher coefficient functions. Computation time is given in seconds.

Penalty	Metric	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
		$SNR = 10$				
Joint	*Size*	12.96 (0.278)	11.78 (0.288)	11.44 (0.286)	11.34 (0.29)	11.31 (0.287)
	TPR	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
	FPR	0.62 (0.035)	0.45 (0.039)	0.41 (0.04)	0.4 (0.041)	0.39 (0.041)
	APE	0.01 (0.0001)	0.01 (0.0001)	0.01 (0.0001)	0.01 (0.0001)	0.01 (0.0001)
	Time	79.19 (0.269)	55.77 (2.671)	36.67 (3.297)	29.32 (3.267)	27.16 (5.17)
	Models	100	100	60	35	16
Sep	*Size*	15.76 (0.09)	15.52 (0.152)	15.44 (0.17)	15.44 (0.17)	15.44 (0.17)
	TPR	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
	FPR	0.97 (0.011)	0.97 (0.017)	0.96 (0.018)	0.96 (0.018)	0.96 (0.018)
	APE	0.01 (0)	0.01 (0)	0.01 (0)	0.01 (0)	0.01 (0)
	Time	62.66 (0.462)	76.09 (1.167)	72.3 (7.55)	65.31 (16.525)	49.66 (20.932)
	Models	100	100	15	6	4
Combo	*Size*	15.14 (0.172)	14.47 (0.248)	14.15 (0.275)	14.08 (0.279)	14.08 (0.279)
	TPR	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
	FPR	0.89 (0.022)	0.83 (0.031)	0.8 (0.035)	0.8 (0.035)	0.81 (0.035)
	APE	0.01 (0)	0.01 (0)	0.01 (0)	0.01 (0)	0.01 (0)
	Time	46.61 (0.197)	52.51 (1.327)	42.57 (2.862)	35.6 (4.251)	28.93 (6.736)
	Models	100	100	52	31	12
		$SNR = 1$				
Joint	*Size*	9.47 (0.256)	7.9 (0.221)	7.67 (0.219)	7.6 (0.215)	7.54 (0.214)
	TPR	0.94 (0.015)	0.89 (0.017)	0.87 (0.017)	0.86 (0.017)	0.86 (0.017)
	FPR	0.24 (0.025)	0.06 (0.016)	0.04 (0.015)	0.04 (0.014)	0.03 (0.013)
	APE	0.102 (0.0014)	0.123 (0.0031)	0.134 (0.0035)	0.137 (0.0035)	0.137 (0.0035)
	Time	71.49 (1.607)	25.42 (1.941)	14.62 (0.86)	13.51 (0.78)	12.71 (0.79)
	Models	100	98	84	73	58
Sep	*Size*	11.41 (0.321)	8.84 (0.309)	8.36 (0.313)	8.2 (0.314)	8.16 (0.317)
	TPR	0.96 (0.015)	0.88 (0.018)	0.84 (0.02)	0.83 (0.02)	0.82 (0.02)
	FPR	0.47 (0.034)	0.13 (0.027)	0.1 (0.026)	0.09 (0.025)	0.09 (0.025)
	APE	0.101 (0.0014)	0.117 (0.0027)	0.125 (0.0033)	0.129 (0.0035)	0.13 (0.0035)
	Time	61.84 (1.557)	38.93 (2.893)	15.49 (1.868)	9.47 (1.019)	8.41 (0.757)
	Models	100	98	72	53	43
Combo	*Size*	10.04 (0.29)	8.38 (0.242)	7.9 (0.22)	7.81 (0.215)	7.71 (0.215)
	TPR	0.95 (0.015)	0.91 (0.017)	0.89 (0.017)	0.88 (0.017)	0.88 (0.017)
	FPR	0.31 (0.029)	0.1 (0.018)	0.06 (0.011)	0.05 (0.01)	0.04 (0.009)
	APE	0.101 (0.0014)	0.119 (0.0032)	0.128 (0.0034)	0.134 (0.0036)	0.137 (0.0036)
	Time	40.81 (0.92)	17.53 (1.565)	8.41 (0.7)	6.52 (0.438)	6.19 (0.344)
	Models	100	98	90	82	72

Table 3.4 SAFE results given as mean (SE) for the scenario with $|S| = 8$. Computation time is given in seconds.

Penalty	Metric	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
		$SNR = 10$				
Joint	*Size*	15.39 (0.107)	14.7 (0.137)	14.5 (0.146)	14.47 (0.145)	14.46 (0.146)
	TPR	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
	FPR	0.85 (0.027)	0.68 (0.035)	0.65 (0.039)	0.65 (0.039)	0.65 (0.04)
	APE	0.017 (0.0002)	0.017 (0.0002)	0.017 (0.0002)	0.017 (0.0002)	0.017 (0.0002)
	Time	75.74 (1.091)	75.74 (1.623)	60.15 (2.291)	53.04 (2.961)	48.97 (3.842)
	Models	100	100	56	31	13
Sep	*Size*	16 (0)	15.89 (0.063)	15.81 (0.084)	15.81 (0.084)	15.81 (0.084)
	TPR	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
	FPR	1 (0)	0.97 (0.016)	0.96 (0.02)	0.96 (0.02)	0.96 (0.02)
	APE	0.018 (0.0001)	0.018 (0.0001)	0.018 (0.0001)	0.018 (0.0001)	0.018 (0.0001)
	Time	69.29 (1.193)	88.8 (1.514)	86.26 (6.125)	75.29 (14.593)	37.13 (0)
	Models	100	100	15	6	1
Combo	*Size*	15.88 (0.046)	15.82 (0.054)	15.8 (0.057)	15.8 (0.057)	15.8 (0.057)
	TPR	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
	FPR	0.97 (0.011)	0.96 (0.013)	0.96 (0.013)	0.96 (0.013)	0.96 (0.013)
	APE	0.018 (0.0001)	0.018 (0.0001)	0.018 (0.0001)	0.018 (0.0001)	0.018 (0.0001)
	Time	45.91 (0.674)	53.45 (0.907)	51.9 (1.532)	50.58 (1.948)	47.86 (3.791)
	Models	100	100	43	19	6
		$SNR = 1$				
Joint	*Size*	13.28 (0.146)	12.46 (0.155)	12.22 (0.161)	12.16 (0.161)	12.13 (0.162)
	TPR	0.98 (0.004)	0.95 (0.006)	0.94 (0.006)	0.94 (0.006)	0.94 (0.007)
	FPR	0.38 (0.032)	0.19 (0.028)	0.16 (0.028)	0.16 (0.028)	0.16 (0.028)
	APE	0.153 (0.0006)	0.169 (0.003)	0.177 (0.0034)	0.181 (0.0037)	0.182 (0.0037)
	Time	56.63 (0.198)	42.19 (1.363)	34.62 (1.175)	32.38 (1.25)	31.26 (1.429)
	Models	100	100	77	61	47
Sep	*Size*	14.65 (0.147)	13.55 (0.194)	13.21 (0.207)	13.16 (0.21)	13.15 (0.211)
	TPR	0.99 (0.003)	0.96 (0.005)	0.95 (0.007)	0.95 (0.007)	0.95 (0.007)
	FPR	0.7 (0.032)	0.49 (0.042)	0.45 (0.044)	0.46 (0.045)	0.47 (0.045)
	APE	0.152 (0.0006)	0.159 (0.0021)	0.168 (0.0032)	0.17 (0.0035)	0.171 (0.0035)
	Time	55.16 (0.633)	57.46 (1.657)	35.85 (2.052)	29.72 (2.067)	25.53 (1.989)
	Models	100	100	50	35	22
Combo	*Size*	13.64 (0.153)	12.73 (0.163)	12.45 (0.16)	12.38 (0.16)	12.34 (0.158)
	TPR	0.98 (0.004)	0.96 (0.005)	0.96 (0.006)	0.95 (0.006)	0.95 (0.006)
	FPR	0.47 (0.033)	0.27 (0.033)	0.22 (0.03)	0.2 (0.03)	0.19 (0.029)
	APE	0.153 (0.0006)	0.164 (0.0028)	0.173 (0.0034)	0.177 (0.0035)	0.177 (0.0035)
	Time	32.61 (0.141)	25.02 (0.952)	17.64 (0.777)	15.28 (0.711)	13.72 (0.69)
	Models	100	100	81	68	50

Table 3.5 SAFE results given as mean (SE) for the scenario with $|S| = 12$. Computation time is given in seconds.

penalty matrix $\boldsymbol{Q}_\varphi = ||\gamma_j||_2 + \varphi_t||\gamma''_{j,t}||_2 + \varphi_z||\gamma''_{j,z}||_2$, the model reparameterization described in Section 3.2 is used to obtain the group lasso optimization criterion

$$\sum_{i=1}^{N}(y_i - \sum_{j=1}^{m} \boldsymbol{W}_{ij}^T \widetilde{\boldsymbol{\beta}}_j)^2 + \lambda \sum_{j=1}^{m} ||\widetilde{\boldsymbol{\beta}}_j||_2, \tag{3.36}$$

for $\boldsymbol{W}_{ij} = \boldsymbol{R}_\varphi^{-1}\widetilde{\boldsymbol{X}}_{ij}$, $\widetilde{\boldsymbol{\beta}}_j = \boldsymbol{R}_\varphi \boldsymbol{\beta}_j$, and $\lambda > 0$. The smoothing parameters were selected from the logarithmic range of values $\{-15, -12, -9, -6, -3, 0\}$ and the sparsity parameter changed by increments of $\Delta\lambda = 0.05$. Tuning parameter selection is done with 5-fold block cross-validation with the extended 1-SE Rule.

Table 3.6 presents model size, true/false positives, average prediction error (APE) with validation occurring on the five EMG data sets not used to train the model, and computation time in minutes for each stage. First, notice that some of the selection and results differ from the original application of SAFE to the EMG data in Stallrich et al. [2020, Table 1]. This is likely due to the smaller level of data thinning conducted here, different tuning parameter ranges, as well as algorithmic differences between GLODE and `gglasso`. However, P^{sep} and P^{combo} retain an extra signal for one of the consistent movement data sets, specifically the externally generated signal, which is an explicit concern about the selection performance with these penalties. Another general concern pertains to the increase in APE between stages one and two of the SAFE algorithm for two of the random movement data sets. Models should theoretically improve in prediction performance with the adaptive weights guiding the coefficient estimates. Further investigation is required to determine the source of these increases. What is not concerning, though, is the often stark difference in computation time between P^{combo} and P^{joint}. Even though P^{joint} yields a relatively sparse model in the first stage, computation time for the second stage does not decrease as expected and occasionally increases. Meanwhile, the stage two time under the P^{combo} always decreases significantly due to the sparsity induced in stage one. Also with regard to computation time is that two stages of SAFE generally yield ideal selection and can be completed in a matter of minutes with the GLODE algorithm. As will be seen in the next chapter, the SAFE algorithm that utilizes `gglasso` requires multiple hours.

The estimated effects are also plotted across time and position to evaluate and compare the effect estimation to the results of Stallrich et al. [2020, bottom of Figure 5]. Figure 3.2 shows the estimated effects of EMGs 7 and 12 in the first consistent finger movement data set for each of the penalties. Although all of the effects are estimated on a slightly smaller scale, the general effect patterns of both EMGs are relatively consistent for each of the penalties. Effect interpretation remaining intact under the GLODE algorithm further supports its use

Data	Penalty	Time	TP|FP	APE	Time	TP|FP	APE
		Consistent			*Random*		
F1	Joint	16.9	2|1	0.27 (0.072)	10.5	2|0	0.32 (0.048)
		10.4	2|0	0.26 (0.072)	4.3	2|0	0.29 (0.031)
	Sep	20.1	3|0	0.28 (0.075)	12.7	2|0	0.34 (0.051)
		9	3|0	0.24 (0.069)	3.9	2|0	0.38 (0.019)
	Combo	15.4	2|0	0.28 (0.075)	11	2|0	0.32 (0.05)
		2.4	2|0	0.25 (0.069)	2.1	2|0	0.3 (0.031)
F2	Joint	16.1	2|1	0.29 (0.08)	11.6	2|0	0.52 (0.078)
		11.7	2|0	0.27 (0.077)	5.7	2|0	0.77 (0.158)
	Sep	18.2	2|2	0.29 (0.081)	22.1	3|1	0.58 (0.104)
		10.5	2|0	0.28 (0.076)	11.9	3|0	0.8 (0.166)
	Combo	17.8	2|2	0.29 (0.081)	12.9	3|0	0.49 (0.071)
		5.2	2|0	0.28 (0.079)	6.1	3|0	0.75 (0.154)
F3	Joint	14.9	3|0	0.27 (0.079)	13	2|1	0.75 (0.146)
		32.4	2|0	0.26 (0.074)	21.7	2|0	0.89 (0.186)
	Sep	16.6	3|1	0.28 (0.082)	14.2	3|1	0.74 (0.137)
		10.3	3|1	0.27 (0.074)	12.4	2|0	0.94 (0.202)
	Combo	15.1	2|1	0.28 (0.082)	14.8	3|1	0.75 (0.138)
		5	2|1	0.27 (0.074)	5	2|0	0.88 (0.183)

Table 3.6 SAFE results across two stages for each of the penalties when applied to the EMG data sets capturing consistent (left) and random (right) finger movement. Computation time is presented in minutes.

in future analyses.

3.6 Discussion

The primary contributions of this work are an investigation of two alternate smooth-sparse penalties and the implementation of a more efficient group lasso algorithm within the SAFE framework. The combined penalty was found to be competitive with the joint penalty, yielding similar selection results and requiring less computation time, overall. Although the alternative penalties did not reduce the number of stages needed for accurate EMG selection, using the GLODE algorithm did and thus reduced overall computation time. Further, a deeper understanding of the limitations of group-wise majorization descent as implemented in the `gglasso` package was gained.

Although GLODE is noticeably more efficient than `gglasso`, it is currently limited by the requirement that the Gram matrix be positive definite. In high-dimensional settings such as functional data, this is likely not the case unless a high degree of sparsity is assumed. This limits the current algorithm's ability to explore the full tuning parameter space and resulting model space. Yau & Hui [2017] note that to allow for the Gram matrix to only be nonnegative definite, the matrix inversion in the ordinary differential equation could be changed to a

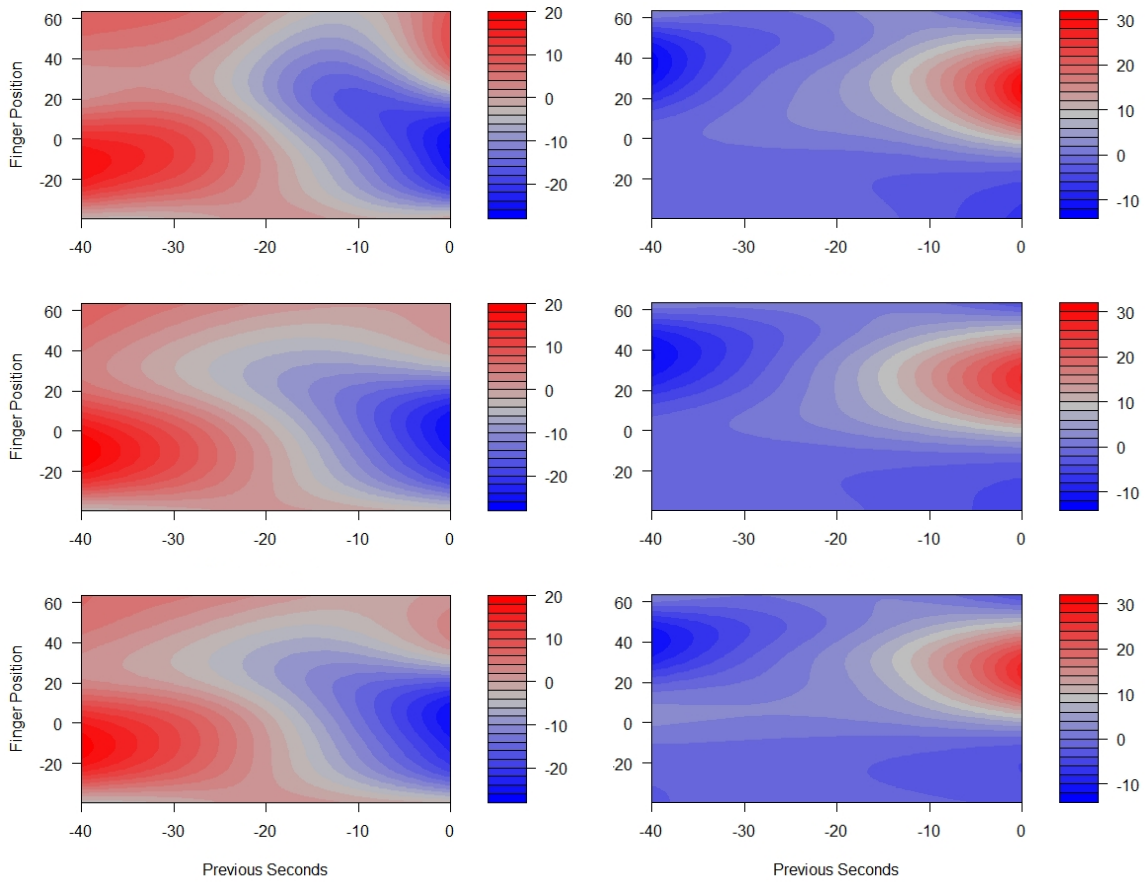

Figure 3.2 Estimated effects of EMGs 7 (left) and 12 (right) from the second consistent movement data set across time and position for the joint (top), separate (middle), and combined (bottom) penalties.

generalized inverse. For future use of the GLODE algorithm with high-dimensional data, this alteration should be made.

The major limitation that made `gglasso` undesirable for this investigation was the reduction in step size, and hence increased computation time, for even moderately-sized smoothing parameters. Working with the separate penalty, the step size depends on the inverse maximum eigenvalue of the sum of the Gram matrix and scaled smoothing matrix $(\boldsymbol{X}_j^T\boldsymbol{X}_j+\varphi\boldsymbol{\Omega}_t)$. As φ increases, the maximum eigenvalue increases and so the step size decreases. When adaptive weights are added to the equation, this problem is exacerbated. Stallrich et al. [2020] noted that the magnitude of the adaptive smoothing weights can cause instability and so are not permitted to exceed a certain value within the SAFE algorithm—and that was with the joint smooth-sparse penalty.

These observations lead to two important areas of future research. First, is there a better

way to calculate adaptive smoothing weights? The current formula is $w_1 = 1/||\gamma''(t)||$. If $\gamma(t)$ is relatively smooth, the weight has the potential to be quite large. Under the separate or combined penalty, this is problematic for reason already described. Under the joint penalty, this may also pose a problem during the Cholesky reparameterization in combination for relatively large smoothing parameters. To control the weight magnitudes more effectively than just setting a maximum value, it could help to increase the root degree in the weight formula. That is, the current formula has a power of $2/d$ with $d = 2$ on the Euclidean norm. Values of $d > 2$ could be considered specifically for smoothing weights, similar to our adjustment in the separate and combined penalties of taking an additional square root of w_1 which is equivalent to $d = 4$.

The other area of future work is to develop a better system for proposing smoothing parameter values. This is of particular importance for coordinate descent algorithms such as `gglasso` that rely on step sizes. In particular, the ideal range of smoothing parameters is likely to vary for each of the three penalties considered here. This can be inferred from a simple ordering of the penalties. Starting with the standard joint smooth-sparse penalty, the Triangle Inequality bounds the joint penalty above by

$$\sum_j \left(\lambda ||\boldsymbol{\beta}_j|| + \lambda \sqrt{\varphi} \sqrt{\boldsymbol{\beta}_j^T \boldsymbol{\Omega} \boldsymbol{\beta}_j} \right), \tag{3.37}$$

which resembles the fourth penalty proposed by Meier et al. [2009] that was not considered here. Note, though, that for given values of φ and λ, the separate penalty is strictly greater than (3.37) when $\lambda > \sqrt{\varphi}$. The combined penalty is also strictly greater than (3.37) for $\varphi > 0$. To account for this, and that the smoothing matrix $\boldsymbol{\Omega}$ often contains large elements, the values of φ considered should be smaller for the separate and combined penalties than for the joint penalty. Although we partially incorporated this idea by taking the square root of φ on the separate smoothing components, computational stability and efficiency may be further improved by adjusting the range of φ values, particularly for the combined penalty.

Chapter 4

Multi-Stage Functional Variable Selection with Latent Features

4.1 Introduction

As described in Chapter 1, a second possible explanation for the number of stages that SAFE requires to perform accurate selection is the redundancy of information collected across the sensors. Since the EMG sensors are surface electrodes, each sensor can detect the contractions of its intended muscle as well as neighboring muscles. Given the proximity of the EMG sensors, the same signal can be measured by multiple EMGs. Indeed, the clusters of correlated EMG signals in Figure 1.5 verify that this kind of cross-talk among the sensors is occurring. The underlying muscle contractions can, thus, be viewed as latent factors and the EMGs as surrogate measurements of those latent factors. For example, consider two latent signals $v_1(t)$ and $v_2(t)$ with expected response $E(y_i|v_1(t), v_2(t)) = \int_{\mathcal{T}} v_1(t)\beta_1(t)dt + \int_{\mathcal{T}} v_2(t)\beta_2(t)dt$, and suppose the observed signals can be written as linear combinations of the latent features as $X_j(t) = \alpha_{j1}v_1(t) + \alpha_{j2}v_2(t)$. Here, each latent signal represents the summed action potentials from the collection of active motor units coordinating a single muscle's contractions, and the corresponding functional coefficient describes the muscle's influence on hand velocity. EMG signals are known to be composed of motor unit action potentials (MUAPs) from the active motor units within the electrode recording range, and so each observed EMG signal, $X_j(t)$, measures the MUAPs at varying strengths, denoted by α_{j1} and α_{j2}.

As discussed in Chapter 2, penalized regression methods are known to experience difficulties in the presence of correlated predictors, such as predictors generated by latent factors. One method designed to recognize correlation clusters as latent features is principal components analysis (PCA). By design, PCA reduces the dimension of the predictor set by

estimating directions of maximal variation, where each PC is a linear combination of the original variables. A cluster of correlated predictors could then be represented by a single component. Although the number of estimable PCs is determined by the dimensions of the data, only the first few components are typically used since they explain the majority of the variation in the data. If the PCs are to be used in a regression model, though, the first few PCs need to be useful for prediction. In other words, the PCA needs to somehow be supervised by the response variable to guarantee the PCs have sufficient predictive power. Supervision does not inherently imply variable selection and can be conducted during construction of the PCs [Barshan et al., 2011]. However, the original proposal by Bair et al. [2006] performs an initial screening of the predictors that exhibit low correlation with the response. Correlation coefficients are calculated between each predictor and the response, and the predictors with absolute correlation values below a specified threshold are removed. Thus, the PCs are derived from only predictors that are sufficiently associated with the response. Bair et al. [2006] show that regression on the supervised PCs yields consistent coefficient estimates and predictions, whereas unsupervised PC regression does not.

Paul et al. [2008] extend Bair et al. [2006] to perform selection on the predictors after regression on the PCs. In particular, supervised PC regression yields the preconditioned response $\widehat{\boldsymbol{y}}$, on which the original predictors are regressed under some model fitting procedure like the lasso [Tibshirani, 1996]. Paul et al. [2008] show that using $\widehat{\boldsymbol{y}}$ instead of $\boldsymbol{y}$ in the lasso step improves selection under a latent variable setting by reducing the effects of noisy predictors. The goal of this chapter is to extend this feature extraction framework to functional data in order to achieve accurate EMG selection at a lower computational cost than SAFE, even for a large number of predictors. To accomplish this, each stage of the framework must be extended to the functional case.

In the supervision stage, the univariate regressions can be naturally extended to functional data using standard methods for scalar-on-function regression [Cardot et al., 2003; Ramsay & Silverman, 2005]. However, choosing the best measure of correlation and determining the correlation threshold may not be as straightforward. In particular, recent penalized regression literature has suggested penalizing the fit $||X_j(t)\gamma_j(t)||$ rather than the coefficient $||\gamma_j(t)||$ for $j = 1, ..., m$ [Fan et al., 2015; Simon & Tibshirani, 2012]. It may be similarly beneficial in this setting to measure correlations by the strength of each predictor's univariate fit. Related to this metric is the correlation between $\boldsymbol{y}$ and the predicted response $\widehat{\boldsymbol{y}}_j$ obtained from regression on the jth predictor. This work considers both quantities as plausible screening metrics.

Additionally, Bair et al. [2006] propose determining the threshold parameter from an arbitrary set of values through cross-validation. Not only will additional rounds of cross-

validation add to the computational complexity of the framework, but there is no guarantee that the set of possible values will contain a plausible option. The work herein proposes a flexible threshold generation procedure that simulates fake factors—variables with no constructed relation to the response—to empirically generate a distribution from which the threshold parameter can be selected. Fake factors, or 'pseudovariables,' have been used multiple times in the context of dimension reduction, with origins in factor analysis where the number of factors was determined as the number of eigenvalues of the sample correlation matrix that were at least as large as those provided by simulated data [Horn, 1965]. Miller [1990] suggest adding artificial variables to a regression model and terminating a model selection procedure when the first artificial variable enters the estimated model. Wu et al. [2007] determine the level of underfitting or overfitting using pseudovariables to tune a variable selection procedure. Alternatively, Luo et al. [2006] tune variable selection by adding noise to the response to obtain an unbiased estimate for error variance and to better evaluate the bias-variance trade-off. None of these methods, though, utilize fake factors in the exact manner that we propose. Although Horn [1965] use pseudovariables to derive a threshold, it is for determining the number of factors to retain whereas our threshold is for screening the original variables.

In the preconditioning stage, standard PCA must be exchanged for functional PCA (FPCA). Originally conceived by Grenander [1950], Karhunen [1946], Loève [1946], and Rao [1958], FPCA has been extended to various settings including densely observed functional data [Cardot, 2000; Castro et al., 1986; Pezzulli & Silverman, 1993; Rice & Silverman, 1991], sparse functional data [James et al., 2000; Paul & Peng, 2009; Rice & Wu, 2001; Shi et al., 1996; Staniswalis & Lee, 1998; Yao & Lee, 2006; Yao et al., 2005], and the presence of covariates [Cardot, 2007; Chiou & Müller, 2009; Chiou et al., 2003]. For the EMG application, the FPCA must accommodate multiple signals and may also impose sparsity since selection is the goal of the overall framework. Relevant advances have been made by Happ & Greven [2018], Nie et al. [2018], Paynabar et al. [2016], Wang & Tsung [2020], and Zhang et al. [2018]. Specifically, Nie et al. [2018] propose a smooth supervised FPCA that essentially combines the screening and preconditioning steps from Bair et al. [2006]. While their method can be extended to multiple functional predictors, it performs FPCA on each predictor individually without leveraging the correlation between the predictors. Paynabar et al. [2016] develop multivariate FPCA under the assumption that the multiple signals exhibit similar patterns and so share a common set of eigenfunctions. Zhang et al. [2018] follow a similar framework but impose sparsity on the scores so that each profile is represented by a combination of only some of the latent features. This allows for some clusters of strongly correlated predictors while other predictors are weakly correlated. Happ & Greven [2018] develop a more general

multivariate FPCA (MFPCA) that assumes different eigenfunctions for each predictor and also allows the time domains to vary across predictors. In the event that all of the predictors do exhibit similar patterns, MFPCA yields similar results to the methods of Paynabar et al. [2016]. Wang & Tsung [2020] propose a hierarchical method that vectorizes the multiple functional predictors and imposes sparsity on the loadings similar to Zou et al. [2006]. Sparse loadings allow for the interpretation that each latent feature can be represented by a sparse number of observed signals. They decompose the coefficient into three components to impose sparsity at a stage level, the profile level, and the time level.

To determine the optimal preconditioning method for the EMG data, I compare the performances of Sparse Multi-Channel FPCA [Zhang et al., 2018, SMFPCA], MFPCA [Happ & Greven, 2018], and a novel extension of Wang & Tsung [2020] called Sparse Loadings Multivariate FPCA (SLiM FPCA) that only considers profile-wise sparsity. The functional predictors are reconstructed from the principal components and regressed on the response to obtain the preconditioned response. In the final stage of the framework, the pre-screened functional predictors are regressed on the preconditioned response using SAFE, where smoothness and sparsity are adaptively imposed on the coefficients. This round of variable selection produces a final model that can then be estimated with standard functional linear regression techniques. As this chapter will demonstrate, this three-stage framework performs better than SAFE by itself, and reduces the overall computation time.

The remainder of this chapter proceeds as follows. Background information on PCA for multivariate data and multivariate functional data is presented in Section 4.2.1. The framework of Paul et al. [2008] is formally extended to the functional case in Section 4.3, with the novel SLiM FPCA developed therein. The extended framework is investigated through simulation in Section 4.4 and applied to the EMG data in Section 4.5. Final conclusions, observations, limitations, and possible extensions are discussed in Section 4.6

4.2 Background

The following section reviews principal components analysis for multivariate data, including estimation techniques and using principal components to predict a response variable. This review is followed by background information on functional principal components analysis for multivariate functional data.

4.2.1 Principal components analysis

The goal of PCA is to transform the observations on a set of correlated variables into uncorrelated observations along orthogonal axes in the directions that maximize the amount of variance explained. Explicitly, the first PC explains the largest amount of variance among the PCs using a linear combination of the original variables. The coefficients in this linear combination form a weight vector, say $\boldsymbol{v}_1$, such that

$$\boldsymbol{v}_1 = \underset{||\boldsymbol{v}||=1}{\arg\max}\{\boldsymbol{v}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{v}\} = \underset{\boldsymbol{v}}{\arg\max}\left\{\frac{\boldsymbol{v}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{v}}\right\}, \tag{4.1}$$

where the columns of $\boldsymbol{X}$ are assumed to be centered [Hastie et al., 2015]. For nonnegative definite matrices such as $\boldsymbol{X}^T\boldsymbol{X}$, the matrix's maximum eigenvalue λ_1 is the maximum possible value for the above expression, with $\boldsymbol{v}_1$ the corresponding eigenvector. Thus, the first principal component is $\{\boldsymbol{X}\boldsymbol{v}_1\}\boldsymbol{v}_1^T$. Remaining components are found using residuals, where $\boldsymbol{X}$ in the above expression is replaced with $\widehat{\boldsymbol{X}_k} = \boldsymbol{X} - \sum_{s=1}^{k-1}\boldsymbol{X}\boldsymbol{v}_s\boldsymbol{v}_s^T$ for the kth component. This process yields p orthogonal eigenvectors of $\boldsymbol{X}^T\boldsymbol{X}$ that are collected into the matrix $\boldsymbol{V} = \{\boldsymbol{v}_1, ..., \boldsymbol{v}_p\}$. The full PC decomposition of $\boldsymbol{X}$ can be given as $\boldsymbol{W} = \boldsymbol{X}\boldsymbol{V}$, where $\boldsymbol{w}_k = \boldsymbol{X}\boldsymbol{v}_k$ denotes the kth PC with loading $\sqrt{\lambda_k}\boldsymbol{v}_k$.

The PCs can also be calculated using a singular value decomposition (SVD) of $\boldsymbol{X}$, denoted $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$. The columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ are called the left and right singular vectors, respectively, and $\boldsymbol{D}$ is a diagonal matrix of the singular values of $\boldsymbol{X}$. To see the equivalence of these methods, write

$$\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\boldsymbol{D}^T\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T = \boldsymbol{V}\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{V}^T = \boldsymbol{V}\widehat{\boldsymbol{D}}^2\boldsymbol{V}^T, \tag{4.2}$$

where $\widehat{\boldsymbol{D}}$ is the square p-dimensional matrix of squared singular values of $\boldsymbol{X}$ with excess 0's removed and satisfying $\widehat{\boldsymbol{D}}^2 = \boldsymbol{D}^T\boldsymbol{D}$. Thus, the right singular vectors $\boldsymbol{V}$ of $\boldsymbol{X}$ are equivalent to the eigenvectors of $\boldsymbol{X}^T\boldsymbol{X}$, and the eigenvalues of $\boldsymbol{X}^T\boldsymbol{X}$ are the squared singular values of $\boldsymbol{X}$.

Alternatively, PCA can be framed as a type of ridge regression problem [Zou et al., 2006]. Let $\boldsymbol{A}_{p\times K} = [\alpha_1, ..., \alpha_K]$ and $\boldsymbol{B}_{p\times K} = [\boldsymbol{\beta}_1, .., .\boldsymbol{\beta}_K]$. For $\varphi > 0$ and the constraint $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}_{K\times K}$, let

$$(\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{B}}) = \underset{\boldsymbol{A},\boldsymbol{B}}{\arg\min}\sum_{i=1}^{N}||\boldsymbol{x}_i - \boldsymbol{A}\boldsymbol{B}^T\boldsymbol{x}_i||_2^2 + \varphi\sum_{k=1}^{K}||\boldsymbol{\beta}_k||_2^2. \tag{4.3}$$

Then $\widehat{\boldsymbol{\beta}}_k \propto \widehat{\boldsymbol{v}}_k$ for $k = 1, ..., K$. Under the additional restriction that $\boldsymbol{B} = \boldsymbol{A}$, the minimizer

of (4.3) is the first K loading vectors from usual PCA. Without this restriction, Zou et al. [2006] show that exact PCA is still obtained by adding the ridge penalty term.

Further, the ridge regression framework can be extended to induce sparsity in the loading vectors [Zou et al., 2006]. Specifically, adding the lasso penalty to (4.3) achieves the interpretation that the PCs are linear combinations of only a few of the original variables rather than all p of them. The optimization problem becomes

$$(\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{B}}) = \underset{\boldsymbol{A},\boldsymbol{B}}{\arg\min} \sum_{i=1}^{N} ||\boldsymbol{x}_i - \boldsymbol{A}\boldsymbol{B}^T\boldsymbol{x}_i||_2^2 + \varphi \sum_{k=1}^{K} ||\boldsymbol{\beta}_k||_2^2 + \sum_{k=1}^{K} \lambda_k ||\boldsymbol{\beta}_k||_1, \tag{4.4}$$

subject to $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$, where different sparsity parameters λ_k are allowed for penalizing the loadings of different PCs. The algorithm for solving (4.4) alternates between updating $\boldsymbol{A}$ and $\boldsymbol{B}$. First, initialize $\boldsymbol{A}$ as the first K columns of $\boldsymbol{V}$, i.e., the first K right singular vectors of $\boldsymbol{X}$. Given $\boldsymbol{A}$, let $\boldsymbol{Y}_k^* = \boldsymbol{X}\boldsymbol{\alpha}_k$ for each $k = 1, ..., K$. Then $\widehat{\boldsymbol{\beta}}_k$ is an elastic net estimate [Zou & Hastie, 2005],

$$\widehat{\boldsymbol{\beta}}_k = \underset{\boldsymbol{\beta}_k}{\arg\min} ||\boldsymbol{Y}_k^* - \boldsymbol{X}\boldsymbol{\beta}_k||_2^2 + \varphi||\boldsymbol{\beta}_k||_2^2 + \lambda_k||\boldsymbol{\beta}||_1. \tag{4.5}$$

Given $\boldsymbol{B}$, the penalties in (4.4) can be ignored and only $\sum_{i=1}^{N} ||\boldsymbol{x}_i - \boldsymbol{A}\boldsymbol{B}^T\boldsymbol{x}_i||_2^2 = ||\boldsymbol{X} - \boldsymbol{X}\boldsymbol{B}\boldsymbol{A}^T||_2^2$ needs to be minimized subject to $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$. Compute the SVD $(\boldsymbol{X}^T\boldsymbol{X})\boldsymbol{B} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, then $\widehat{\boldsymbol{A}} = \boldsymbol{U}\boldsymbol{V}^T$ minimizes the constrained problem by a reduced rank Procrustes rotation [Zou et al., 2006]. The algorithm for solving (4.3) is a simplified version of this.

Oftentimes, the components obtained from PCA are used to predict an outcome variable $\boldsymbol{y}$ through regression. With $\boldsymbol{y}$ centered and the first K PCs contained in $\boldsymbol{W}_K = \boldsymbol{X}\boldsymbol{V}_K$, the regression problem on $\boldsymbol{X}$ can be rewritten as

$$\underset{\boldsymbol{\beta}}{\arg\min} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||^2 = \underset{\boldsymbol{b}}{\arg\min} ||\boldsymbol{y} - \boldsymbol{W}_K\boldsymbol{b}||^2. \tag{4.6}$$

Thus, the solution of a principal component regression is $\widehat{\boldsymbol{b}} = (\boldsymbol{W}_K^T\boldsymbol{W}_K)^{-1}\boldsymbol{W}_K^T\boldsymbol{y}$, and $\widehat{\boldsymbol{y}} = \boldsymbol{W}_K\widehat{\boldsymbol{b}}$. Alternatively, the reconstructed model matrix $\widetilde{\boldsymbol{X}} = \boldsymbol{U}_K\boldsymbol{D}_K\boldsymbol{V}_K^T$ can be used to predict $\boldsymbol{y}$. Although this regression does not yield a unique estimate of $\boldsymbol{\beta}$, the estimate $\widehat{\boldsymbol{y}}$ of $\boldsymbol{y}$ is unique and equivalent to the predictions from the PC regression. A proof of this equivalence is given in Appendix C.

Principal component regression may not perform well because the PC that explains the most variation between the predictor variables is not guaranteed to be correlated with $\boldsymbol{y}$. In fact, the leading component may be completely unrelated to $\boldsymbol{y}$. This happens because the outcome variable does not influence the calculation of the PCs in the usual, unsupervised, PCA. Supervised PCA, on the other hand, only computes the PCs from the set of predictors

that are most correlated with the outcome [Bair et al., 2006]. Specifically, for an $N \times p$ matrix $\boldsymbol{X}$ with centered and scaled columns and centered response $\boldsymbol{y}$, let $\boldsymbol{s}$ denote the $p \times 1$ vector of correlation coefficients. That is, the jth entry of $\boldsymbol{s}$ is $s_j = \boldsymbol{x}_j^T \boldsymbol{y} / ||\boldsymbol{x}_j||_2$ for $j = 1, ..., p$, where $\boldsymbol{x}_j$ is the jth column of $\boldsymbol{X}$ and the sample standard deviation of $\boldsymbol{y}$ is omitted since it is common for all j. Define the set C_θ as the collection of indices such that $|s_j| > \theta$ for some $\theta > 0$ and let $\boldsymbol{X}_\theta$ be the submatrix of $\boldsymbol{X}$ corresponding to C_θ. The supervised principal components are then calculated by conducting PCA on $\boldsymbol{X}_\theta$. Bair et al. [2006] suggest selecting θ by cross-validation.

4.2.2 Multivariate functional principal components analysis

Functional PCA (FPCA) extends PCA to the case where observations are made along a continuum of an underlying function. Analysis of a single functional variable describes the primary sources of variation along the function's domain. FPCA is commonly used to observe a function's characterizing features and to gain a sense of the data complexity. See Ramsay & Silverman [2005] for full details. Of interest in this paper, however, is FPCA for $m > 1$ functional variables that takes into account both the features of a given function and those shared across functions. Multiple versions of multivariate FPCA have been studied [Berrendero et al., 2011; Chiou et al., 2014; Happ & Greven, 2018; Jacques & Preda, 2014; Ramsay & Silverman, 2005]. The multivariate FPCA (MFPCA) approach by Happ & Greven [2018], however, is the most general in that the data are not required to lie on the same interval nor have the same one-dimensional domains. This method is also conveniently available in the `MFPCA` package in `R`. Although we work with data observed on the same domain in this paper, we present the general MFPCA framework and refer the reader to the article for estimation details.

Consider the observed data set $\{X_{ij}(t_{jr}) : r = 1, .., n; j = 1, ..., m; i = 1, ..., N\}$, where $X_{ij}(t_{jr})$ is the ith sample of the jth random function $X_j(\cdot)$ at the rth point in the closed, compact domain $\mathcal{T}_j$. Denote the m random functions succinctly as the random vector function $\boldsymbol{X}(\boldsymbol{t}_r) = (X_1(t_{1r}), ..., X_m(t_{mr}))^T$ with $\boldsymbol{t}_r = (t_{1r}, ..., t_{mr})^T \in \mathcal{T} = \mathcal{T}_1 \times \cdots \times \mathcal{T}_m$, where each function is assumed to be square-integrable, $X_j(\cdot) \in \mathcal{L}_2(\mathcal{T}_j)$, so that $\boldsymbol{X}(\boldsymbol{t})$ lies in the Hilbert space $\mathbb{H} = \mathcal{L}_2(\mathcal{T}_1) \times \cdots \times \mathcal{L}_2(\mathcal{T}_m)$. Define the mean vector function as $\boldsymbol{\mu}(\boldsymbol{t}) = E[\boldsymbol{X}(\boldsymbol{t})] = (E[X_1(t_1)], ..., E[X_m(t_m)])^T$ and the covariance matrix as $\boldsymbol{\Gamma}(\boldsymbol{t}, \boldsymbol{t}') = \{\Gamma_{jl}(t_j, t_l)\}$ for $\boldsymbol{t}, \boldsymbol{t}' \in \mathcal{T}$ with elements $\Gamma_{jl}(t_j, t_l') = \text{Cov}(X_j(t_j), X_l(t_l'))$ for $t_j \in \mathcal{T}_j$ and $t_l' \in \mathcal{T}_l$.

The Hilbert space $\mathbb{H}$ is defined by the inner product,

$$\langle \boldsymbol{f}, \boldsymbol{g} \rangle_{\mathbb{H}} = \sum_{j=1}^{m} \langle f_j, g_j \rangle = \sum_{j=1}^{m} \int_{\mathcal{T}_j} f_j(t_j) g_j(t_j) dt_j \tag{4.7}$$

for $\boldsymbol{f}, \boldsymbol{g} \in \mathbb{H}$, $\boldsymbol{f} = (f_1, ..., f_m)^T$, with corresponding norm $||\boldsymbol{f}||_{\mathbb{H}} = \langle \boldsymbol{f}, \boldsymbol{f} \rangle_{\mathbb{H}}^{1/2}$.With the covariance operator $\mathcal{G} : \mathbb{H} \to \mathbb{H}$ defined as

$$(\mathcal{G}\boldsymbol{f})(\boldsymbol{t}) = \int \boldsymbol{\Gamma}(\boldsymbol{t}, \boldsymbol{t}')\boldsymbol{f}(\boldsymbol{t}')d\boldsymbol{t}' = \begin{pmatrix} \langle \boldsymbol{\Gamma}_1(t_1, \cdot), \boldsymbol{f} \rangle_{\mathbb{H}} \\ \vdots \\ \langle \boldsymbol{\Gamma}_m(t_m, \cdot), \boldsymbol{f} \rangle_{\mathbb{H}} \end{pmatrix}, \tag{4.8}$$

for $\boldsymbol{\Gamma}_j(t_j, \cdot) = (\Gamma_{j1}(t_j, \cdot), ..., \Gamma_{jm}(t_j, \cdot))^T$ and $t_j \in \mathcal{T}_j$, and assumed to be a compact positive operator, the Hilbert-Schmidt Theorem [Reed & Simon, 1980] proves the existence of a complete orthonormal basis of eigenfunctions $\boldsymbol{\psi}_k(\boldsymbol{t}) = (\psi_{k1}(t_1), ..., \psi_{km}(t_m))^T \in \mathbb{H}$, $k = 1, 2, ...,$ of $\mathcal{G}$ such that $\mathcal{G}\boldsymbol{\psi_k} = \rho_k \boldsymbol{\psi}_k$ and $\rho_k \to 0$ as $k \to \infty$. By the Spectral Theorem we have the decomposition $\mathcal{G}\boldsymbol{f} = \sum_{k=1}^{\infty} \rho_k \langle \boldsymbol{f}, \boldsymbol{\psi}_k \rangle_{\mathbb{H}} \boldsymbol{\psi}_k$ for all $\boldsymbol{f} \in \mathbb{H}$, and a multivariate version of Mercer's Theorem shows with absolute and uniform convergence that

$$\text{Cov}(X_j(t_j), X_l(t_l')) = \Gamma_{jl}(t_j, t_l') = \sum_{k=1}^{\infty} \rho_k \psi_{kj}(t_j)\psi_{kl}(t_l'). \tag{4.9}$$

Finally, the Multivariate Karhunen-Loève Theorem gives the representation $\boldsymbol{X}(\boldsymbol{t}) = \sum_{k=1}^{\infty} \xi_k \boldsymbol{\psi}_k(\boldsymbol{t})$ for $\boldsymbol{t} \in \mathcal{T}$ and random variables $\xi_k = \langle \boldsymbol{X}(\boldsymbol{t}), \boldsymbol{\psi}_k(\boldsymbol{t}) \rangle_{\mathbb{H}}$ with mean zero and $\text{cov}(\xi_k, \xi_{k'}) = \rho_k \delta_{kk'}$ where $\delta_{kk'} = 1$ if $k = k'$ and zero otherwise. The leading eigenfunctions portray the most important features of $\boldsymbol{X}(\boldsymbol{t})$, so we truncate the Karhuen-Loève expansion at K dimensions in practice as $\boldsymbol{X}(\boldsymbol{t}) \approx \sum_{k=1}^{K} \xi_k \boldsymbol{\psi}_k(\boldsymbol{t})$.

Under this representation, each function $X_j(\cdot)$ is assumed to be generated by a distinct set of eigenfunctions $\{\psi_{kj}\}_k$ and all functions are related to their respective first eigenfunction ψ_{1j} by the common value ξ_k. In latent factor settings such as the EMG application, it may be more practical to assume that a common set of eigenfunctions $\{\psi_k\}_k$ generate all of the random functions observed and allow each $X_j(\cdot)$ to depend on ψ_k through different coefficients, ξ_{kj}. This representation is explored by Paynabar et al. [2016] and is appropriate when all observed functions are highly correlated with each other. Recall that this is the model we suppose for the EMG signals, but it may not be the most appropriate. That is, each EMG sensor in Figure 1.2 intends to measure a single muscle with known influence over a particular type of hand movement. Given the locations of EMGs 3 and 6, it is reasonable to assume that they fail to capture the same information as, say, EMGs 5, 7, and 12. Indeed, Figure 1.5 shows that EMG 6 is uncorrelated with all other signals and EMG 3 is only moderately correlated with other signals. Thus, we may prefer a model that allows for clusters of correlated signals, with little to no correlation between the clusters.

Zhang et al. [2018] propose such a model and estimation procedure, known as sparse

multi-channel FPCA (SMFPCA), that imposes sparsity on the scores to all for each function to be represented by a sparse combination of the extracted features. As before, assume there are N samples of a multivariate functional process, $\boldsymbol{X}(t) = [X_1(t), ..., X_m(t)]^T$, but with all functions measured on the same domain $\mathcal{T}$. With the assumption of continuity over $\mathcal{T}$, the covariance operator $\Gamma f(t) = \int f(t')\Gamma(t, t')dt'$ has orthogonal eigenfunctions $\psi_k(t)$, $k = 1, 2, ...$ with non-increasing eigenvalues satisfying $\Gamma\psi_k = \rho_k\psi_k$. Then the Karhunen-Loève expansion yields the representation

$$\boldsymbol{X}_i(t) = \boldsymbol{\mu}(t) + \sum_{k=1}^{\infty} \psi_k(t)\boldsymbol{\xi}_{ik}, \tag{4.10}$$

for $i = 1, ..., N$, where $\boldsymbol{\xi}_{ik} = [\xi_{ik1}, ..., \xi_{ikm}]^T$ is an m-dimensional random vector with mean zero and jth element denoted $\xi_{ikj} = \int\{X_{ij}(t) - \mu_j(t)\}\psi_k(t)dt$. Again, the expansion is often truncated to yield an approximation of $\boldsymbol{X}_i(t)$ based on $K < \infty$ features. To induce sparsity in $\boldsymbol{\xi}_{ik}$, the FPCA is framed as a minimization problem of the reconstruction error with an added Lasso penalty on the score vectors,

$$\underset{\boldsymbol{\Xi},\boldsymbol{\Psi}}{\arg\min} \sum_{i=1}^{N} ||\boldsymbol{X}_i - \boldsymbol{\mu} - \boldsymbol{\Psi}\boldsymbol{\Xi}_i^T||_F^2 + \lambda \sum_{k=1}^{K}\sum_{i=1}^{N} ||\boldsymbol{\xi}_{ik}||_1 \text{ subject to } \boldsymbol{\Psi}^T\boldsymbol{\Psi} = \boldsymbol{I}_{K\times K}, \tag{4.11}$$

where $\boldsymbol{X}_i$, $\boldsymbol{\mu} \in \mathbb{R}^{n\times m}$ with rth rows denoted by $\boldsymbol{X}_i(t_r)$ and $\boldsymbol{\mu}(t_r)$, respectively, $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_K] \in \mathbb{R}^{n\times K}$ with $\boldsymbol{\psi}_k = [\psi_k(t_1), ..., \psi_k(t_n)]^T$ for $k = 1, ..., K$, and $\boldsymbol{\Xi}_i = [\boldsymbol{\xi}_{i1}, ..., \boldsymbol{\xi}_{iK}] \in \mathbb{R}^{m\times K}$ for $i = 1, ..., N$. The level of sparsity is controlled by the tuning parameter $\lambda > 0$ and $||\cdot||_F^2$ denotes the square of the Frobenius norm of a matrix. A Block Coordinate Descent algorithm is used to iteratively update $\boldsymbol{\Psi}$ and $\boldsymbol{\Xi}_i$ $(i = 1, ..., N)$ until convergence, with closed form expressions for each given in Section 2.2 of Zhang et al. [2018].

4.3 Methodology

This section proposes an extension of the supervised preconditioning for feature selection and regression by Paul et al. [2008] to the case of functional data. This three step approach conducts initial screening of functional predictors, multivariate functional principal component analysis and regression to obtain a preconditioned response, and a final sparse functional regression using the preconditioned response and selected predictors. Specifically, the first stage provides supervision for the multivariate FPCA by screening predictors with univariate predictions that are weakly correlated with the response variable, an extension of Bair et al. [2006]. The second stage utilizes one of three multivariate FPCA methods to estimate latent features, which are then used to estimate the response. The three FPCA methods

are MFPCA [Happ & Greven, 2018], SMFPCA [Zhang et al., 2018], and a novel method that simplifies the work of Wang & Tsung [2020], where the latter two impose sparsity on the scores and loadings, respectively. Finally, the preconditioned response is regressed on the final set of functional predictors, taking into account the screening and induced sparsity from the previous two stages.

A common assumption for PCA is that the data is pre-centered. Centering is also standard in many regression problems, including those involving sparsity penalties, for simple presentation of the data and to set the intercept coefficient to zero. As such, both $\boldsymbol{y}$ and $\boldsymbol{X}$ are assumed to be pre-centered, where $\boldsymbol{X}$ is centered for each signal at each time point. Another common assumption is that the data are standardized to be on the same scale. In applications such as those considered by Wang & Tsung [2020] where the profiles are measuring different physical quantities, the data need to be standardized so that the profiles are on the same scale. However, in applications where a latent factor scenario generates the data, such as forearm muscle contractions measured by EMGs, the signals are already on the same scale and standardization would remove the variation unique to each signal and important for accurate variable selection. For the purpose of screening, though, all variables are scaled so that marginal correlations with the original response are evaluated.

4.3.1 Supervision with Fake Factor Thresholding

Recall the description of supervised PCA from Section 4.2.1 [Bair et al., 2006]. This method calculates the univariate correlation coefficients between each predictor and $\boldsymbol{y}$ and screens predictors with sufficiently small coefficient magnitudes. The authors focus their discussion of the method on latent factor scenarios, so it is reasonable that a similar screening procedure would perform well in the EMG application. Thus, I propose a simple extension for functional predictors. Specifically, perform smoothed univariate functional linear regression of the centered scalar response on each centered and scaled functional predictor using the following criterion function,

$$\sum_{i=1}^{N}\left(y_i - \int_{\mathcal{T}} X_{ij}(t)\gamma_j(t)dt\right)^2 + \varphi\left[\int_{\mathcal{T}}\{\gamma_j''(t)\}^2 dt\right]^{1/2}. \tag{4.12}$$

Similar to the regression problems in Chapter 3, we solve the above regression by approximating $\gamma_j(t)$ with a linear combination of orthogonal cubic B-spline basis functions and solving the resulting ridge regression problem. The smoothing parameter φ is chosen via V-fold CV as the value that minimizes CV error. Then, one of two values can be calculated, either $||X_j(t)\gamma_j(t)||$ or $\text{Corr}(\boldsymbol{y}, \widehat{\boldsymbol{y}}_j)$, where $\widehat{\boldsymbol{y}}_j$ is the predicted response from regression on

$X_j(t)$. Denote the chosen metric as s_j, and only keep predictors with sufficiently large values in the set $C_\theta = \{j : s_j > \theta\}$.

For selecting the threshold θ, Paul et al. [2008] suggest using cross-validation yet do not provide rationale for proposing a set of possible values. Instead, I propose a data-driven approach that simulates a distribution for θ based on user-generated curves that are unrelated to the response. An appropriate summary statistic is then selected as an estimate for θ to obtain a 'Fake Factor Threshold.' In more detail, this Fake Factor Thresholding proceeds as follows. First, generate N observations of Q centered functional predictors. Regress the centered response on each fake predictor F_q to obtain coefficient estimates $\widehat{\gamma}_q$, $q = 1, ..., Q$. Calculate the standardized norm of each estimated fit s_q to form an empirical distribution for θ and select some statistic $\widehat{\theta}^{(d)}$ from this distribution, say the first, second, or third quartile. Repeat this process D times and obtain a final estimate $\widehat{\theta}$ as the mean of the $\widehat{\theta}^{(d)}$'s. Associated standard errors can also be obtained. This approach is demonstrated in Sections 4.4 and 4.5 with full details regarding the fake factor generation given in Section 4.4.1.

4.3.2 Feature Selection for Preconditioning

The second stage of the analysis extracts the latent features from the pre-screened set of predictors, i.e., $\{X_j(\cdot) : j \in C_\theta\}$, and regresses the response on these features to obtain a preconditioned response. The particular type of FPCA to be used will depend on the application of interest. In our context, we compare MFPCA [Happ & Greven, 2018] and SMFPCA [Zhang et al., 2018], as described in Section 4.2.2, to a novel method called Sparse Loadings Multivariate FPCA (SLiM FPCA). The following sections detail SLiM FPCA and describe preconditioning the response with the reconstructed signals.

In standard PCA of an $N \times p$ matrix $\boldsymbol{X}$, there are a total of p possible PCs. However, a common goal of PCA is to reduce the dimensionality of the data, and so fewer than p components are typically retained. A general rule for determining the number K of PCs to keep is by using the proportion of variance explained. Specifically, the ratio of the sum of the first K eigenvalues to the sum of all p eigenvalues of $\boldsymbol{X}^T\boldsymbol{X}$ yields the proportion of variance explained by the first K components, $\sum_{k=1}^{K} \rho_k / \sum_{k=1}^{p} \rho_k$. The value K is chosen so that the proportion is at least as large as some threshold, say 0.8. For the functional setting, we use the MFPCA algorithm to determine the first 10 eigenfunctions and corresponding eigenvalues, without sparsity, and keep the first K components such that 80% of variance is explained. This value K then informs the number of components calculated by SMFPCA and SLiM FPCA. The 80% explained variance threshold is notably lower than typically used in standard PCA due to the increased amount of noise present in functional data.

4.3.2.1 Sparse Loadings Multivariate Functional Principal Components Analysis

As previously noted, existing multivariate FPCA methods either do not perform selection or they impose sparsity on the FPC scores. The latter provides the interpretation that the observed signals are sparse combinations of the latent features. For the EMG application, however, we desire the FPCs to be sparse combinations of the observed signals. Given that muscle locations can change during amputation, this interpretation is conducive for understanding the latent relationships between signals. As will be detailed in this section, we achieve this interpretation by imposing group-wise sparsity on the loading vector components that correspond to each signal.

First, though, recall the standard perspective of PCA is to search for a linear subspace that maximizes the variance of the projected data. Alternatively, PCA can be viewed as a sequential minimization of reconstruction errors [Bishop, 2006; Shen & Huang, 2008]. To build this in the functional setting, denote the M observed signals in the ith sample by the function vector $\boldsymbol{x}_i(t) = (x_{i1}(t), ..., x_{im}(t))^T$. The eigenfunction vector corresponding to the first FPC is denoted as $\boldsymbol{v}(t) = (v_1(t), ..., v_m(t))^T$ and can be obtained as the solution to

$$\min_{\boldsymbol{v}(t)} \sum_{i=1}^{N} \int_{\mathcal{T}} ||\boldsymbol{x}_i(t) - \boldsymbol{v}(t)c_i||^2 dt \quad \text{s.t. } \langle \boldsymbol{v}(t) \rangle_{\mathbb{H}} = 1, \tag{4.13}$$

where $c_i = \langle \boldsymbol{x}_i(t), \boldsymbol{v}(t) \rangle_{\mathbb{H}} = \sum_{j=1}^{m} \int_{\mathcal{T}} x_{ij}(t) v_j(t) dt$ is the PC score of the ith sample. Since signals are not continuously observed in practice, but rather at a discrete grid of time points, the signal vector can be vectorized as $\boldsymbol{x}_{ij} = (x_{ij1}, ..., x_{ijn})^T$ where the number of time points is assumed to be the same for all signals. Then denote the ith sample as the vector $\boldsymbol{x}_i = (\boldsymbol{x}_{i1}^T, ..., \boldsymbol{x}_{im}^T)^T$ of dimension $p = m \cdot n$. The MFPCA in (4.13) is simplified to VPCA [Fang et al., 2017; Grasso et al., 2014] by replacing the integration with summation, where the eigenvector $\boldsymbol{v} = (v_{11}, ..., v_{1n}, ..., v_{m1}, ..., v_{mn})^T$ for the first PC is the solution to

$$\min_{\boldsymbol{v}} \sum_{i=1}^{N} ||\boldsymbol{x}_i - \boldsymbol{v}\boldsymbol{v}^T\boldsymbol{x}_i||^2 \quad \text{s.t. } \boldsymbol{v}^T\boldsymbol{v} = 1. \tag{4.14}$$

Eigenvectors for remaining PCs can be obtained by iteratively solving (4.14) with the update $\boldsymbol{x}_i = \boldsymbol{x}_i - \boldsymbol{v}\boldsymbol{v}^T\boldsymbol{x}_i$ at each iteration.

Since the length of $\boldsymbol{v}$ is often greater than N for functional data and $\boldsymbol{v}$ is not interpretable when all entries are non-zero, a form of regularization is desired. As described in Section 4.2.1, Zou et al. [2006] developed Sparse PCA to address these issues for usual multivariate data. Wang & Tsung [2020] extended their work to the functional case, which is presented here

for completeness. First, define $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_N)^T$ as the data matrix and note that (4.14) is then equivalent to the following ridge regression-type problem,

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} ||\boldsymbol{X} - \boldsymbol{X}\boldsymbol{\beta}\boldsymbol{\alpha}^T||_F^2 + \varphi||\boldsymbol{\beta}||^2 \quad \text{s.t. } \boldsymbol{\alpha}^T\boldsymbol{\alpha} = 1, \tag{4.15}$$

where $||\cdot||_F$ is the Frobenius norm and $\varphi \geq 0$ is the regularization parameter. The first term $||\boldsymbol{X} - \boldsymbol{X}\boldsymbol{\beta}\boldsymbol{\alpha}^T||_F^2 = \sum_{i=1}^N ||\boldsymbol{x}_i - \boldsymbol{\alpha}\boldsymbol{\beta}^T\boldsymbol{x}_i||^2$ represents the reconstruction errors and, as noted by Zou et al. [2006], the second term $\varphi||\boldsymbol{\beta}||^2$ with $\varphi > 0$ is necessary when $p > N$ in order to obtain a unique solution. Hence, the purpose of the second term is to enforce a unique reconstruction rather than to penalize or shrink the regression coefficients as in usual ridge regression.

Theorem 1 in Wang & Tsung [2020] demonstrates this explicitly and provides the following two results:

(a) $\widehat{\boldsymbol{\beta}} = d_1^2/(d_1^2 + \varphi)\boldsymbol{v}$, i.e., $\widehat{\boldsymbol{\beta}} \propto \boldsymbol{v}$, where d_1 is the first singular value of $\boldsymbol{X}$.

(b) As $\varphi \to \infty$, (4.15) is reduced to the problem as below:

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} ||\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{\beta}||^2 - ||\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha}||^2 \quad \text{s.t. } \boldsymbol{\alpha}^T\boldsymbol{\alpha} = 1. \tag{4.16}$$

Using the model in (b), sparsity can be induced by adding a penalty function for $\boldsymbol{\beta}$ to obtain

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} ||\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{\beta}||^2 - ||\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha}||^2 + \lambda h(\boldsymbol{\beta}) \quad \text{s.t. } \boldsymbol{\alpha}^T\boldsymbol{\alpha} = 1. \tag{4.17}$$

Previous works have used the Lasso penalty $h(\boldsymbol{\beta}) = \sum_{j=1}^p |\beta_j|$ from Tibshirani [1996] to induce element-wise sparsity [Shen & Huang, 2008; Wang & Tsung, 2020; Zou et al., 2006]. Specifically, Wang & Tsung [2020] decompose the coefficient into a product of three coefficients, each controlling sparsity at one of three levels: stage of the process, observed profiles within each stage, and time points within each profile. For our application, we propose using the group Lasso penalty $h(\boldsymbol{\beta}) = \sum_{j=1}^m ||\boldsymbol{\beta}_j||$ of Yuan & Lin [2006] to impose only profile-wise sparsity in $\boldsymbol{\beta}$. The proposed method, Sparse Loadings Multivariate FPCA (SLiM FPCA), is formally written as

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} ||\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{\beta}||^2 - ||\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha}||^2 + \lambda \sum_{j=1}^m ||\boldsymbol{\beta}_j|| \quad \text{s.t. } \boldsymbol{\alpha}^T\boldsymbol{\alpha} = 1. \tag{4.18}$$

SLiM FPCA is an extended simplification of the Hierarchical Sparse MFPCA (HSMFPCA) of Wang & Tsung [2020]. The simplification lies in not decomposing $\boldsymbol{\beta}$ to obtain

sparsity at different levels; we are only concerned about sparsity at the signal level. The extension is a result of this simplification. That is, since we are not concerned with sparsity across time for a given signal, all time points for a given signal form a group that corresponds to the jth segment of $\boldsymbol{\beta}$, $\boldsymbol{\beta_j}$. This requires the use of the group Lasso penalty over the Lasso penalty. Note that Wang & Tsung [2020] compare HSMFPCA to a heuristic sequential approach that employs the group lasso to perform stage-wise and profile-wise selection followed by the lasso to screen across time. However, to our knowledge there is no formalized method for achieving only profile-wise sparsity.

Although SLiM FPCA differs from HSMFPCA in construction, they share the property of closed-form update equations since there is no multiplying matrix in front of $\boldsymbol{\beta}$ in $||\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{\beta}||^2$. Thus, SLiM FPCA enjoys model interpretability and computational efficiency similar to HSMFPCA. In particular, the SLiM FPCA is a bi-convex optimization problem that can be solved with a Block Coordinate Descent algorithm in which $\boldsymbol{\beta}$ is estimated for a fixed $\boldsymbol{\alpha}$, then $\boldsymbol{\alpha}$ for a fixed $\boldsymbol{\beta}$. The update equation for $\boldsymbol{\alpha}$ is given in Proposition 2 of Wang & Tsung [2020] as $\widehat{\boldsymbol{\alpha}} = \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}/||\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}||$, the proof of which we provide in Appendix C. The update equation for $\boldsymbol{\beta}$ is given in the following proposition. These two parameters are iteratively updated in the algorithm until a convergence condition is met.

Proposition 4.3.1. In (4.18), when $\boldsymbol{\alpha}$ is fixed, $\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha}$ is known and $\widehat{\boldsymbol{\beta}}_j$ can be obtained for $j = 1, ..., m$ as

$$\widehat{\boldsymbol{\beta}}_j = \left(1 - \frac{\lambda}{2||\boldsymbol{w}_j||}\right)_+ \boldsymbol{w}_j, \tag{4.19}$$

where $\boldsymbol{w}_j = \boldsymbol{X}_j^T\boldsymbol{X}\boldsymbol{\alpha}$.

Proof. For a fixed $\boldsymbol{\alpha}$, we set $\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha}$ and optimize

$$||\boldsymbol{w} - \boldsymbol{\beta}||^2 + \lambda \sum_{j=1}^{m} ||\boldsymbol{\beta}_j||,$$

where the squared norm of $\boldsymbol{w}$ is dropped because it does not depend on $\boldsymbol{\beta}$. Differentiation of the optimization criterion yields the KKT conditions below:

$$\frac{\delta}{\delta\boldsymbol{\beta}_j} = -2(\boldsymbol{w}_j - \boldsymbol{\beta}_j) + \lambda\delta_j, \quad \delta_j = \begin{cases} \boldsymbol{\beta}_j/||\boldsymbol{\beta}_j|| & \text{if } \boldsymbol{\beta}_j \neq 0 \\ \{\boldsymbol{u} : ||\boldsymbol{u}|| \leq 1\} & \text{if } \boldsymbol{\beta}_j = 0. \end{cases}$$

Working with the first condition where $\boldsymbol{\beta}_j \neq 0$, we have

$$0 = -2(\boldsymbol{w}_j - \boldsymbol{\beta}_j) + \lambda \frac{\boldsymbol{\beta}_j}{||\boldsymbol{\beta}_j||}, \tag{4.20}$$

$$2\boldsymbol{w_j} = (2 + \lambda/||\boldsymbol{\beta}_j||)\boldsymbol{\beta}_j, \tag{4.21}$$

$$2||\boldsymbol{w}_j|| = (2 + \lambda/||\boldsymbol{\beta}_j||)||\boldsymbol{\beta}_j||, \tag{4.22}$$

$$||\boldsymbol{\beta}_j|| = ||\boldsymbol{w}_j|| - \lambda/2. \tag{4.23}$$

Substitution of $||\boldsymbol{\beta}_j||$ into (4.21) yields the update equation, $\widehat{\boldsymbol{\beta}}_j = \left(1 - \frac{\lambda}{2||\boldsymbol{w}_j||}\right) \boldsymbol{w}_j$. Now in the second KKT condition where $\boldsymbol{\beta}_j = 0$, similar arithmetic shows

$$0 = -2\boldsymbol{w}_j + \lambda \boldsymbol{u}_j \quad \Rightarrow \quad \left(1 - \frac{\lambda}{2||\boldsymbol{w}_j||}\right) \leq 0. \tag{4.24}$$

Combining the two conditions yields the full update equation. □

The algorithm for SLiM FPCA is detailed in Algorithm 1. The parameter $\boldsymbol{\alpha}$ is initialized as the first eigenvector of $\boldsymbol{X}^T\boldsymbol{X}$. Zou et al. [2006] calculate the eigenvector directly via singular value decomposition (SVD). However, when $\boldsymbol{X}$ is a particularly wide matrix, i.e., $p >> N$, this can require extensive computation. Instead, Bishop [2006] showed that the first eigenvector of $\boldsymbol{X}^T\boldsymbol{X}$ is equivalent to $\boldsymbol{X}^T\boldsymbol{e}/||\boldsymbol{X}^T\boldsymbol{e}||$, where $\boldsymbol{e}$ is the first eigenvector of $\boldsymbol{X}\boldsymbol{X}^T$. Since functional data usually have $p >> N$, this initiation of α is used. Further, the regularization parameter λ is selected through V-fold CV with the SE Rule described in Section 2.3.3, where the APE is calculated from reconstruction errors.

Algorithm 1 SLiM FPCA Algorithm

1: Initialize $\widehat{\boldsymbol{\alpha}}_0 = \boldsymbol{X}^T\boldsymbol{e}/||\boldsymbol{X}^T\boldsymbol{e}||$, where $\boldsymbol{e}$ is first eigenvector of $\boldsymbol{X}\boldsymbol{X}^T$
2: Let $\boldsymbol{y} = \boldsymbol{X}^T\boldsymbol{X}\widehat{\boldsymbol{\alpha}}$. Update $\widehat{\boldsymbol{\beta}}_j = \left(1 - \frac{\lambda}{2||\boldsymbol{y}_j||}\right)_+ \boldsymbol{y}_j$
3: Update $\widehat{\boldsymbol{\alpha}} = \boldsymbol{X}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}}/||\boldsymbol{X}^T\boldsymbol{X}\widehat{\boldsymbol{\beta}}||$
4: Iterate 2-3 until convergence ($||f_m - f_{m-1}|| < \epsilon$)
5: Normalize $\widehat{\boldsymbol{v}} = \widehat{\boldsymbol{\beta}}/||\widehat{\boldsymbol{\beta}}||$. Update $\boldsymbol{X} = \boldsymbol{X} - \boldsymbol{X}\widehat{\boldsymbol{v}}\widehat{\boldsymbol{v}}^T$.
6: Cycle 1-5 for K PCs

Remark 4.3.1. For sufficiently large values of λ, it is possible for the entire vector $\boldsymbol{\beta}$ to be estimated as 0 for a particular PC. When this happens, $\boldsymbol{\alpha}$ cannot be updated nor can $\widehat{\boldsymbol{v}}$ exist given their formulae. Thus, the PC is deemed not estimable and the algorithm is forced to

exit for that value of λ. To determine a feasible range of λ values, a maximum value of λ can be approximated by exploiting the second KKT condition from the proof of Proposition 4.3.1 which states $2||\boldsymbol{w}_j|| \leq \lambda$. Using the initial value of $\boldsymbol{\alpha}$ and substituting for $\boldsymbol{w}_j$, we have $||\boldsymbol{w}_j|| = \nu||\boldsymbol{X}_j^T\boldsymbol{e}||/||\boldsymbol{X}^T\boldsymbol{e}||$ where ν and $\boldsymbol{e}$ are the first eigenvalue and eigenvector of $\boldsymbol{X}\boldsymbol{X}^T$, respectively. Hence, the maximum value of λ can be set as $\lambda_{\max} = 2\nu||\boldsymbol{X}_j^T\boldsymbol{e}||/||\boldsymbol{X}^T\boldsymbol{e}||$.

4.3.2.2 Preconditioning

For multivariate data in Paul et al. [2008], after conducting PCA the response is regressed on the first K principal components. If we directly extend this to the functional case with a standard linear model, we lose the smoothing of coefficients. In the case of a signle functional predictor, one could adopt the methods of Reiss & Ogden [2007] and conduct PCA on the signals after a basis expansion. This could work well for SLiM FPCA under a univariate functional coefficient model, but computation time is greatly increased under a bivariate coefficient model because the basis expansion squares the group size. However, methods like MFPCA and SMFPCA are meant for operation on the densely-observed raw curves or sparsely-observed curves that have been approximated and interpolated on a dense grid of points. Other key differences between the three methods are summarized in Table 4.1.

Characteristic	MFPCA	SMFPCA	SLiM FPCA
Eigenfunctions	Different sets for each $X_j(\cdot)$	Common set across $X_j(\cdot)$'s	Different sets for each $X_j(\cdot)$
Scores	Common across $X_j(\cdot)$'s for the ith sample	Different for each $X_j(\cdot)$	Common across $X_j(\cdot)$'s for the ith sample
Sparsity	No sparsity	Sparsity in score vectors	Group-wise sparsity in eigenfunction vectors
Interpretation	Each $X_j(\cdot)$ is represented by a combination of eigenfunctions	Eigenfunctions represented by sparse combinations of $X_j(\cdot)$'s	$X_j(\cdot)$'s represented by sparse combination of eigenfunctions

Table 4.1 Key differences between the three preconditioning methods considered.

Alternatively, note that in usual PCA for multivariate data, the predicted response is equivalent whether the response is regressed on the PCs or on the reconstructed predictors.

A proof of this is provided in Appendix C, where even though the coefficient is not unique for regression on the reconstructed predictors, the predicted response is unique and is equivalent to regressing on the PCs. Thus, we suggest reconstructing the functional predictors from the estimated eigenfunctions and scores and regressing the response on them with a smoothing penalty to obtain the preconditioned response. As in the supervision stage, the smoothing parameter is chosen to minimize CV error. For completeness, we now describe the reconstruction and modeling as applicable for SLiM FPCA and MFPCA, and provide the reconstruction for SMFPCA in Appendix C.

Denote the eigenfunctions and scores corresponding to the kth FPC as $v^k = (v_1^k, ..., v_m^k)$ and $\boldsymbol{c}^k = (c_1^k, ..., c_N^k)^T$, respectively, for $k = 1, ..., K$. Then the ith observation of the jth predictor at time t has the truncated Karhunen-Loève expansion $X_{ij}(t) \approx \sum_{k=1}^K c_i^k v_j^k(t)$. Now, under a univariate coefficient model, the coefficient function is approximated by a linear combination of orthogonal basis functions $\{\omega_\ell\}_{\ell=1}^L$ as $\gamma_j(t) \approx \sum_{\ell=1}^L \beta_{j\ell}\omega_\ell(t)$. With the reconstruction of X_{ij} and linear approximation of γ_j, the functional linear model can be written as

$$\sum_{j=1}^m \int X_{ij}(t)\gamma_j(t) \approx \sum_{j=1}^m \int \left(\sum_{k=1}^K c_i^k v_j^k(t)\right)\left(\sum_{\ell=1}^L \beta_{j\ell}\omega_\ell(t)\right) dt \tag{4.25}$$

$$= \sum_j \sum_k \sum_\ell c_i^k \beta_{j\ell} \left(\int v_j^k(t)\omega_\ell(t)dt\right). \tag{4.26}$$

A Riemann sum approximates the integral as $\int v_j^k(t)\omega_\ell(t)dt \approx \Delta_t \boldsymbol{v}_{jk}^T \boldsymbol{\omega}_\ell$ with Δ_t denoting the distance between two consecutive time points, $\boldsymbol{v}_{jk}^T = (v_j^k(t_1), ..., v_j^k(t_n))$, and $\boldsymbol{\omega}_\ell^T = (\omega_\ell(t_1), ..., \omega_\ell(t_n))$. Then the model can be more succinctly written as

$$\sum_k \sum_\ell c_i^k \beta_{j\ell}(\Delta_t \boldsymbol{v}_{jk}^T \boldsymbol{\omega}_\ell) = \sum_\ell \beta_{j\ell} \boldsymbol{c}_i^T (\Delta_t \boldsymbol{V}_j \boldsymbol{\omega}_\ell) \tag{4.27}$$

$$= \Delta_t \boldsymbol{c}_i^T \boldsymbol{V}_j \boldsymbol{\omega} \boldsymbol{\beta}_j, \tag{4.28}$$

where $\boldsymbol{c}_i^T = (c_i^1, ..., c_i^K)$, $\boldsymbol{V}_j = [\boldsymbol{v}_{j1}|\cdots|\boldsymbol{v}_{jK}]^T$, $\boldsymbol{\omega} = [\boldsymbol{\omega}_1|\cdots|\boldsymbol{\omega}_L]^T$, and $\boldsymbol{\beta}_j = (\beta_{j1}, ..., \beta_{jL})^T$. Thus, the L_2-loss function can be expressed as

$$||\boldsymbol{y} - \sum_j \Delta_t \boldsymbol{C}\boldsymbol{V}_j\boldsymbol{\omega}\boldsymbol{\beta}_j||^2 = ||\boldsymbol{y} - \sum_j \boldsymbol{W}_j\boldsymbol{\beta}_j||^2, \tag{4.29}$$

for $\boldsymbol{C} = [\boldsymbol{c}_1, ..., \boldsymbol{c}_N]^T$ and $\boldsymbol{W}_j = \Delta_t \boldsymbol{C}\boldsymbol{V}_j\boldsymbol{\omega}$.

If, instead, the functional effects are bivariate functions of time and a scalar covariate, say concurrent position z_i, then they are approximated with a tensor product of orthogonal

basis functions $\{\omega_\ell\}_{\ell=1}^L$ and $\{\tau_r\}_{r=1}^R$ as $\gamma_j(t, z_i) \approx \sum_\ell \sum_r \beta_{j\ell r}\omega_\ell(t)\tau_r(z_i)$. Then the functional model can be rewritten as

$$\sum_j \int X_{ij}(t)\gamma_j(t, z_i)dt \approx \sum_j \int \left(\sum_k c_i^k v_j^k(t)\right)\left(\sum_\ell \sum_r \beta_{j\ell r}\omega_\ell(t)\tau_r(z_i)\right) dt \tag{4.30}$$

$$= \sum_j \sum_k \sum_\ell \sum_r c_i^k \beta_{j\ell r} \int v_j^k(t)\omega_\ell(t)\tau_r(z_i)dt \tag{4.31}$$

$$\approx \sum_j \Delta_t \boldsymbol{c}_i^T \boldsymbol{V}_j \boldsymbol{\omega} \otimes \boldsymbol{\tau}^T(z_i)\boldsymbol{\beta}_j, \tag{4.32}$$

with $\boldsymbol{\tau}^T(z_i) = (\tau_1(z_i), ..., \tau_R(z_i))$ and the other structures defined as before. Then the loss function can be written as $||\boldsymbol{y} - \sum_j \boldsymbol{W}_j\boldsymbol{\beta}_j||^2$ with $\boldsymbol{W}_j = \Delta_t \boldsymbol{c}_i^T \boldsymbol{V}_j \boldsymbol{\omega} \otimes \boldsymbol{\tau}^T(z_i)$.

4.3.3 Sparse preconditioned regression

The last stage in the Paul et al. [2008] framework is sparse regression of the preconditioned response on the original predictors. In the functional setting, this stage requires the use of the adaptive functional group lasso as in Gertheiss et al. [2013] or the SAFE procedure [Stallrich et al., 2020] in order to adapt the levels of smoothing and sparsity to each functional predictor. Paul et al. [2008] include all of the original predictors in this stage in case the supervision stage was too greedy and removed some important predictors. However, the data-driven approach for selecting the screening threshold θ can be tuned to be less greedy by altering the statistic used to estimate θ. For this reason, and because functional regression can be quite computationally intense for a large number of functional predictors, the predictors removed during supervision are not included in the final regression.

An aspect of this stage that Paul et al. [2008] neglect to address is selection of the regularization parameter when applying the lasso. When the lasso is performed in practice, V-fold CV is commonly used to perform tuning parameter selection and is the method implemented in the `glmnet` package Friedman et al. [2010]. However, CV for with a preconditioned response is more complex than in standard usage. Specifically, as we discovered during preliminary simulations, CV that trains and validates on $\widehat{\boldsymbol{y}}$ tends to over-select. We found that training the models on $\widehat{\boldsymbol{y}}$ and validating on $\boldsymbol{y}$ for each fold results in better performance. If $\boldsymbol{y}$ and $\widehat{\boldsymbol{y}}$ are exhibit moderate to high correlation, say greater than 0.7, then the 1-SE Rule can be used. If the correlation between $\boldsymbol{y}$ and $\widehat{\boldsymbol{y}}$ is weak, though, the standard errors generated by V-fold CV will be inflated and the 1-SE Rule will produce too sparse of a model, and often yield a trivial model. In such cases, the tuning parameter(s) that minimizes CV error should be selected.

Remark 4.3.2. A possible caveat to the above discussion is that the preconditioned response under supervised PCA has been shown to be a consistent estimator of $\boldsymbol{y}$ [Bair et al., 2006], but there is currently no analogous result for supervised FPCA, let alone supervised *sparse* FPCA. This could explain why CV with a preconditioned response did not merit discussion by Paul et al. [2008], but it is worth a formal investigation before a definitive explanation can be given.

4.4 Simulation Study

In this section, we investigate the performance of the three step approach through simulations. Three separate cases are considered with varying latent curve structures, where factors such as the numbers of relevant and irrelevant observed signals, sample size, and signal size are allowed to vary. Specific evaluation criteria are used at each stage of the analysis, with comparisons to competing methods when available. In particular, the supervision by Fake Factor Thresholding is evaluated in terms of model size and true and false positive rates, comparing the second and third quartiles and maximum to select the threshold parameter. The results of this stage inform the supervision for the rest of the analysis. The multivariate FPCA methods with and without supervision are evaluated by model size, true and false positives, proportion of variance explained, and the correlation between $\boldsymbol{y}$ and $\widehat{\boldsymbol{y}}$. The regression stage compares SAFE to preconditioned SAFE, with preconditioning from MFPCA, SMFPCA, or SLiM FPCA, as well as their supervised counterparts. Each of the competing methods is evaluated in terms of true and false positive rates and overall computation time. The summary metrics are defined as follows:

- **Model size:** $|\widehat{\mathcal{J}}|$, the number of functional coefficients in the estimated active set, $\widehat{\mathcal{J}}$;
- **True positives:** $TP = |\mathcal{J} \cap S|$, the number of important functional predictors in $\widehat{\mathcal{J}}$ that strongly depend on at least one predictive latent feature;
- **False positives:** $FP = |\mathcal{J} \cap S^c|$, the number of unimportant functional predictors in $\widehat{\mathcal{J}}$ that do not or weakly depend on a predictive latent feature;
- **Proportion of variance explained:** $PVE = \sum_{k=1}^{K} \mathrm{Var}(\boldsymbol{X}\boldsymbol{V}_k) / \sum_{j=1}^{m}\sum_{r=1}^{n} \mathrm{Var}(\boldsymbol{X}_{jr})$, the proportion of variance in the observed signals explained by the estimated latent features;
- **Correlation between $\boldsymbol{y}$ and $\widehat{\boldsymbol{y}}$:** $Corr(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = \sum_{i=1}^{N} (\boldsymbol{y} - \bar{\boldsymbol{y}})(\widehat{\boldsymbol{y}} - \bar{\widehat{\boldsymbol{y}}}) / (||\boldsymbol{y}|| \cdot ||\widehat{\boldsymbol{y}}||)$.

4.4.1 Data generation

The data is generated according to the latent factor functional linear regression model,

$$y_i = \mu + \sum_{k=1}^{K} \int_{\mathcal{T}} v_{ik}(t)\zeta_k(t)dt + e_i, \tag{4.33}$$

where y_i is a scalar response for $i = 1, ..., N$, μ is the mean response, $v_{ik}(t)$ is the ith realization of the kth latent curve with $t \in \mathcal{T} = [0, 1]$ and $k = 1, ..., K$, $\zeta_k(t)$ is the time-varying effect of the kth latent curve on the response, and $e_i \overset{iid}{\sim} N(0, \sigma_e^2)$ is white noise. For simplicity, we take $\mu = 0$ and $\sigma_e^2 = 1$. The latent functions $v_{ik}(t)$ are random combinations of known functions. That is, $v_{ik}(t) = \sum_{\ell=1}^{L} u_{i\ell} f_{k\ell}(t)$ with $u_{i\ell} \overset{ind}{\sim} N(0, 30\rho_\ell)$, where $\rho_\ell = e^{(-(\ell+1)/2)}$. Figure 4.1 visualizes the functions used for each of the scenarios described below. Specifically, we use Chebyshev polynomials $\{T_s(t) : s = 2, 3, 4\}$ with domain $[-1, 1]$ shifted to $[0, 1]$, B-spline basis functions $\{B_s(t) : s = 2, 3, 4\}$, and Fourier basis functions $\{F_s(t) : s = 2, 3, 4\}$. For the Fourier basis, an even s indicates a sine function $F_s(t) = \sqrt{2}\sin(2\pi dt)$ with $d = s/2$ and an odd s indicates a cosine function $F_s(t) = \sqrt{2}\cos(2\pi dt)$ with $d = (s - 1)/2$.

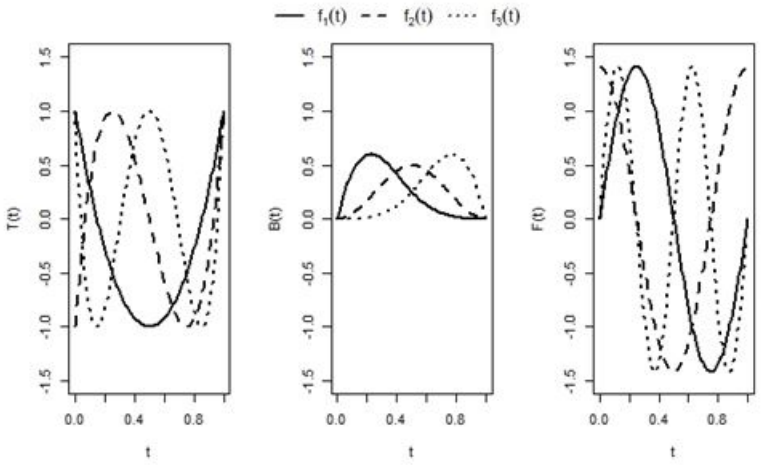

Figure 4.1 Underlying functions used to generate the latent curves for each example: Chebyshev polynomials (left), B-spline basis functions (center), and Fourier basis functions (right).

Although the latent curves generate the response, we assume they cannot be directly observed. Instead, the 'observed' signals $\{X_{ij}(t) : i = 1, ..., N; j = 1, ..., m\}$ are generated at

$n = 100$ equally-spaced time points $t \in \mathcal{T} = [0, 1]$ as

$$X_{ij}(t) = \sum_{k=1}^{K} \alpha_{jk} v_{ik}(t), \tag{4.34}$$

where α_{jk} is the realization of the kth factor in the jth observed signal. The realization factors α_{jk} are generated as mean zero normal random variables with variance $20C_k$, $C_k = (K+1-k)/K$, for j in the set S of important predictors and variance 0.05 for $j \notin S$. Finally, assuming each observed predictor has an associated time-varying effect $\gamma_j(t)$, we estimate the surrogate model,

$$E[y_i | X_{i1}(t), ..., X_{im}(t)] = \sum_{j=1}^{m} \int_{\mathcal{T}} X_{ij}(t) \gamma_j(t) dt. \tag{4.35}$$

The following examples describe three different processes for generating the above quantities.

Example 4.4.1. We first consider a model with a two latent features, $v_1(t)$ and $v_2(t)$, with functional effects $\zeta_1(t) = 1 + \sqrt{2}\cos(\pi t)$ and $\zeta_2(t) = 0$. Setting $\zeta_2(t) = 0$ promotes the need for supervision to screen out the observed predictors with sole dependence on $v_2(t)$. Realizations of $v_1(t)$, $v_{i1}(t)$, are combinations of Chebyshev polynomials with $f_{11}(t) = T_3(t)$, $f_{12}(t) = T_4(t)$, and $f_{13}(t) = T_2(t)$. Realizations of $v_2(t)$ are combinations of B-spline basis functions with $f_{2\ell}(t) = B_{\ell+1}(t)$, $\ell = 1, 2, 3$. The observed predictors for $j = 1, 2, 3$ are generated with $\alpha_{j2} = 0$ and $\alpha_{j1} \sim N(0, 20C_1)$; for $j = 4, 5$, $\alpha_{j2} = 0$ but $\alpha_{j1} \sim N(0, 0.05)$; for $j > 5$, $\alpha_{j1} = 0$ and $\alpha_{j2} \sim N(0, 0.05)$. With this set-up, only the first three predictors strongly depend on $v_1(t)$. We generate 100 data sets with $N = 50$ samples and $m = 10$ observable predictors.

Example 4.4.2. The second model also contains two latent features but with functional effects $\zeta_1(t) = \sin^2(6t)$, which is more wiggly than the non-zero effect in the previous example, and $\zeta_2(t) = 0$. The realizations $v_{i1}(t)$ are again generated from Cheybshev polynomials but with $f_{1\ell}(t) = T_{\ell+1}(t)$, and the $v_{i2}(t)$ are generated from B-spline basis functions as before. The predictors for $j = 1, 2, 3$ are generated as in the previous example; for $j = 4, 5$, $\alpha_{jk} \sim N(0, 20C_k)$; and for $j > 5$, $\alpha_{j1} = 0$ and $\alpha_{j2} \sim N(0, 20C_1)$. Hence, the first five predictors strongly depend on $v_1(t)$, two of which also depend on $v_2(t)$, and the remaining predictors strongly depend on $v_2(t)$. Again, 100 data sets with $N = 50$ samples and $m = 10$ observable predictors are generated.

Example 4.4.3. This model contains $K = 3$ latent signals with $N = 100$ measured responses. Realizations of the first latent curve are generated from Chebyshev functions with

$f_{11}(t) = T_4(t)$, $f_{12}(t) = T_2(t)$, and $f_{13}(t) = T_3(t)$; the second curve from B-spline basis functions with $f_{2\ell}(t) = B_{\ell+1}(t)$; and the third from Fourier basis functions with $f_{3\ell}(t) = F_{\ell+1}(t)$. The corresponding functional effects are $\zeta_1(t) = \sin^2(6t)$, $\zeta_2(t) = \cos(6t)$, and $\zeta_3(t) = 0$. The $X_{ij}(t)$ are generated as follows. For $j = 1, 2$, $\alpha_{jk} = 0$ for $k = 2, 3$ so that $X_{ij}(t)$ only depends on $v_1(t)$. Similarly, $\alpha_{jk} = 0$, $k = 1, 3$, for $j = 3, 4$ and $\alpha_{j3} = 0$ for $j = 5, 6$. For $j = 7, 8$, the $X_{ij}(t)$ depend on all three latent signals, and for $j > 8$ the observed signals solely depend on $v_3(t)$. However, the variance of α_{j3} is fixed at 0.05 only for $j > 10$. A total of $m = 20$ observable predictors are generated and, due to increased complexity of the data, only 50 data sets are generated.

In the supervision stage, the fake factors are generated similarly to Tutz & Gertheiss [2010]. In particular, each of the $Q = 100$ fake factor observations are generated as

$$X_q(t) = \sum_{r=1}^{5} \{\sin(2\pi t(5 - a_{qr})) - m_{qr}\}, \tag{4.36}$$

at $n = 100$ equally-spaced time points $t \in \mathcal{T}$, where $a_{qr} \sim U(0, 5)$ and $m_{qr} \sim U(0, 2\pi)$. The curves are then centered and scaled by time point. As described in Sections 1.2.3 and 3.2, the corresponding coefficient functions are approximated with a linear combination of 10 orthonormal cubic B-spline basis functions and the integral with a Riemann sum since the fake factors are densely generated on $\mathcal{T}$. The smoothed regression problem is solved with smoothing parameter φ chosen from $\log \varphi \in \{-10, -5, 0, 5, 10\}$. The fake factor generation and regression process is repeated $D = 50$ times to obtain $\widehat{\theta}$.

For each of the described examples, the response and observed predictors are centered prior to analysis. The linear approximation of the functional model described at the beginning of Section 3.2 is conducted with 10 orthonormal cubic B-spline basis functions. The sparsity and smoothing parameters in the preconditioning and sparse regression stages are selected from $\log \lambda \in \{-20, -19, ..., 3, 4\}$ and $\log \varphi \in \{-10, -5, 0, 5, 10\}$, respectively, using the 1-SE Rule. A data-driven range of regularization parameters is used for the FPCA stage when sparsity is induced. Specifically, the maximum value $\lambda_{\max}$ is determined according to Remark 4.3.1 for SLiM FPCA and the full sequence on the log scale ranges from -10 to $\lambda_{\max}$ with increments of 1. A similar approach is adopted to determine $\lambda_{\max}$ for SMFPCA. That is, let $\boldsymbol{X}^T = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$ denote the SVD of $\boldsymbol{X}^T$, where $\boldsymbol{X}$ is the full augmented matrix of the $\boldsymbol{X}_i$'s defined in Section 4.2.2. Define $\boldsymbol{V}_K$ as the first K columns of $\boldsymbol{V}$. Then $\lambda_{\max} = \max_i ||\boldsymbol{X}_i^T \boldsymbol{V}_K||_\infty$, where $||\boldsymbol{x}||_\infty = \max(|x_1|, ..., |x_n|)$ for an n-dimensional vector $\boldsymbol{x}$. The full range of regularization parameters is then defined as for SLiM FPCA.

Remark 4.4.1. Note that the observed signals are generated without noise on a dense grid

of time points. If the $X_{ij}(t)$ were sparsely observed or contaminated with noise, then the observed trajectories would need to be smoothed prior to supervision and SAFE. Appropriate smoothing methods are given by Yao et al. [2005], Xiao et al. [2018], James et al. [2000], and Paul & Peng [2009].

4.4.2 Results

Supervision results for each of the examples are given in Table 4.2, where the goal is to retain only predictors with strong dependence on the latent factor(s) with non-zero effects. That is, we desire $TP \leq 3, 5, 8$, respectively. However, the two metrics we consider are unable to distinguish between high and low levels of dependence. Example 4.4.1 highlights this screening limitation by all five of the variables with any amount of dependence on $v_1(t)$ being retained. However, for all three examples, the variables with no dependence on the important latent factor(s) are typically dropped from the model. In the models where unimportant variables are kept, the screening procedure is unable to remove any variables from the model due to the randomness of the data generation. These occurrences explain the elevated false positive standard errors. Another interesting observation is in Example 4.4.3. The predictors with $j = 3, 4$ have sole dependence on $v_2(t)$, but that factor's contribution to the response is weaker than $v_1(t)$. Thus, $X_3(t)$ and $X_4(t)$ are occasionally screened from the model. However, the predictors with contribution from both latent factors ($j = 5, 6$), or even all three factors ($j = 7, 8$), are retained in those scenarios. Finally, since $\text{Corr}(\boldsymbol{y}, \widehat{\boldsymbol{y}}_j)$ consistently performs slightly better than $||X_j(t)\gamma_j(t)||$, all further analyses with supervision utilize the correlation metric.

Next, Table 4.3 displays the results of the three preconditioning methods considered. In terms of selection, SMFPCA tends to produce a sparser model than SLiM FPCA, likely only selecting the strongest signals or the signals most strongly exhibiting similar patterns. On the other hand, SLiM FPCA seems to try to keep any signal that was generated by a latent factor. Interestingly, the unsupervised versions of SMFPCA and SLiM FPCA both yield sparser models than the supervision step did for Example 4.4.1 (2.13 vs 5.65, SMFPCA; 2.95 vs 5.35, SLiM FPCA), but the same success does not carry over to the other examples or for the supervised versions. Likely, the weakly observed Chebyshev functions ($j = 4, 5$) and B-spline basis functions ($j > 5$) are similar enough that the two sparse preconditioning methods do not distinguish between the curve features and so remove all variables with $j > 3$, whereas the variables with $j > 5$ have already been removed in the supervised versions and so the variables with weak dependence on $v_1(t)$ are often retained.

Although MFPCA does not induce sparsity, it does establish a benchmark for proportion

s_j	Q2	Q3	Max
	Example 4.4.1		
$\text{Corr}(\boldsymbol{y}, \widehat{\boldsymbol{y}}_j)$	6.95 (0.244)	**5.65 (0.168)**	5 (0)
	3 (0)	**3 (0)**	3 (0)
	3.95 (5.948)	**2.65 (2.828)**	2 (0)
$\|\|X_j(t)\gamma_j(t)\|\|$	7.05 (0.246)	6 (0.2)	5 (0)
	3 (0)	3 (0)	3 (0)
	4.05 (6.047)	3 (4)	2 (0)
	Example 4.4.2		
$\text{Corr}(\boldsymbol{y}, \widehat{\boldsymbol{y}}_j)$	6.65 (0.235)	**5.35 (0.128)**	2.29 (0.229)
	5 (0)	**5 (0)**	2.29 (0.229)
	1.65 (5.528)	**0.35 (1.628)**	0 (0)
$\|\|X_j(t)\gamma_j(t)\|\|$	6.9 (0.243)	5.6 (0.162)	2.97 (0.226)
	5 (0)	5 (0)	2.97 (0.226)
	1.9 (5.89)	0.6 (2.64)	0 (0)
	Example 4.4.3		
$\text{Corr}(\boldsymbol{y}, \widehat{\boldsymbol{y}}_j)$	11.08 (0.749)	**8.4 (0.48)**	6.08 (0.055)
	7.96 (0.04)	**7.44 (0.127)**	6.08 (0.055)
	3.12 (27.706)	**0.96 (10.598)**	0 (0)
$\|\|X_j(t)\gamma_j(t)\|\|$	11.56 (0.782)	9.12 (0.56)	6.08 (0.055)
	7.96 (0.04)	7.68 (0.104)	6.08 (0.055)
	3.6 (30.24)	1.44 (15.206)	0 (0)

Table 4.2 Supervision results for each example. For each metric, the rows give mean (SE) model size, true positives, and false positives, respectively. Results in bold denote supervision results for the full analyses given the choices of s_j and $\widehat{\theta}$.

of variance explained (PVE) and correlation between the response $\boldsymbol{y}$ and preconditioned response $\widehat{\boldsymbol{y}}$ for the other two methods. SLiM FPCA achieves similar PVE and correlation values to MFPCA, where as the correlations for unsupervised SMFPCA vary a bit. In Example 4.4.1, where unsupervised SMFPCA typically drops one of the three important variables, its preconditioned response is less correlated with $\boldsymbol{y}$ than the other two preconditioners. However, unsupervised SMFPCA yields higher correlations for the other two examples, which may be a result of its assumed latent factor model being more appropriate for the simulation examples. SMFPCA also has a smaller mean model size than SLiM FPCA in Examples 4.4.2 4.4.3 but a higher preconditioned correlation value. Further investigation into the cause of this phenomenon may reveal interesting properties of SMFPCA.

As shown in Tables 4.4 and 4.5, the application of 5-stage SAFE to the preconditioned response further reduces the number of selected signals, often tending toward the number of latent factors with non-zero effects in each example. That is, TP tends toward 1 for Examples 4.4.1 and 4.4.2 and toward 2 for Example 4.4.3 for each method. However, SAFE without preconditioning still outperforms the other methods, and supervision isn't necessarily helpful for selection. Overall, the results in these tables have made it apparent that, although SAFE

Method	Example 4.4.1	Example 4.4.2	Example 4.4.3
	Unsupervised		
MFPCA	0.87 (0.003)	0.85 (0.004)	0.85 (0.003)
	0.75 (0.007)	0.5 (0.013)	0.55 (0.02)
SMFPCA	2.13 (0.075)	7.4 (0.224)	8.8 (0.252)
	2.13 (0.007)	4.33 (0.009)	6.68 (0.025)
	0 (0)	3.07 (0.167)	2.12 (0.166)
	-	-	-
	0.64 (0.013)	0.67 (0.011)	0.7 (0.01)
SLiM FPCA	2.95 (0.113)	8.42 (0.168)	11.62 (0.465)
	2.78 (0.004)	4.76 (0.005)	6.94 (0.02)
	0.17 (0.101)	3.66 (0.157)	4.68 (0.414)
	0.86 (0.004)	0.85 (0.004)	0.84 (0.004)
	0.75 (0.007)	0.5 (0.013)	0.55 (0.02)
	Supervised		
MFPCA	0.87 (0.005)	0.86 (0.004)	0.86 (0.005)
	0.76 (0.007)	0.48 (0.013)	0.61 (0.013)
SMFPCA	3.48 (0.116)	3.68 (0.175)	6.36 (0.24)
	1.99 (0.008)	3.43 (0.012)	6.14 (0.027)
	1.49 (0.063)	0.25 (0.103)	0.22 (0.119)
	-	-	-
	0.7 (0.014)	0.5 (0.015)	0.66 (0.014)
SLiM FPCA	4.68 (0.058)	4.96 (0.136)	7.3 (0.277)
	2.79 (0.005)	4.68 (0.006)	6.92 (0.019)
	1.89 (0.035)	0.28 (0.113)	0.38 (0.228)
	0.86 (0.005)	0.85 (0.005)	0.86 (0.005)
	0.76 (0.008)	0.48 (0.014)	0.61 (0.012)

Table 4.3 FPCA results for Examples 4.4.1-4.4.3. For SMFPCA and SLiM FPCA, mean (SE) size, true positives, false positives, proportion of variance explained, and correlation between y and $\widehat{y}$ are presented. For MFPCA, only proportion of variance explained and correlation are presented since sparsity is not induced.

doesn't directly model latent factors, the multiple stages with adaptive weighting promote selection of as few signals as possible, particularly the signal(s) with strongest relation to the working response.

The final aspect of this framework that must be considered, though, is the total computation time required for each method. Table 4.6 shows this information for each of the examples with time presented in minutes. In some cases, the preconditioning methods achieve slight reductions in computation time, but the supervision stage requires noticeably more time for these simplistic simulations. In combination with the selection results in Table 4.5, the supervision and preconditioning stages do not appear to be overly beneficial as compared to unsupervised SAFE conducted on the original response.

4.5 Application to EMG data

The developed framework is now applied to the EMG data with the bivariate coefficient model described in Chapter 1. For proper comparison to the results of Stallrich et al. [2020], the data is prepared in the same manner using the windowing procedure and thinning the data by retaining every 20th observation, the `gglasso` package is used in computation for 5-stage SAFE, and the same ranges of tuning parameters are considered. That is, the smoothing parameters φ_t, φ_z satisfy $\log \varphi \in \{-10, -5, 0, 5, 10\}$ and the log sparsity parameter $\log \lambda$ is selected from the sequence from -15 to -2 with increments of 0.2. The sparsity parameter ranges for SMFPCA and SLiM FPCA are computed as described in Section 4.4.1. The fake factors for supervision are also generated as in the previous section, but then restructured and thinned similar to the EMG data. Also, due to the success of $\text{Corr}(\boldsymbol{y}, \widehat{\boldsymbol{y}}_j)$ in the simulation study, we do not consider the other supervision metric here.

As Table 4.7 indicates, very few EMG signals are screened in the supervision stage, even when determining $\widehat{\theta}$ with the maximum order statistic. Although the externally generated signal (EMG 9) is screened for each of the six data sets using the maximum, we expected EMGs 3 and 6 to also be screened more consistently, especially given their lower correlations with the response relative to the other EMGs. An alternative method for generating the fake factors may result in improved screening performance. Unsurprisingly, however, no other signals are screened since many of the EMGs are highly correlated with each other and so should be similarly correlated with the response.

The results from the rest of the analytic framework are presented in Table 4.8. As in the simulation study, SLiM FPCA achieves similar PVE and correlation values to MFPCA. SMFPCA, on the other hand, tends to differ in its correlation values, sometimes with great improvements over MFPCA. Ideally, an improvement in correlation would lead to an

Method	Stage				
	1	2	3	4	5
	Example 4.4.1				
	Unsupervised				
SAFE	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
MFPCA	1.01 (0.001)	1 (0)	1 (0)	1 (0)	1 (0)
	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
SMFPCA	1.08 (0.003)	1.03 (0.002)	1.02 (0.001)	1.02 (0.001)	1.02 (0.001)
	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
SLiM	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
FPCA	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
	Supervised				
SAFE	0.69 (0.005)	0.62 (0.005)	0.62 (0.005)	0.62 (0.005)	0.62 (0.005)
	0.59 (0.006)	0.43 (0.005)	0.41 (0.005)	0.4 (0.005)	0.4 (0.005)
MFPCA	0.71 (0.005)	0.68 (0.005)	0.66 (0.005)	0.66 (0.005)	0.66 (0.005)
	0.62 (0.006)	0.54 (0.006)	0.51 (0.006)	0.49 (0.006)	0.49 (0.006)
SMFPCA	0.8 (0.004)	0.77 (0.004)	0.76 (0.005)	0.76 (0.005)	0.76 (0.005)
	0.78 (0.006)	0.57 (0.006)	0.54 (0.006)	0.54 (0.006)	0.51 (0.006)
SLiM	0.73 (0.005)	0.68 (0.005)	0.66 (0.005)	0.66 (0.005)	0.66 (0.005)
FPCA	0.6 (0.006)	0.53 (0.006)	0.5 (0.006)	0.5 (0.006)	0.48 (0.005)
	Example 4.4.2				
	Unsupervised				
SAFE	1.07 (0.003)	1 (0)	1 (0)	1 (0)	1 (0)
	0.26 (0.004)	0.08 (0.003)	0.06 (0.002)	0.06 (0.002)	0.06 (0.002)
MFPCA	1.22 (0.005)	1.13 (0.004)	1.11 (0.003)	1.11 (0.003)	1.11 (0.003)
	0.87 (0.007)	0.78 (0.006)	0.76 (0.005)	0.75 (0.005)	0.73 (0.005)
SMFPCA	1.66 (0.011)	1.26 (0.005)	1.18 (0.004)	1.17 (0.004)	1.15 (0.004)
	0.83 (0.011)	0.63 (0.009)	0.51 (0.007)	0.5 (0.007)	0.48 (0.007)
SLiM	1.39 (0.008)	1.25 (0.005)	1.21 (0.005)	1.21 (0.005)	1.19 (0.005)
FPCA	0.86 (0.006)	0.75 (0.006)	0.73 (0.006)	0.72 (0.006)	0.7 (0.006)
	Supervised				
SAFE	1.17 (0.004)	1.02 (0.001)	1.02 (0.001)	1.02 (0.001)	1.02 (0.001)
	0.02 (0.001)	0.02 (0.001)	0.02 (0.001)	0.01 (0.001)	0.01 (0.001)
MFPCA	1.89 (0.005)	1.7 (0.005)	1.69 (0.005)	1.68 (0.005)	1.66 (0.005)
	0.07 (0.003)	0.07 (0.003)	0.07 (0.003)	0.07 (0.003)	0.07 (0.003)
SMFPCA	1.9 (0.008)	1.69 (0.006)	1.68 (0.006)	1.67 (0.007)	1.66 (0.007)
	0.1 (0.005)	0.05 (0.002)	0.05 (0.002)	0.05 (0.002)	0.05 (0.002)
SLiM	2.1 (0.008)	1.85 (0.006)	1.77 (0.006)	1.76 (0.006)	1.74 (0.006)
FPCA	0.08 (0.004)	0.08 (0.004)	0.07 (0.003)	0.07 (0.003)	0.07 (0.003)

Table 4.4 Mean (SE) true positives and false positives for Examples 4.4.1 (top) and 4.4.2 (bottom) from sparse preconditioned regression compared to SAFE.

Method	Stage				
	1	2	3	4	5
	Example 4.4.3				
	Unsupervised				
SAFE	1.86 (0.016)	1.46 (0.011)	1.38 (0.011)	1.38 (0.011)	1.36 (0.011)
	0.28 (0.009)	0.1 (0.006)	0.02 (0.003)	0 (0)	0 (0)
MFPCA	2.94 (0.031)	2.26 (0.013)	2.14 (0.012)	2.12 (0.013)	2.08 (0.012)
	1 (0.027)	0.5 (0.01)	0.5 (0.01)	0.48 (0.01)	0.48 (0.01)
SMFPCA	3.46 (0.039)	2.72 (0.028)	2.32 (0.017)	2.22 (0.014)	2.14 (0.015)
	0.9 (0.014)	0.54 (0.012)	0.44 (0.01)	0.44 (0.01)	0.44 (0.01)
SLiM FPCA	2.9 (0.032)	2.36 (0.023)	2.08 (0.011)	2.04 (0.012)	2.02 (0.012)
	0.94 (0.022)	0.64 (0.013)	0.5 (0.01)	0.46 (0.01)	0.46 (0.01)
	Supervised				
SAFE	2.16 (0.02)	1.64 (0.013)	1.42 (0.011)	1.4 (0.011)	1.4 (0.011)
	0.04 (0.004)	0.04 (0.004)	0.02 (0.003)	0 (0)	0 (0)
MFPCA	3.68 (0.033)	2.62 (0.018)	2.46 (0.015)	2.44 (0.015)	2.4 (0.016)
	0.1 (0.007)	0.06 (0.005)	0.06 (0.005)	0.06 (0.005)	0.06 (0.005)
SMFPCA	3.86 (0.034)	3.42 (0.03)	2.94 (0.026)	2.78 (0.023)	2.64 (0.023)
	0.08 (0.007)	0.06 (0.005)	0.06 (0.005)	0.06 (0.005)	0.06 (0.005)
SLiM FPCA	3.62 (0.034)	2.78 (0.021)	2.5 (0.016)	2.48 (0.017)	2.34 (0.015)
	0.08 (0.005)	0.06 (0.005)	0.06 (0.005)	0.06 (0.005)	0.06 (0.005)

Table 4.5 Mean (SE) true positive and false positive rates for Example 4.4.3 from sparse preconditioned regression compared to SAFE.

Method	Example 4.4.1	Example 4.4.2	Example 4.4.3
	Unsupervised		
SAFE	0.05 (0.001)	2.04 (0.026)	11.19 (0.229)
MFPCA	0.51 (0.014)	2.04 (0.021)	10.76 (0.469)
SMFPCA	1.43 (0.015)	2.84 (0.021)	10.83 (0.129)
SLiM FPCA	0.55 (0.016)	2.29 (0.022)	10.8 (0.442)
	Supervised		
SAFE	3.71 (0.028)	3.75 (0.027)	12.18 (0.209)
MFPCA	2.5 (0.016)	3.68 (0.022)	9.76 (0.319)
SMFPCA	4.89 (0.046)	4.61 (0.053)	11.49 (0.152)
SLiM FPCA	2.64 (0.018)	3.69 (0.023)	13.98 (0.682)

Table 4.6 Mean (SE) overall computation times in minutes.

$\widehat{\theta}$	C1	C2	C3	R1	R2	R3
Q2	0.10	0.10	0.09	0.11	0.10	0.10
Q3	0.16	0.15	0.14	0.16	0.15	0.16
Max	0.39	0.40	0.41	0.34	0.31	0.30
EMGs						
1	0.92	0.89	0.89	0.85	0.87	0.88
2	0.84	0.88	0.91	0.71	0.67	0.52
3	0.73	0.50	0.75	0.39	0.40	**0.13**
4	0.86	0.84	0.87	0.68	0.68	0.69
5	0.91	0.84	0.87	0.86	0.86	0.87
6	0.61	0.66	0.46	**0.16**	**0.23**	0.40
7	0.89	0.88	0.85	0.90	0.91	0.90
8	0.93	0.92	0.94	0.74	0.85	0.66
9	**0.35**	**0.20**	**0.30**	**0.08**	**0.12**	**0.17**
10	0.90	0.93	0.87	0.68	0.79	0.69
11	0.87	0.86	0.86	0.77	0.79	0.82
12	0.90	0.88	0.91	0.75	0.63	0.45
13	0.83	0.85	0.88	0.73	0.74	0.77
14	0.90	0.90	0.88	0.75	0.77	0.68
15	0.89	0.83	0.85	0.69	0.80	0.71
16	0.88	0.80	0.86	0.69	0.64	0.50

Table 4.7 Estimates of the thresholding parameter θ and marginal correlations between EMG signals and response. Bold correlation values denote screening EMGs based on the maximum order statistic for θ.

improvement in final selection. However, SAFE preconditioned by unsupervised SMFPCA actually reduces the active set to one signal with the second random data set, even though the correlation is double that of the other two FPCA methods. Interestingly, with the third random data set, unsupervised SMFPCA achieves a correlation value just below the threshold of 0.7, and so minimum CV error is used for tuning parameter selection, the likely cause of poor selection. Along with investigation into the properties of SMFPCA, this suggests the need for further study of an ideal correlation threshold or the development of a data-driven strategy for determining the threshold.

Although none of the methods uniformly match or outperform SAFE in terms of selection and time, respectively, there are still reductions in computation time in many instances. Supervised SMFPCA appears to be a good preconditioner, overall, even though it struggles a little with the second and third random data sets. In fact, all three preconditioning techniques tend to yield lower correlation values between the original and preconditioned response for the third random data set. This may be due to poor smoothing parameter exploration when

determining $\widehat{\boldsymbol{y}}$ or the random nature of the original response. SLiM FPCA also shows some promise for this application, but selection suffers for lower correlation values and computation tends to require more time than under SMFPCA or SAFE by itself.

Method	C1	C2	C3	R1	R2	R3	C1	C2	C3	R1	R2	R3
	Unsupervised						*Supervised*					
	2 | 2	2 | 1	2 | 2	2 | 0	3 | 1	2 | 1	2 | 1	3 | 1	2 | 1	2 | 0	3 | 0	2 | 0
SAFE	2 | 0	2 | 0	2 | 0	2 | 0	2 | 0	2 | 0	2 | 0	2 | 0	2 | 0	2 | 0	3 | 0	2 | 0
	8.59	9.93	11.62	7.44	16.89	15.55	8.45	10.18	11.84	6.54	18.38	15.51
	-	-	-	-	-	-	-	-	-	-	-	-
	0.85	0.86	0.85	0.82	0.82	0.9	0.83	0.85	0.85	0.86	0.89	0.86
	0.95	0.96	0.92	0.8	0.45	0.23	0.79	0.74	0.77	0.87	0.4	0.19
MFPCA	2 | 2	3 | 2	2 | 2	2 | 1	3 | 6	3 | 7	2 | 1	2 | 0	2 | 1	3 | 3	3 | 5	3 | 11
	2 | 0	2 | 0	2 | 0	1 | 0	3 | 5	3 | 3	2 | 0	2 | 0	2 | 0	2 | 0	3 | 4	3 | 11
	9.58	10.64	9.87	4.54	5.62	3.88	11.93	7.88	11.19	4.81	6.12	3.68
	3 | 7	3 | 6	3 | 11	3 | 11	3 | 11	3 | 8	3 | 6	3 | 5	3 | 5	3 | 10	3 | 11	3 | 5
	-	-	-	-	-	-	-	-	-	-	-	-
	0.9	0.88	0.95	0.83	0.9	0.67	0.9	0.88	0.82	0.83	0.9	0.19
SMFPCA	2 | 2	3 | 2	3 | 5	2 | 1	3 | 1	3 | 8	3 | 4	3 | 2	3 | 2	2 | 0	3 | 1	3 | 5
	2 | 0	2 | 0	2 | 1	2 | 0	1 | 0	3 | 3	2 | 0	2 | 0	2 | 0	2 | 0	1 | 0	3 | 4
	8.74	9.73	9.96	4.62	8.63	6.42	8.01	8.24	6.54	3.85	8.52	2.27
	3 | 11	3 | 11	3 | 11	3 | 11	3 | 11	3 | 11	3 | 11	3 | 11	3 | 11	3 | 11	3 | 11	3 | 10
	0.82	0.83	0.83	0.77	0.79	0.87	0.8	0.82	0.83	0.82	0.87	0.84
SLiM	0.95	0.96	0.92	0.89	0.45	0.23	0.61	0.61	0.76	0.77	0.14	0.29
FPCA	2 | 2	3 | 2	2 | 2	3 | 5	3 | 6	3 | 7	3 | 2	3 | 5	3 | 2	2 | 0	3 | 8	3 | 5
	2 | 0	2 | 0	2 | 0	2 | 0	3 | 5	3 | 3	2 | 1	3 | 1	2 | 0	1 | 0	3 | 4	3 | 3
	9.18	10.77	9.05	5.39	5.5	3.9	11.22	18.97	11.49	2.97	5.69	4.19

Table 4.8 Unsupervised (left) and supervised (right) results from the preconditioning and final regression stages. For each method, we report true and false positives (TP|FP) from each FPCA method, the percentage of variance in $\boldsymbol{X}$ explained by the retained PCs, correlation between $\boldsymbol{y}$ and $\widehat{\boldsymbol{y}}$, TP|FP for SAFE in Stage 1 and Stage 5, and the total computation time in hours.

4.6 Discussion

This chapter has extended the 3-stage framework of Paul et al. [2008] to the case of functional data with latent functions. We first extended the supervision framework of Bair et al. [2006], utilizing the correlation between the original and predicted responses from univariate functional linear regressions on each functional predictor. Additionally, a data-driven selection of the screening threshold was developed that relates noisy signals to the response to derive a distribution for the thresholding parameter. Although our fake factor generation was sufficient in the simulations to yield appropriate thresholds, the generation scheme could be improved for practical applications, especially the EMG data. Next, we compared three preconditioning methods, one of which, SLiM FPCA, is a novel extension of Wang & Tsung [2020] that is suitable for a variety of functional data applications. Finally, an appropriate version of V-fold cross-validation using both the preconditioned and original response data was formalized to perform improved tuning parameter and model selection.

The typical advantage of screening procedures is a significant reduction of a data set's dimension with little computational expense. Although this level of dimension reduction occurred in the simulations, there was not as much benefit in the analysis of the EMG data with respect to screening and overall computation time. However, this kind of supervision is likely to perform better for higher-dimensional data sets with relatively fewer correlated predictors. Further, alternative ways of generating the fake variables for determining the thresholding parameter θ should be explored. For example, the fake signals could be generated as random combinations of the eigenfunctions estimated from the actual data. This would ensure some degree of similarity between the true and phony signals.

A more subtle advantage of conducting supervision through screening is it yields a hierarchy of importance for the functional predictors. Considering multiple percentiles of the simulated θ distribution combined with examining the values of the s_j metric in low-dimensional scenarios, presents an ordering of the variables not unlike traditional stepwise selection. In fact, this stage could be used by itself to inexpensively gain an understanding of variable importance in ultrahigh-dimensional settings.

The sparse preconditioning methods, namely SMFPCA and SLiM FPCA, can also be used independently of the three-stage framework, or at least without supervision, since they scale well to large problems. Additionally, they can be used in situations where response data are not available to perform some selection of predictors and to gain an understanding of the underlying relationships between the functional predictors. For the EMG application in particular, this strategy could be helpful when training robotic devices for bilateral transradial amputees where hand and finger movement cannot be observed.

A notable limitation of the current implementation of this framework lies in the final stage. Computation time was not greatly improved in the analysis of the EMG data as compared to just conducting 5-stage SAFE. However, as noted by Yau & Hui [2017] and demonstrated between this chapter and Chapter 3, the `gglasso` algorithm is not as efficient in high-dimensional settings as the GLODE algorithm is. Future iterations of this framework should adopt the GLODE algorithm within the implementation of SAFE.

Another shortcoming of the current work is the lack of effect estimation and prediction. Sensor selection is not the end goal for EMG data analysis; a predictive model is needed to translate EMG signals into hand movement. Future work should also implement a final estimation stage using the original response and the selected signals, where the reconstructed signals may be used to model the latent muscle activity more directly.

BIBLIOGRAPHY

Alquier, P. & Doukhan, P. (2011). "Sparsity considerations for dependent variables". *Electron. J. Statist.* **5**, pp. 750–774.

Bair, E. et al. (2006). "Prediction by Supervised Principal Components". *Journal of the American Statistical Association* **101**.473, pp. 119–137.

Barshan, E. et al. (2011). "Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds". *Pattern Recognit.* **44**, pp. 1357–1371.

Berrendero, J. et al. (2011). "Principal components for multivariate functional data". *Computational Statistics & Data Analysis* **55**.9, pp. 2619–2634.

Besse, P. & Ramsay, J. (1986). "Principal components analysis of sampled functions". *Psychometrika* **51**, pp. 285–311.

Bickel, P. et al. (2009). "Simultaneous analysis of LASSO and Dantzig selector". *The Annals of Statistics* **37**.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag.

Boor, C. de (2001). *A Practical Guide to Spline*. Vol. 27. Springer, New York.

Butterworth, S. (1930). "On the Theory of Filter Amplifiers". *Wireless Engineer* **7**, pp. 536–541.

Cardot, H. (2000). "Nonparametric estimation of smoothed principal components analysis of sampled noisy functions". *Journal of Nonparametric Statistics - J NONPARAMETR STAT* **12**.

— (2007). "Conditional functional principal components analysis". *Scandinavian Journal of Statistics* **13**, pp. 317–335.

Cardot, H. et al. (2003). "SPLINE ESTIMATORS FOR THE FUNCTIONAL LINEAR MODEL". *Statistica Sinica* **13**.3, pp. 571–591.

Castro, P. E. et al. (1986). "Principal Modes of Variation for Processes with Continuous Sample Curves". *Technometrics* **28**.4, pp. 329–337.

Chiou, J.-M. & Müller, H.-G. (2009). "Modeling Hazard Rates as Functional Data for the Analysis of Cohort Lifetables and Mortality Forecasting". *Journal of the American Statistical Association* **104**.486, pp. 572–585.

Chiou, J.-M. et al. (2003). "Functional Quasi-Likelihood Regression Models with Smooth Random Effects". *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **65**.2, pp. 405–423.

Chiou, J.-M. et al. (2014). "MULTIVARIATE FUNCTIONAL PRINCIPAL COMPONENT ANALYSIS: A NORMALIZATION APPROACH". *Statistica Sinica* **24**.4, pp. 1571–1596.

Collazos, J. A. et al. (2016). "Consistent variable selection for functional regression models". *Journal of Multivariate Analysis* **146**. Special Issue on Statistical Models and Methods for High or Infinite Dimensional Spaces, pp. 63–71.

Cordella, F. et al. (2016). "Literature Review on Needs of Upper Limb Prosthesis Users". *Front Neurosci* **1**.

Crouch, D. L. & Huang, H. (2016). "Lumped-parameter electromyogram-driven musculoskeletal hand model: A potential platform for real-time prosthesis control". *Journal of Biomechanics* **49**, pp. 3901–3907.

Crouch, D. L. & Huang, H. H. (2017). "Musculoskeletal model-based control interface mimics physiologic hand dynamics during path tracing task". *J. Neural. Eng.* **14**.036008.

Efron, B. et al. (2004). "Least angle regression". *The Annals of Statistics* **32**.2, pp. 407–499.

Fan, Y. et al. (2015). "Functional additive regression". *Ann. Statist.* **43**.5, pp. 2296–2325.

Fang, X. et al. (2017). "Scalable prognostic models for large-scale condition monitoring applications". *IISE Transactions* **49**.7, pp. 698–710.

Friedman, J. et al. (2007). "Pathwise Coordinate Optimization". *The Annals of Applied Statistics* **1**.2, pp. 302–332.

Friedman, J. et al. (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent". *Journal of Statistical Software* **33**.1, pp. 1–22.

Gertheiss, J. et al. (2013). "Variable Selection in Generalized Functional Linear Models". *Stat* **2**, pp. 86–101.

Grasso, M. et al. (2014). "Profile monitoring via sensor fusion: the use of PCA methods for multi-channel data". *International Journal of Production Research* **52**.20, pp. 6110–6135.

Grenander, U. (1950). "Stochastic processes and statistical inference". *Arkiv för Matematik* **1**.3, pp. 195 –277.

Gupta, S. (2012). "A note on the asymptotic distribution of LASSO estimator for correlated data". *Sankhy: The Indian Journal of Statistics, Series A (2008-)* **74**.1, pp. 10–28.

Happ, C. & Greven, S. (2018). "Multivariate Functional Principal Component Analysis for Data Observed on Different (Dimensional) Domains". *Journal of the American Statistical Association* **113**.522, pp. 649–659.

Hastie, T. et al. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, New York.

Hastie, T. et al. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC. Chap. 11.

Horn, J. L. (1965). "A rationale and test for the number of factors in factor analysis". *Psychometrika* **30**, pp. 179–185.

Hsing, T. & Eubank, R. L. (2015). *Theoretical Foundations of Functional Data Analysis, with an Introduction to Linear Operators*. John Wiley and Sons, Ltd.

Huang, H. H. et al. (2010). "Design of a robust EMG sensing interface for pattern classification". *Journal of Neural Engineering* **7**.

Huang, J. & Zhang, T. (2010). "The benefit of group sparsity". *Ann. Statist.* **38**.4, pp. 1978–2004.

Jacques, J. & Preda, C. (2014). "Model-based clustering for multivariate functional data". *Computational Statistics & Data Analysis* **71**.C, pp. 92–106.

James, G. et al. (2013). *An Introduction to Statistical Learning*. Springer, New York.

James, G. et al. (2000). "Principal component models for sparse functional data". *Biometrika* **87**.3, pp. 587–602.

Jia, J. & Rohe, K. (2015). "Preconditioning the Lasso for sign consistency". *Electronic Journal of Statistics* **9**.1, pp. 1150 –1172.

Karhunen, K. (1946). "Zur Spektraltheorie stochastischer prozesse".

Kaul, A. (2014). "Lasso with long memory regression errors". *Journal of Statistical Planning and Inference* **153**, pp. 11–26.

Kohavi, R. (1995). "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection". *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI'95. Montreal, Quebec, Canada: Morgan Kaufmann Publishers Inc., 1137–1143.

Krstajic, D. et al. (2014). "Cross-validation pitfalls when selecting and assessing regression and classification models". *Journal of Chemoinformatics* **6**.1.

Li, C. et al. (2020). "Fast covariance estimation for multivariate data". *Stat* **9**.1.

Liu, Y et al. (2014). "Sparse generalized functional linear model for predicting remission status of depression patients". *Pacific Symposium on Biocomputing*, pp. 364–375.

Loève, M. (1946). "Fonctions aléatoires à décomposition orthogonale exponentielle". *La Revue Scientifique* **84**, pp. 159–162.

Lounici, K. et al. (2009). "Taking Advantage of Sparsity in Multi-Task Learning". *Proceedings of the 22nd Conference on Information Theory*.

Luo, X. et al. (2006). "Tuning Variable Selection Procedures by Adding Noise". *Technometrics* **48**.2, pp. 165–175.

Matsui, H. & Konishi, S. (2011). "Variable selection for functional regression models via the L1 regularization". *Computational Statistics & Data Analysis* **55**.12, pp. 3304–3310.

Medeiros, M. C. & Mendes, E. F. (2015). "ℓ_1-regularization of high-dimensional time-series models with non-Gaussian and heteroskedastic errors". *Journal of Econometrics* **191**.1, pp. 255–271.

Mee, R. W. et al. (2017). "Selecting an Orthogonal or Nonorthogonal Two-Level Design for Screening". *Technometrics* **59**.3, pp. 305–318.

Meier, L. et al. (2009). "HIGH-DIMENSIONAL ADDITIVE MODELING". *The Annals of Statistics* **37**.6B, pp. 3779–3821.

Meinshausen, N. (2007). "Relaxed Lasso". *Computational Statistics & Data Analysis* **52**, pp. 374–393.

Mercier, C. et al. (2006). "Mapping phantom movement representations in the motor cortex of amputees". *Brain* **129**.8, pp. 2202–2210.

Miller, A. J. (1990). "4". *Subset selection in regression*. Chapman and Hall.

Nie, Y. et al. (2018). "Supervised functional principal component analysis". *Statistics and Computing* **28**, pp. 713–723.

O'Rahilly, R. & Müller, F. (1983). *Basic Human Anatomy: A Regional Study of Human Structure*. Saunders, Philadelphia.

Pannu, J. & Billor, N. (2017). "Robust Group-lasso for Functional Regression Model". *Communications in Statistics - Simulation and Computation* **46**, pp. 3356–3374.

Paul, D. & Peng, J. (2009). "Consistency of restricted maximum likelihood estimators of principal components". *The Annals of Statistics* **37**.3, pp. 1229 –1271.

Paul, D. et al. (2008). "'Preconditioning' for feature selection and regression in high-dimensional problems". *Ann. Statist.* **36**.4, pp. 1595–1618.

Paynabar, K. et al. (2016). "A Change-Point Approach for Phase-I Analysis in Multivariate Profile Monitoring and Diagnosis". *Technometrics* **58**.2, pp. 191–204.

Pezzulli, S. & Silverman, B. (1993). "Some properties of smoothed principal components analysis for functional data". English. *Computational Statistics* **8**, pp. 1 –16.

Puntanen, S. & Styan, G. P. H. (1989). "The Equality of the Ordinary Least Squares Estimator and the Best Linear Unbiased Estimator". *The American Statistician* **43**.3, pp. 153–161. eprint: `https://www.tandfonline.com/doi/pdf/10.1080/00031305.1989.10475644`.

Ramsay, J. O. & Dalzell, C. J. (1991). "Some Tools for Functional Data Analysis". *Journal of the Royal Statistical Society. Series B (Methodological)* **53**.3, pp. 539–572.

Ramsay, J. & Silverman, B. (2005). *Functional Data Analysis*. Springer Series in Statistics. Springer.

Rao, C. R. (1958). "Some Statistical Methods for Comparison of Growth Curves". *Biometrics* **14**.1, pp. 1–17.

Raskutti, G. et al. (2010). "Restricted Eigenvalue Properties for Correlated Gaussian Designs". *Journal of Machine Learning Research* **11**, pp. 2241–2259.

Ravikumar, P. et al. (2009). "Sparse additive models". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **71**.5, pp. 1009–1030. eprint: `https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9868.2009.00718.x`.

Reed, M. & Simon, B. (1980). *Methods of Modern Mathematical Physics I: Functional Analysis*. San Diego, CA: Academic Press.

Reiss, P. T. & Ogden, R. T. (2007). "Functional Principal Component Regression and Functional Partial Least Squares". *Journal of the American Statistical Association* **102**.479, pp. 984–996.

Rice, J. & Wu, C. (2001). "Nonparametric mixed effects models for unequally sampled noisy curves". *Biometrics* **57**.1, pp. 253–9.

Rice, J. A. & Silverman, B. W. (1991). "Estimating the Mean and Covariance Structure Nonparametrically When the Data are Curves". *Journal of the Royal Statistical Society. Series B (Methodological)* **53**.1, pp. 233–243.

Roberts, D. et al. (2017). "Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure". *Ecography* **40**.8, pp. 913–929.

Scheme, E. & Englehart, K. (2011). "Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use". *Journal of Rehabilitation Research and Development* **48**, pp. 643–659.

Shen, H. & Huang, J. Z. (2008). "Sparse principal component analysis via regularized low rank matrix approximation". *Journal of Multivariate Analysis* **99**.6, pp. 1015 –1034.

Shi, M. et al. (1996). "An Analysis of Paediatric CD4 Counts for Acquired Immune Deficiency Syndrome Using Flexible Random Curves". *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **45**.2, pp. 151–163.

Simon, N. & Tibshirani, R. (2012). "STANDARDIZATION AND THE GROUP LASSO PENALTY". *Statistica Sinica* **22**.3, pp. 983–1001.

Smaga, łLukasz & Matsui, H. (2018). "A note on variable selection in functional regression via random subspace method". *Statistical Methods & Applications* **27**.3, pp. 455–477.

Stallrich, J. et al. (2020). "Optimal EMG placement for a robotic prosthesis controller with sequential, adaptive functional estimation (SAFE)". *Ann. Appl. Stat.* **14**.3, pp. 1164–1181.

Staniswalis, J. G. & Lee, J. J. (1998). "Nonparametric Regression Analysis of Longitudinal Data". *Journal of the American Statistical Association* **93**.444, pp. 1403–1418.

Su, Y. et al. (2017). "Hypothesis testing in functional linear models". *Biometrics* **73**.2, pp. 551–561.

Tibshirani, R. (1996). "Regression shrinkage and selection via the lasso". *Journal of the Royal Statistical Society, Series B: Methodological* **58**, pp. 267–288.

Tibshirani, R. et al. (2005). "Sparsity and smoothness via the fused LASSO". *Journal of the Royal Statistical Society Series B* **67**, pp. 91–108.

Tutz, G. & Gertheiss, J. (2010). "Feature Extraction in Signal Regression: A Boosting Technique for Functional Data Regression". *Journal of Computational and Graphical Statistics* **19**, pp. 154–174.

Uustal, H. (2020). *What Are the Different Types of Prostheses?*

Wainwright, M. J. (2009). "Sharp Thresholds for High-Dimensional and Noisy Sparsity Recovery Using ℓ_1 -Constrained Quadratic Programming (Lasso)". *IEEE Transactions on Information Theory* **55**.5, pp. 2183–2202.

Wang, H. et al. (2007). "Regression Coefficient and Autoregressive Order Shrinkage and Selection Via the Lasso". *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **69**.1, pp. 63–78.

Wang, K. & Tsung, F. (2020). "Hierarchical sparse functional principal component analysis for multistage multivariate profile data". *IISE Transactions* **0**.0, pp. 1–16.

Wong, K. C. et al. (2020). "Lasso guarantees for β-mixing heavy-tailed time series". *Ann. Statist.* **48**.2, pp. 1124–1142.

Wood, S. N. (2003). "Thin Plate Regression Splines". *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **65**.1, pp. 95–114.

Wu, Y. et al. (2007). "Controlling Variable Selection by the Addition of Pseudovariables". *Journal of the American Statistical Association* **102**.477, pp. 235–243.

Xiao, L. et al. (2016). "Fast Covariance Estimation for High-dimensional Functional Data". *Statistics and Computing* **26**.1, pp. 409–421.

Xiao, L. et al. (2018). "Fast covariance estimation for sparse functional data". *Statistics and Computing* **28**, pp. 511–522.

Yang, Y. & Zou, H. (2014). "A fast unified algorithm for solving group-lasso penalize learning problems". *Statistics and Computing* **25**.6, pp. 1129–1141.

— (2017). *gglasso: Group Lasso Penalized Learning Using a Unified BMD Algorithm*. R package version 1.4.

Yao, F. & Lee, T. (2006). "Penalized spline models for functional principal component analysis". *Journal of the Royal Statistical Society Series B* **68**, pp. 3–25.

Yao, F. et al. (2005). "Functional Data Analysis for Sparse Longitudinal Data". *Journal of the American Statistical Association* **100**.470, pp. 577–590.

Yau, C. Y. & Hui, T. S. (2017). "LARS-type algorithm for group lasso". *Statistics and Computing* **27**, pp. 1041–1048.

Yoon, Y. J. et al. (2013). "Penalized regression models with autoregressive error terms". *Journal of Statistical Computation and Simulation* **83**.9, pp. 1756–1772.

Yoon, Y. J. et al. (2017). "Adaptive LASSO for linear regression models with ARMA-GARCH errors". *Communications in Statistics - Simulation and Computation* **46**.5, pp. 3479–3490.

Yuan, M. & Lin, Y. (2006). "Model selection and estimation in regression with grouped variables". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**.1, pp. 49–67.

Zhang, C. et al. (2018). "Weakly correlated profile monitoring based on sparse multi-channel functional principal component analysis". *IISE Transactions* **50**.10, pp. 878–891.

Zhou, L. et al. (2008). "Joint modelling of paired sparse functional data using principal components". *Biometrika* **95**.3, pp. 601–619.

Zhou, P. et al. (2007). "Decoding a new neural-machine interface for control of artificial limbs". *Journal of Neuralphysiology* **98**, pp. 2974–2982.

Zhu, H. & Cox, D. (2009). "A Functional Generalized Linear Model with Curve Selection in Cervical Pre-cancer Diagnosis Using Fluorescence Spectroscopy". **57**, pp. 173–189.

Zou, H. (2006). "The Adaptive Lasso and Its Oracle Properties". *Journal of the American Statistical Association* **101**.476, pp. 1418–1429. eprint: `https://doi.org/10.1198/016214506000000735`.

Zou, H. & Hastie, T. (2005). "Regularization and variable selection via the elastic net". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**.2, pp. 301–320.

Zou, H. et al. (2006). "Sparse Principal Component Analysis". *Journal of Computational and Graphical Statistics* **15**.2, pp. 265–286.

APPENDICES

Appendix A

Additional details for Chapter 2

A.1 GLS Lasso

To the best of our knowledge, the existing literature on the GLS lasso centers $\boldsymbol{y}$ and $\boldsymbol{X}$ in the usual way. However, this would mean the GLS lasso would not converge to the unpenalized GLS estimates for small λ. We instead recommend first scaling the columns of $\boldsymbol{X}$ and define the GLS lasso objective function to be

$$\frac{1}{2N}(\boldsymbol{y} - \beta_0\mathbf{1} - \boldsymbol{X}\boldsymbol{\beta})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \beta_0\mathbf{1} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda||\boldsymbol{\beta}||_1 \ .$$

Taking the derivative with respect to β_0, which is not included in the penalty, we get the estimator: $\hat{\beta}_0 = \frac{\mathbf{1}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{y}}{\mathbf{1}^T\boldsymbol{\Sigma}^{-1}\mathbf{1}} - \frac{\mathbf{1}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X}\boldsymbol{\beta}}{\mathbf{1}^T\boldsymbol{\Sigma}^{-1}\mathbf{1}}$. Plugging this into the objective function, we can rewrite the GLS lasso estimator as

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2N}(\boldsymbol{y}^* - \boldsymbol{X}^*\boldsymbol{\beta})^T(\boldsymbol{y}^* - \boldsymbol{X}^*\boldsymbol{\beta}) + \lambda||\boldsymbol{\beta}||_1 \ ,$$

where $\boldsymbol{y}^* = (\boldsymbol{I} - \boldsymbol{P}_{\Sigma 1})\boldsymbol{\Sigma}^{-1/2}\boldsymbol{y}$, $\boldsymbol{X}^* = (\boldsymbol{I} - \boldsymbol{P}_{\Sigma 1})\boldsymbol{\Sigma}^{-1/2}\boldsymbol{X}$, and $\boldsymbol{P}_{\Sigma 1}$ is the orthogonal projector onto the column space of $\boldsymbol{\Sigma}^{-1/2}\mathbf{1}$. This may be minimized using common lasso optimization techniques. Note that if $\boldsymbol{\Sigma}$ has an eigenvector of $\mathbf{1}$ then $(\boldsymbol{I} - \boldsymbol{P}_{\Sigma 1}) = (\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T)$ will perform the traditional centering employed by the lasso. Moreover, when $\boldsymbol{\Sigma}$ has a compound symmetry structure the GLS and OLS lasso solutions are equivalent.

A.2 Lasso MSE for Orthogonal Designs

For $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}$, the lasso solution is

$$\widehat{\beta}_j = \text{sign}(\widehat{\beta}_j^{OLS})(|\widehat{\beta}_j^{OLS}| - \lambda)_+, \tag{A.1}$$

where $\widehat{\beta}_j^{OLS}$ is the jth estimated OLS coefficient, $\text{sign}(x) = 1$ or -1 if $x > 0$ or $x < 0$, respectively, and $(x)_+ = \max\{x, 0\}$. Let $\pi(\widehat{\beta}_j^{OLS})$ denote the marginal probability distribution function of $\widehat{\beta}_j^{OLS}$. The law of total expectation on the three disjoint events $\left\{|\widehat{\beta}_j^{OLS}| \leq \lambda\right\}$, $\left\{\widehat{\beta}_j^{OLS} > \lambda\right\}$, $\left\{\widehat{\beta}_j^{OLS} < -\lambda\right\}$ admits the expression

$$\mathbb{E}(\widehat{\beta}_j) = \int_{\lambda}^{\infty} (\widehat{\beta}_j^{OLS} - \lambda)\pi(\widehat{\beta}_j^{OLS})d\widehat{\beta}_j^{OLS} + \int_{-\infty}^{-\lambda} (\widehat{\beta}_j^{OLS} + \lambda)\pi(\widehat{\beta}_j^{OLS})d\widehat{\beta}_j^{OLS}$$

which follows from properties of truncated probability distributions. Adding and subtracting $\int_{-\lambda}^{\lambda} \widehat{\beta}_j^{OLS}\pi(\widehat{\beta}_j^{OLS})d\widehat{\beta}_j^{OLS}$ to this expression gives

$$\mathbb{E}(\widehat{\beta}_j) = \beta_j - \int_{-\lambda}^{\lambda} \widehat{\beta}_j^{OLS}\pi(\widehat{\beta}_j^{OLS})d\widehat{\beta}_j^{OLS} - \lambda\left[P(\widehat{\beta}_j^{OLS} > \lambda) - P(\widehat{\beta}_j^{OLS} < -\lambda)\right] ,$$

when $\mathbb{E}(\widehat{\beta}_j^{OLS}) = \beta_j$. The same technique can be applied to derive $\text{Var}(\widehat{\beta}_j) = \mathbb{E}(\widehat{\beta}_j^2) - \mathbb{E}(\widehat{\beta}_j)^2$. With $\mathbb{E}(x, a, b) = \int_a^b x\,\pi(\widehat{\beta}_j^{OLS})d\widehat{\beta}_j^{OLS}$ and $\mathbb{E}(x, a) = \int_{-a}^{a} x\,\pi(\widehat{\beta}_j^{OLS})d\widehat{\beta}_j^{OLS}$, we have

$$\begin{aligned}
\text{Var}(\widehat{\beta}_j) = \text{Var}(\widehat{\beta}_j^{OLS}) &- \left[\mathbb{E}\left((\widehat{\beta}_j^{OLS})^2, \lambda\right) - \mathbb{E}\left(\widehat{\beta}_j^{OLS}, \lambda\right)^2\right] \\
&+ \left[\beta_j - \mathbb{E}\left(\widehat{\beta}_j^{OLS}, \lambda\right)\right]\left[2\mathbb{E}\left(\widehat{\beta}_j^{OLS}, \lambda\right) + 2\lambda\left(P(\widehat{\beta}_j^{OLS} > \lambda) - P(\widehat{\beta}_j^{OLS} < -\lambda)\right)\right] \\
&- \lambda\left(\mathbb{E}(\widehat{\beta}_j^{OLS}, \lambda, \infty) - \mathbb{E}(\widehat{\beta}_j^{OLS}, -\infty, -\lambda)\right) \\
&+ \lambda^2\left(1 - P(|\widehat{\beta}_j^{OLS}| < \lambda) - (P(\widehat{\beta}_j^{OLS} > \lambda) - P(\widehat{\beta}_j^{OLS} < -\lambda))^2\right) .
\end{aligned}$$

Finally, the MSE for $\widehat{\boldsymbol{\beta}}$ is found by taking the sum of the individual MSE's:

$$MSE = \sum_j \left\{\text{Var}(\widehat{\beta}_j) + \left(\mathbb{E}(\widehat{\beta}_j^{OLS}, \lambda) + \lambda\left[P(\widehat{\beta}_j^{OLS} > \lambda) - P(\widehat{\beta}_j^{OLS} < -\lambda)\right]\right)^2\right\} .$$

A.3 Group Lasso Irrepresentable Conditions

Recall the definition of the group lasso estimator

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2N}||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \sum_{j=1}^{m} \lambda_j ||\boldsymbol{\beta}_j||_2 \ . \tag{A.2}$$

Let $S = \{j \ : \ \boldsymbol{\beta}_j \neq 0\}$ now denote the groups having at least one non-zero coefficient, making $\boldsymbol{y} = \sum_{j \in S} \boldsymbol{X}_j \boldsymbol{\beta}_j + e$. For the estimator to recover the true support, it must satisfy the following conditions

$$\frac{1}{N}\boldsymbol{X}_j^T \left(\sum_{j' \in S} \boldsymbol{X}_{j'} \hat{\boldsymbol{\beta}}_{j'} \right) - \frac{1}{N}\boldsymbol{X}_j^T \boldsymbol{y} + \hat{\boldsymbol{z}}_j = 0 \ , \tag{A.3}$$

where $\hat{\boldsymbol{z}}_j = \lambda_j \hat{\boldsymbol{\beta}}_j / ||\hat{\boldsymbol{\beta}}_j||_2$ when $\hat{\boldsymbol{\beta}}_j \neq \boldsymbol{0}$ and otherwise $||\hat{\boldsymbol{z}}_j||_2 \leq \lambda_j$. Let $\boldsymbol{X}_S$ be the submatrix comprised of those $\boldsymbol{X}_j$ where $j \in S$. Plugging in $\boldsymbol{y} = \sum_{j \in S} \boldsymbol{X}_j \boldsymbol{\beta}_j + \boldsymbol{e}$ into (A.3), we arrive at the expression for the $\hat{\boldsymbol{\beta}}_S$

$$(\hat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S) = (\frac{1}{N}\boldsymbol{X}_S^T \boldsymbol{X}_S)^{-1} \left[\frac{1}{N}\boldsymbol{X}_S^T \boldsymbol{e} - \hat{\boldsymbol{z}}_S \right] \ , \tag{A.4}$$

where $\hat{\boldsymbol{z}}_S$ is the vector of corresponding $\hat{\boldsymbol{z}}_j$. For $\hat{\boldsymbol{\beta}}_{S^c}$ to be shrunk to $\boldsymbol{0}$, each corresponding $\hat{\boldsymbol{z}}_j$ must satisfy

$$\hat{\boldsymbol{z}}_j = \frac{1}{N}\boldsymbol{X}_j^T (\boldsymbol{y} - \boldsymbol{X}_S \hat{\boldsymbol{\beta}}_S) \tag{A.5}$$

$$\Leftrightarrow \ \hat{\boldsymbol{z}}_j = \frac{1}{N}\boldsymbol{X}_j^T (\boldsymbol{X}_S(\boldsymbol{\beta}_S - \hat{\boldsymbol{\beta}}_S) + \boldsymbol{e}) \tag{A.6}$$

$$\Leftrightarrow \hat{\boldsymbol{z}}_j = \boldsymbol{X}_j^T \boldsymbol{X}_S (\boldsymbol{X}_S^T \boldsymbol{X}_S)^{-1} \hat{\boldsymbol{z}}_S + \frac{1}{N}\boldsymbol{X}_j^T (\boldsymbol{I} - \boldsymbol{P}_S)\boldsymbol{e} \ , \tag{A.7}$$

where $\boldsymbol{P}_S = \boldsymbol{X}_S (\boldsymbol{X}_S^T \boldsymbol{X}_S)^{-1} \boldsymbol{X}_S^T$, and $||\hat{\boldsymbol{z}}_j||_2 \leq \lambda_j$. Applying the triangle inequality to (A.7), and then again to $||\boldsymbol{X}_j^T \boldsymbol{X}_S (\boldsymbol{X}_S^T \boldsymbol{X}_S)^{-1} \hat{\boldsymbol{z}}_S||_2$, we replace $||\hat{\boldsymbol{z}}_j||_2 \leq \lambda_j$ with the more stringent condition

$$||\frac{1}{N}\boldsymbol{X}_j^T (\boldsymbol{I} - \boldsymbol{P}_S)\boldsymbol{e}||_2 + ||\boldsymbol{X}_j^T \boldsymbol{X}_S (\boldsymbol{X}_S^T \boldsymbol{X}_S)^{-1} \hat{\boldsymbol{z}}_S||_2 \leq \lambda_j \tag{A.8}$$

$$\Leftrightarrow \frac{1}{N\lambda_j}||\boldsymbol{X}_j^T (\boldsymbol{I} - \boldsymbol{P}_S)\boldsymbol{e}||_2 \leq 1 - \frac{1}{\lambda_j}||\boldsymbol{X}_j^T \boldsymbol{X}_S (\boldsymbol{X}_S^T \boldsymbol{X}_S)^{-1} \hat{\boldsymbol{z}}_S||_2 \tag{A.9}$$

The random variable on the left-hand side of (A.9) is nonnegative so the right-hand side must be positive for this more stringent event to be feasible. Note this condition's similarity to (2.2) for the lasso, but for the lasso each element of $\hat{\boldsymbol{z}}_S = \pm\lambda$. For group lasso, $||\hat{\boldsymbol{z}}_j||_2 = \lambda_j = \lambda\sqrt{k_j}$. We desire a diagnostic around this event that does not depend on the specific $\hat{\boldsymbol{z}}_S$ and do so using an eigenvalue bound:

$$||\boldsymbol{X}_j^T\boldsymbol{X}_S(\boldsymbol{X}_S^T\boldsymbol{X}_S)^{-1}\hat{\boldsymbol{z}}_S||_2^2 \leq \Lambda_{\max}(\boldsymbol{X}_j^T\boldsymbol{X}_S(\boldsymbol{X}_S^T\boldsymbol{X}_S)^{-2}\boldsymbol{X}_S^T\boldsymbol{X}_j)||\hat{\boldsymbol{z}}_S||_2^2 \tag{A.10}$$

$$= \Lambda_{\max}(\boldsymbol{X}_j^T\boldsymbol{X}_S(\boldsymbol{X}_S^T\boldsymbol{X}_S)^{-2}\boldsymbol{X}_S^T\boldsymbol{X}_j)(\lambda^2\sum_{j\in S}k_j)\ . \tag{A.11}$$

Giving the stronger condition

$$\sqrt{\Lambda_{\max}(\boldsymbol{X}_j^T\boldsymbol{X}_S(\boldsymbol{X}_S^T\boldsymbol{X}_S)^{-2}\boldsymbol{X}_S^T\boldsymbol{X}_j)} < \frac{\sqrt{k_j}}{\sqrt{\sum_{j\in S}k_j}}\ . \tag{A.12}$$

This would need to be checked for each $j \in S^c$.

A.4 Group Lasso Deterministic $\mathcal{L}_2$ Bound

Assume a positive group lasso restricted eigenvalue, κ, and $\lambda_j \geq 2||\boldsymbol{X}_j^T\boldsymbol{e}||_2/N$. By definition of the group lasso estimator, we have

$$\frac{1}{2N}||\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}||_2^2 + \sum_{j=1}^m \lambda_j||\hat{\boldsymbol{\beta}}_j||_2 \leq \frac{1}{2N}||\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}^*||_2^2 + \sum_{j=1}^m \lambda_j||\boldsymbol{\beta}_j^*||_2\ . \tag{A.13}$$

Plugging in $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}^* + \boldsymbol{e}$ and simplifying gives

$$\frac{1}{2N}||\boldsymbol{X}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*)||_2^2 \leq \sum_j \lambda_j(||\boldsymbol{\beta}_j^*||_2 - ||\hat{\boldsymbol{\beta}}_j||_2) + \frac{1}{N}\boldsymbol{e}^T\boldsymbol{X}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*)\ . \tag{A.14}$$

By the Cauchy-Schwarz inequality,

$$\frac{1}{N}\boldsymbol{e}^T\boldsymbol{X}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*) = \frac{1}{N}\sum_j \boldsymbol{e}^T\boldsymbol{X}_j(\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_j^*) \leq \frac{1}{N}\sum_j ||\boldsymbol{X}_j^T\boldsymbol{e}||_2||(\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_j^*)||_2 \tag{A.15}$$

$$\leq \frac{1}{2}\sum_j \lambda_j||(\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_j^*)||_2\ , \tag{A.16}$$

leading to the bound

$$\frac{1}{2N}||\boldsymbol{X}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*)||_2^2+\frac{1}{2}\sum_j \lambda_j||(\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_j^*)||_2 \leq \sum_j \lambda_j(||\boldsymbol{\beta}_j^*||_2-||\hat{\boldsymbol{\beta}}_j||_2+||\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_j^*||_2) \quad \text{(A.17)}$$

$$\leq 2\sum_j \lambda_j \min(||\boldsymbol{\beta}_j^*||_2, ||\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_j^*||_2) \quad \text{(A.18)}$$

$$\leq 2\sum_{j\in S} \lambda_j||\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_j^*||_2 \ , \quad \text{(A.19)}$$

where (A.18) follows from two applications of the reverse triangle inequality and (A.19) follows from $||\boldsymbol{\beta}_j^*||_2=0$ for $j\in S^c$. This series of inequalities establishes the two inequalities

$$\frac{1}{N}||\boldsymbol{X}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*)||_2^2 \leq 4\sum_{j\in S}\lambda_j||\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_j^*||_2 \leq 4\sqrt{\sum_{j\in S}\lambda_j^2}\ ||\hat{\boldsymbol{\beta}}_S-\boldsymbol{\beta}_S^*||_2 \quad \text{(A.20)}$$

$$\sum_j \lambda_j||\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_j^*||_2 \leq 4\sum_{j\in S}\lambda_j||\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_j^*||_2 \leq 4\sqrt{\sum_{j\in S}\lambda_j^2}\ ||\hat{\boldsymbol{\beta}}_S-\boldsymbol{\beta}_S^*||_2 \ . \quad \text{(A.21)}$$

Inequality (A.21) implies the group lasso error vector $\boldsymbol{\nu}=\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*$ lies in the set defined by the group lasso restricted eigenvalue condition and by our initial assumptions

$$||\hat{\boldsymbol{\beta}}_S-\boldsymbol{\beta}_S^*||_2 \leq \frac{||\boldsymbol{X}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*)||_2}{\sqrt{N}\kappa} \ . \quad \text{(A.22)}$$

Using this bound in (A.20) gives

$$\frac{1}{\sqrt{N}}||\boldsymbol{X}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*)||_2 \leq \frac{4}{\kappa}\sqrt{\sum_{j\in S}\lambda_j^2} \Rightarrow \frac{1}{N}||\boldsymbol{X}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*)||_2^2 \leq \frac{16}{\kappa^2}\sum_{j\in S}\lambda_j^2 \ . \quad \text{(A.23)}$$

Using the restricted eigenvalue bound in (A.21), and noting that $\sum_j ||\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_j^*||_2 \leq \frac{1}{\lambda_{\min}}\sum_j \lambda_j||\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_j^*||_2$ gives

$$\sum_j ||\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_j^*||_2 \leq \frac{16}{\kappa^2}\sum_{j\in S}\frac{\lambda_j^2}{\lambda_{\min}} \ . \quad \text{(A.24)}$$

The tightest upper bound replaces each λ_j with their lower bound giving

$$\sum_j ||\hat{\boldsymbol{\beta}}_j-\boldsymbol{\beta}_j^*||_2 \leq \frac{16}{\kappa^2}\sum_{j\in S}\frac{4||\boldsymbol{X}_j^T\boldsymbol{e}||_2^2/N^2}{\min 2||\boldsymbol{X}_j^T\boldsymbol{e}||_2/N} = \frac{32}{N\kappa^2}\sum_{j\in S}\frac{||\boldsymbol{X}_j^T\boldsymbol{e}||_2^2}{\min||\boldsymbol{X}_j^T\boldsymbol{e}||_2} \ . \quad \text{(A.25)}$$

Appendix B

Additional details for Chapter 3

B.1 GMD Update Expression

Define

$$Q = L(\boldsymbol{\beta}^t|\boldsymbol{D}) - (\boldsymbol{\beta}_k - \boldsymbol{\beta}_k^t)^T U_k + \frac{1}{2}\gamma_k(\boldsymbol{\beta}_k - \boldsymbol{\beta}_k^t)^T(\boldsymbol{\beta}_k - \boldsymbol{\beta}_k^t) + \lambda||\boldsymbol{\beta}_k||_2$$

Notice that the solution that minimizes Q is equivalent to the generalized gradient update,

$$\boldsymbol{\beta}^{t+1} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\left\{ g(\boldsymbol{\beta}^t) + \langle \nabla g(\boldsymbol{\beta}^t), \boldsymbol{\beta} - \boldsymbol{\beta}^t\rangle + \frac{1}{2s^t}||\boldsymbol{\beta} - \boldsymbol{\beta}^t||^2 + h(\boldsymbol{\beta})\right\}, \tag{B.1}$$

with positive step size $s^t = 1/\gamma_k$, loss function $g(\boldsymbol{\beta}^t)$ equal to the least squares loss, and penalty function $h(\boldsymbol{\beta}) = \lambda||\boldsymbol{\beta}||$. To minimize the right side of (B.1), we take the first order derivative with respect to a given group k and set it to 0:

$$\boldsymbol{U}_k + \frac{1}{s^t}(\boldsymbol{\beta}_k - \boldsymbol{\beta}_k^t) + \lambda u(\boldsymbol{\beta}_k) = 0 \Leftrightarrow s^t\boldsymbol{U}_k + \boldsymbol{\beta}_k - \boldsymbol{\beta}_k^t + s^t\lambda u(\boldsymbol{\beta}_k) = 0\ ,$$

where $u(\boldsymbol{\beta}_k)$ is the subgradient with values:

$$u(\boldsymbol{\beta}_k) = \begin{cases} \boldsymbol{\beta}_k/||\boldsymbol{\beta}_k||_2 & \boldsymbol{\beta}_k \neq 0 \\ \boldsymbol{u}\ s.t.\ ||\boldsymbol{u}||_2 \leq 1 & \boldsymbol{\beta}_k = 0\ . \end{cases}$$

Solving for $\boldsymbol{\beta}_k$ yields

$$\boldsymbol{\beta}_k = \boldsymbol{\beta}_k^t - s^tU_k - s^t\lambda u(\boldsymbol{\beta}_k) = s^t\lambda\left\{\frac{1}{s^t\lambda}(\boldsymbol{\beta}_k^t - s^tU_k) - u(\boldsymbol{\beta}_k)\right\}.$$

If $\frac{1}{s^t\lambda}(\boldsymbol{\beta}_k^t - s^t U_k) = u(\boldsymbol{\beta}_k) \Rightarrow \frac{1}{s^t\lambda}||\boldsymbol{\beta}_k^t - s^t U_k||_2 = ||u(\boldsymbol{\beta}_k)||_2 \leq 1$, then $\boldsymbol{\beta}_k^{t+1} = \mathbf{0}$. So if $||\boldsymbol{\beta}_k^t - s^t U_k||_2 \leq s^t\lambda$, then $\boldsymbol{\beta}_k^{t+1} = 0$. Otherwise, we have

$$\begin{aligned} & s^t \boldsymbol{U}_k + \boldsymbol{\beta}_k - \boldsymbol{\beta}_k^t + s^t\lambda \frac{\boldsymbol{\beta}_k}{||\boldsymbol{\beta}_k||_2} = 0 \\ \Leftrightarrow \quad & \boldsymbol{\beta}_k^{t+1}\left(1 + \frac{s^t\lambda}{||\boldsymbol{\beta}_k^{t+1}||_2}\right) = \boldsymbol{\beta}_k^t - s^t\boldsymbol{U}_k \ . \end{aligned}$$

Taking the Euclidean norm of both sides and solving for $||\boldsymbol{\beta}_k^{t+1}||_2$ yields

$$\begin{aligned} ||\boldsymbol{\beta}_k^{t+1}||_2\left(1 + \frac{s^t\lambda}{||\boldsymbol{\beta}_k||_2}\right) &= ||\boldsymbol{\beta}_k^t - s^t U_k||_2 \\ ||\boldsymbol{\beta}_k^{t+1}||_2 + s^t\lambda &= ||\boldsymbol{\beta}_k^t - s^t U_k||_2 \\ ||\boldsymbol{\beta}_k^{t+1}||_2 &= ||\boldsymbol{\beta}_k^t - s^t\boldsymbol{U}_k||_2 - s^t\lambda. \end{aligned}$$

By substitution,

$$\begin{aligned} \boldsymbol{\beta}_k^{t+1}\left(1 + \frac{s^t\lambda}{||\boldsymbol{\beta}_k^t - s^t U_k||_2 - s^t\lambda}\right) &= \boldsymbol{\beta}_k^t - s^t U_k \\ \boldsymbol{\beta}_k^{t+1} &= \frac{\boldsymbol{\beta}_k^t - s^t U_k||_2 - s^t\lambda}{||\boldsymbol{\beta}_k^t - s^t U_k||_2}(\boldsymbol{\beta}_k^t - s^t U_k) \\ &= \left(1 - \frac{s^t\lambda}{||\boldsymbol{\beta}_k^t - s^t U_k||_2}\right)(\boldsymbol{\beta}_k^t - s^t U_k). \end{aligned}$$

Thus, using both KKT conditions, the update equation is given as

$$\boldsymbol{\beta}_k^{t+1} = \left(1 - \frac{s^t\lambda}{||\boldsymbol{\beta}_k^t - s^t U_k||_2}\right)_+ (\boldsymbol{\beta}_k^t - s^t U_k).$$

One thing to point out though is that the notation assumes $||\boldsymbol{\beta}_k^t - s^t\boldsymbol{U}_k||_2 > 0$, which is not necessarily guaranteed (although it likely holds with probability 1). Using step size $s^t = 1/\gamma_k$, we arrive at the update equation given by Yang & Zou [2014]:

$$\boldsymbol{\beta}_k^{t+1} = \frac{1}{\gamma_k}(U_k + \gamma_k\boldsymbol{\beta}_k^t)\left(1 - \frac{\lambda}{||U_k + \gamma_k\boldsymbol{\beta}_k^t||}\right)_+ .$$

Appendix C

Additional details for Chapter 4

C.1 Update equation for $\boldsymbol{\alpha}$

When $\boldsymbol{\beta}$ is fixed, (4.18) is reduced to solving for $\boldsymbol{\alpha}$ under the constraint that $\boldsymbol{\alpha}^T\boldsymbol{\alpha} = 1$ with

$$\widehat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}}\{||\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{\beta}||^2 - ||\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha}||^2\} \tag{C.1}$$

$$= \arg\min_{\boldsymbol{\alpha}}\{||\boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\beta}^T\boldsymbol{\beta} - \boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha}\} \tag{C.2}$$

$$= \arg\min_{\boldsymbol{\alpha}} -2\boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}. \tag{C.3}$$

By the Cauchy-Schwarz inequality, $\boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} = ||\boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}|| \leq ||\boldsymbol{\alpha}|| \cdot ||\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}||$, and equality holds when $\boldsymbol{\alpha} \propto \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}$. So the updating equation is $\widehat{\boldsymbol{\alpha}} = \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}/||\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}||$ to enforce the constraint $\boldsymbol{\alpha}^T\boldsymbol{\alpha} = 1$.

C.2 PC regression equivalence

First, notice that the first K principal components can be rewritten as

$$\begin{aligned}\boldsymbol{W}_K &= \boldsymbol{X}\boldsymbol{V}_K = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T\boldsymbol{V}_K \\ &= \boldsymbol{U}\boldsymbol{D}\begin{bmatrix}\boldsymbol{I}_K \\ \boldsymbol{0}_{p-K}\end{bmatrix} = \boldsymbol{U}\begin{bmatrix}\boldsymbol{D}_K \\ \boldsymbol{0}_{p-K}\end{bmatrix} \\ &= \boldsymbol{U}_K\boldsymbol{D}_K.\end{aligned}$$

Then the estimated coefficient is

$$\begin{aligned}\widehat{\boldsymbol{b}} &= (\boldsymbol{W}_K^T\boldsymbol{W}_K)^{-1}\boldsymbol{W}_K^T\boldsymbol{Y} \\ &= (\boldsymbol{D}_K\boldsymbol{U}_K^T\boldsymbol{U}_K\boldsymbol{D}_K)^{-1}\boldsymbol{D}_K\boldsymbol{U}_K^T\boldsymbol{Y} \\ &= \boldsymbol{D}_K^{-1}\boldsymbol{U}_K^T\boldsymbol{Y},\end{aligned}$$

and the estimated response is $\widehat{\boldsymbol{Y}} = \boldsymbol{U}_K\boldsymbol{U}_K^T\boldsymbol{Y}$. Using the reconstructed matrix $\widetilde{\boldsymbol{X}} = \boldsymbol{U}_K\boldsymbol{D}_K\boldsymbol{V}_K^T$ with rank $K < p$, the estimated coefficient is $\widehat{\boldsymbol{\beta}} = (\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}})^+\widetilde{\boldsymbol{X}}\boldsymbol{Y}$, where $\boldsymbol{A}^+$ is the Moore-Penrose inverse of $\boldsymbol{A}$. Although $\widehat{\boldsymbol{\beta}}$ is not unique, the predicted response is unique, as the following equation shows.

$$\begin{aligned}\widehat{\boldsymbol{Y}} &= \widetilde{\boldsymbol{X}}\widehat{\boldsymbol{\beta}} \\ &= \widetilde{\boldsymbol{X}}(\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}})^+\widetilde{\boldsymbol{X}}\boldsymbol{Y} \\ &= \boldsymbol{U}_K\boldsymbol{D}_K\boldsymbol{V}_K^T(\boldsymbol{V}_K\boldsymbol{D}_K\boldsymbol{U}_K^T\boldsymbol{U}_K\boldsymbol{D}_K\boldsymbol{V}_K^T)^+\boldsymbol{V}_K\boldsymbol{D}_K\boldsymbol{U}_K^T\boldsymbol{Y} \\ &= \boldsymbol{U}_K\boldsymbol{D}_K\boldsymbol{V}_K^T(\boldsymbol{V}_K\boldsymbol{D}_K^2\boldsymbol{V}_K^T)^+\boldsymbol{V}_K\boldsymbol{D}_K\boldsymbol{U}_K^T\boldsymbol{Y} \\ &= \boldsymbol{U}_K\boldsymbol{D}_K\boldsymbol{V}_K^T(\boldsymbol{V}_K\boldsymbol{D}_K^{-2}\boldsymbol{V}_K^T)\boldsymbol{V}_K\boldsymbol{D}_K\boldsymbol{U}_K^T\boldsymbol{Y} \\ &= \boldsymbol{U}_K\boldsymbol{U}_K^T\boldsymbol{Y},\end{aligned}$$

where the second to last equality holds because $\boldsymbol{V}_K$ is orthogonal. Hence, not only is $\widehat{\boldsymbol{Y}}$ unique, but it is equivalent to that from PC regression.

C.3 Predictor reconstruction for SMFPCA

For SMFPCA by Zhang et al. [2018], we use the notation from their paper. As output of SMFPCA, we have $K < m$ FPCs in the $n \times K$ matrix $\boldsymbol{V} = [\boldsymbol{v}_1, ..., \boldsymbol{v}_K]$ with associated scores $\boldsymbol{\xi}_i^k$. This yields the approximate signals $x_{ij}(t) \approx \sum_k \xi_{ij}^k v^k(t)$.

For a univariate effect, $\gamma_j(t) \approx \sum_\ell \beta_{j\ell}\omega_\ell(t)$, write the FLR model as

$$\sum_j \int x_{ij}(t)\gamma_j(t)dt \approx \sum_j \int \left(\sum_k \xi_{ij}^k v^k(t)\right)\left(\sum_\ell \beta_{j\ell}\omega_\ell(t)\right) dt \tag{C.4}$$

$$= \sum_j \sum_k \sum_\ell \xi_{ij}^k \beta_{j\ell} \int v^k(t)\omega_{j\ell}(t)dt \tag{C.5}$$

$$= \sum_j \sum_\ell \beta_{j\ell}\boldsymbol{\xi}_{ij}^T(\Delta_t \boldsymbol{V}^T\boldsymbol{\omega}_\ell) \tag{C.6}$$

$$= \sum_j \Delta_t \boldsymbol{\xi}_{ij}^T \boldsymbol{V}^T\boldsymbol{\omega}\boldsymbol{\beta}_j, \tag{C.7}$$

where $\boldsymbol{\xi}_{ij}^T = (\xi_{ij}^1, ..., \xi_{ij}^k)$. Then the loss function is written as

$$||\boldsymbol{y} - \sum_j \Delta_t \boldsymbol{\Xi}_j \boldsymbol{V}^t\boldsymbol{\omega}\boldsymbol{\beta}_j||^2 = ||\boldsymbol{y} - \sum_j \boldsymbol{W}_j\boldsymbol{\beta}_j||,$$

where $\boldsymbol{\Xi}_j \in \mathbb{R}^{N\times K}$ row-stacks the N vectors $\boldsymbol{\xi}_{ij}$ and $\boldsymbol{W}_j = \Delta_t\boldsymbol{\Xi}_j\boldsymbol{V}^T\boldsymbol{\omega}$.

Then, for bivariate functional effects with the same approximation as before, we have

$$\sum_j \int x_{ij}(t)\gamma_j(t, z_i)dt \approx \sum_j \int \left(\sum_k \xi_{ij}^k v^k(t)\right)\left(\sum_\ell \sum_r \beta_{j\ell}\omega_\ell(t)\tau_r(z_i)\right) dt \tag{C.8}$$

$$= \sum_j \sum_k \sum_\ell \sum_r \xi_{ij}^k \beta_{j\ell r} \int v^k(t)\omega_\ell(t)\tau_r(z_i)dt \tag{C.9}$$

$$\approx \sum_j \{\Delta_t \boldsymbol{\xi}_{ij}^T \boldsymbol{V}^T\boldsymbol{\omega} \otimes \boldsymbol{\tau}^T(z_i)\}\boldsymbol{\beta}_j. \tag{C.10}$$

The matrix $\boldsymbol{W}_j$ is formed by row-stacking the vectors $\boldsymbol{W}_{ij}^T = \Delta_t\boldsymbol{\xi}_{ij}^T\boldsymbol{V}^T\boldsymbol{\omega} \otimes \boldsymbol{\tau}^T(z_i)$.