ABSTRACT

HOLTER, JULIA CAROL. New Approaches to Tuning Parameter Selection in Penalized Estimation. (Under the direction of Jonathan Stallrich).

The tuning parameter selection strategy for penalized estimation is crucial to identify a model that is both interpretable and predictive. For some penalties, this can mean that tuning parameter selection plays a vital role in variable selection. Popular strategies can struggle to balance model fit with model complexity, especially for penalties such as the Lasso, that tend to include many inactive coefficients before active coefficients can achieve their full magnitude. Often, this challenge results in selecting models with more predictors than necessary.

This dissertation proposes a simple, yet powerful K-fold Cross Validation strategy based on maximizing squared correlations between the observed and predicted values. The strategy can be applied to all penalized least-squares estimators and, under certain conditions, the metric informs a multiplicative bias adjustment. The bias correction can be extended to GLMs and used as a more flexible K-fold Cross Validation tool as well. An alternative penalized correction is also developed that reduces the bias of estimates more reliably than the original correction. The efficacy of the new approaches are demonstrated through simulation studies, on a functional variable selection problem regarding the placement of surface electromyogram sensors to control a robotic hand prosthesis and on a classification problem involving Turkish raisins.

The computational challenges of K-fold Cross Validation to choose tuning parameters are a motivating concern throughout this dissertation, so we explore a sequential optimization approach to approximating K-fold Cross Validation results. This application of a well known design tool can reduce the number of calculations needed in K-fold Cross Validation and allow practitioners to more thoroughly explore the tuning parameter space.

The new method's effectiveness is evaluated using a simulation study for functional data analysis and Poisson regression, and it is then applied to the EMG data problem.

New Approaches to Tuning Parameter Selection in Penalized Estimation

by
Julia Carol Holter

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
2022

APPROVED BY:

Ana-Maria Staicu

Arnab Maity

He Huang

Jonathan Stallrich
Chair of Advisory Committee

BIOGRAPHY

Julia Carol Holter grew up outside of Rochester, NY. After graduating Victor High School in 2013, she attended St. Lawrence University where she majored in Mathematics and minored in English Literature. It was at St. Lawrence that she discovered an interest in (and love of) the field of Statistics. She graduated college in 2017 and began her graduate studies in Statistics the following Fall, earning her Masters degree in the Spring of 2019 and her PhD in the Summer of 2022.

ACKNOWLEDGEMENTS

Above all, I would like to thank my indefatigable advisor, Dr. Jonathan Stallrich. Through your thoughtful guidance and enthusiastic curiosity, you have made me a better statistician and given me the tools I needed to do work that I am proud of. Thank you as well to the other members of my committee: Dr. Ana-Maria Staicu, Dr. Arnab Maity, and Dr. Helen Huang for your invaluable feedback.

Thank you to the NC State Department of Statistics community as a whole. I would particularly like to thank the members of my lab, who have read pretty much every version of the ideas in this dissertation: Nic Larsen, Michael McKibben, Ethan Davis, Kasia Dobrzycka, Kade Young, Adam Bolton, and Rebecca North. I also want to thank all the other wonderful friends I made in classrooms and SAS Commons, especially Laura Wendelberger, Chelsea Robalino, Annie Tang, and Marschall Furman.

Thank you to my emotional support task-force for making Monday nights a delight: Meaghan Demers-Peel, Sarah McEachern, Thressa Zimmerman, and Alice Antoniou. You can't make old friends, so I'm lucky that you're mine.

And, of course, thank you to Alex Long (for everything).

Finally, thank you to my exceptional family. To two of my favorite people in the world: my sister Tara, and my brother Kyle. To my dad, for his early-rising, nose-to-the-grindstone, let's-get-in-a-workout mentality. Last but certainly not least, thank you to my mom, who insisted I take AP Calculus (despite vehement protest) because she usually knows me better than I know myself. Love, love you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1 INTRODUCTION

1.1 Motivation and Outline

Penalized estimation is a statistical tool that uses a penalty on maximum likelihood to shrink estimates. Depending on the penalty, estimates can sometimes be shrunk to exactly zero, thereby performing variable selection simultaneously with parameter estimation. There are many reasons penalized estimates may be preferred to unpenalized, often relating to two desirable qualities: prediction accuracy and interpretability. While the bias of maximum likelihood estimates is low, they tend to have high variance and imposing a penalty can improve prediction accuracy. Furthermore, a goal of analysis is often to determine a small set of predictors that exhibit the strongest effects, a practice that can be made simpler through the use of penalized estimation (Tibshirani 1996).

To emphasize the practical importance of penalized estimation and variable selection, consider the recent work of Stallrich et al. (2020) who proposed a decoder for a prosethesis controller (PC) for a robotic hand. Electromyogram (EMG) signals taken from surface sensors placed on the residual muscles are input into the PC and translated into movement of the robotic limb by the decoder. Biomechanical models suggest few muscles are needed to inform certain movements, implying a PC can reliably predict movement with few strategically-placed EMG sensors. Modeling movement using as few EMG sensors as possible will simultaneously improve prediction accuracy and make the device more comfortable. The specifics of this application will be described in more detail in Section 1.3.

There are a few specific models of particular interest to this work. Chapter 2 primarily concerns problems that are well approximated by a sparse linear model:

$$\mathbf{y}_{n\times 1} = \mathbf{X}_{n\times p}\boldsymbol{\beta}^*_{p\times 1} + \boldsymbol{\epsilon}_{n\times 1} \tag{1.1}$$

where $E[\boldsymbol{\epsilon}] = \mathbf{0}_n$, $V[\boldsymbol{\epsilon}] = \boldsymbol{\Sigma}$, $\boldsymbol{\beta}^{*T} = [\beta^*_1, ..., \beta^*_{p^*}, 0, ..., 0]$, and p^* is the number of active covariates. Assume that both $\mathbf{y}$ and $\mathbf{X}$ are centered to render an intercept term unnecessary, and $\mathbf{X}$ is scaled such that the diagonals of $\mathbf{X}^T\mathbf{X}$ are equal to n. In Chapter 3, some of the discussion includes a class of model known as Generalized Linear Models (GLMs). In a GLM, the expected value of the response may be written:

$$E[\mathbf{y}_{n\times 1}|\mathbf{X}_{n\times p}] = \boldsymbol{\mu} = g^{-1}(\beta^*_0\mathbf{1}_n + \mathbf{X}_{n\times p}\boldsymbol{\beta}^*_{p\times 1}) \tag{1.2}$$

where again $\boldsymbol{\beta}^{*T} = [\beta^*_1, \ldots, \beta^*_{p^*}, 0, \ldots, 0]^T$ and p^* is the number of active parameters in the true underlying model. Model 1.1 is a type of GLM where g^{-1} is the identity. Assume that $\mathbf{y}$ is drawn from an exponential family of distributions. This family contains all distributions whose probability density functions can be written:

$$f(y_i;\theta_i,\phi) = \exp\Big(\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi)\Big) \tag{1.3}$$

where ϕ is some nuisance parameter. The family includes—but is not limited to—the Normal, Binomial, and Poisson distributions. The function $g(\cdot)$ is known as the link function and can take many forms, but for the purposes of this discussion suppose the link function is "canonical" in that $g^{-1}(\cdot) = b'(\cdot)$. As an example, in logistic regression, where the response follows a Bernoulli distribution with probability $\boldsymbol{\mu}$, the canonical link function is $g(\mu_i) = \log\big(\mu_i/(1-\mu_i)\big)$. For all models, let $\mathcal{M}^* = \{1, ..., p^*\}$ index the true model's support with cardinality $|\mathcal{M}^*| = p^*$. Notice that model selection will be important to accurate coefficient estimation whenever $p^* < p$.

General Additive Models (GAMs) (Hastie and Tibshirani 1986) are a flexible alternative to the traditional linear model:

$$y_i = \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i \tag{1.4}$$

where the f_j's are functions of one or more covariates. Each $f(\cdot)$ is approximated using a flexible pre-set basis expansion: $f_j(x_{ij}) \approx \sum_{r=1}^{R} \gamma_{jr} b_{jr}(x_{ij})$ where γ_{jr} are the basis coefficients corresponding to the finite basis $\boldsymbol{b}_j(x) = \{b_{j1}(x), ..., b_{jR}(x)\}$. Estimation of $f_j(\cdot)$ is equivalent to estimating γ_{jr} for $r = \{1, ..., R\}$. The process of estimating GAM coefficients can often be reduced to a version of Model 1.1, however the structure of the GAM requires a different approach to sparsity wherein covariates are selected, not individually, but in groups through a penalty such as the Group Lasso (Yuan and Lin 2006), which will be more fully described in Section 1.2. The complexity of the model is measured across two equally important aspects: smoothness and sparsity. Sparsity, as in the linear model, refers to the number of functional covariates estimated to be nonzero in the model. Smoothness describes how locally variable or "wiggly" a given function is estimated to be (Ramsay and Silverman 2005). Regularization often takes place across both sparsity and smoothness.

The functional linear model (Ramsay and Silverman 2005) is used in the EMG data application described in Section 1.3. The scalar-on-function linear model has functional covariates $x_{ij}(t)$ on domain $\mathcal{T}$:

$$y_i = \sum_{j=1}^{p} \int x_{ij}(t)\beta_j(t)dt + \epsilon_i \ . \tag{1.5}$$

Generally, it is not possible to directly observe the covariate functions, but rather they are observed at a dense grid of time points, $\{t_{j1}, ...t_{jN_j}\}$. It is common to assume that the $x_{ij}(\cdot)$ are observed without measurement error and to approximate $\beta_j(t)$ with a basis expansion, similar to the approximation needed for GAMs. In this dissertation, assume that R, the number of basis functions, is large enough to capture the complexity of the function and a B-spline basis is used for the basis expansion. Then the model can be approximated using the Riemann sum:

$$\int x_{ij}(t)\beta_j(t)dt \approx \sum_r \big[(t_{jl} - t_{j,l-1}) \sum_l x_{ij}(t_{jl})b_{jr}(t_{jl})\big]\gamma_{jr} = Z_{ij}^T\gamma_j \tag{1.6}$$

where $Z_{ij} = (Z_{ij1}, ..., Z_{ijR})$ and $Z_{ijr} = (t_{jl} - t_{j,l-1}) \sum_l x_{ij}(t_{jl})b_{jr}(t_{jl})$. With this expansion, the model can be treated as a GAM. Extensions of the estimation techniques used for Models 1.4 and 1.5 have been used for GLMs with functional predictors as well (James 2002; Müller and Stadtmüller 2005).

A penalized estimator is represented generally by $\hat{\boldsymbol{\beta}}_\lambda = \arg\min L(\boldsymbol{\beta}) + P_\lambda(\boldsymbol{\beta})$ where

$L(\boldsymbol{\beta})$ is a loss function comparing $\mathbf{y}$ to its predicted values, and $P_\lambda(\boldsymbol{\beta})$ is a penalty function that depends on tuning parameter(s) $\lambda \geq 0$. $L(\boldsymbol{\beta})$ is frequently taken to be a scaling of the negative loglikelihood such that, for the linear model, $L(\boldsymbol{\beta}) = (2n)^{-1}||\mathbf{y} - \hat{\mathbf{y}}_\lambda||_2^2$ where $\hat{\mathbf{y}}_\lambda = \mathbf{X}\hat{\boldsymbol{\beta}}_\lambda$. Penalty functions can take myriad forms but this dissertation is interested in those that increase as $\boldsymbol{\beta}$ moves away from $\mathbf{0}$. The Lasso (Tibshirani 1996) penalty, $P_\lambda(\boldsymbol{\beta}) = \lambda||\boldsymbol{\beta}||_1 = \lambda \sum_{j=1}^{p} |\beta_j|$, is one that can force estimates to equal 0, thereby performing simultaneous variable selection and estimation. For such sparsity-inducing estimators, comparing the estimated support, $\mathcal{M}_\lambda = \{j \mid \hat{\beta}_{\lambda,j} \neq 0\}$, to $\mathcal{M}^*$ is of interest.

The value of λ balances the importance of minimizing $P_\lambda(\boldsymbol{\beta})$ relative to $L(\boldsymbol{\beta})$, so it is recommended to explore the tuning parameter space to identify an "optimal" value. Potential criteria for an optimal value include identifying a $\hat{\boldsymbol{\beta}}_\lambda$ that minimizes $||\hat{\boldsymbol{\beta}}_\lambda - \boldsymbol{\beta}^*||_2^2$, minimizes $||\mathbf{X}\hat{\boldsymbol{\beta}}_\lambda - \mathbf{X}\boldsymbol{\beta}^*||_2^2$, or has $\mathcal{M}_\lambda = \mathcal{M}^*$. The latter criterion is referred to as support recovery and is a primary focus of this dissertation. Even if a λ exists where $\mathcal{M}_\lambda = \mathcal{M}^*$, there is no guarantee that it will be correctly identified. Popular approaches, such as minimizing information criteria, like AIC (Akaike 1974) and BIC (Schwarz 1978), or minimizing squared prediction error from K-fold Cross Validation often choose a λ that overselects the number of important variables (Feng and Yu 2013; Hastie et al. 2017), i.e., $\mathcal{M}^* \subset \mathcal{M}_\lambda$. Post-selection inference techniques (e.g., Covariance Test for the Lasso (Lockhart et al. 2014) and the Debiased Lasso (Zhang and Zhang 2014)) and multi-stage modifications (e.g., Adaptive (Zou 2006) and Relaxed Lasso (Meinshausen 2007)) can correct for this overselection, albeit with added computations and assumptions. Furthermore, not all of these approaches are available for Model 1.2 and some are penalty-specific.

This dissertation proposes and investigates new methods of tuning parameter selection that are flexible to both the model and penalty used. Section 1.2 is a literature review of topics that appear throughout this work and Section 1.3 describes the EMG data application from Stallrich et al. (2020). Chapter 2 proposes two related methods of tuning parameter selection that can be used for penalized estimation on linear models: α-Modified and AR2 Cross Validation. Both these approaches are informed by penalties that can perform variable selection and rely on a correlation-based evaluation of prediction accuracy. The use of a correlation-based metric for evaluating model fit gives rise to a bias-reduction technique called the α-Modification that can be used to improve estimates without sacrificing variable selection properties. The effectiveness of the methods are evaluated using a simulation study and the approaches are applied to the EMG application to verify their usefulness.

In Chapter 3, the α-Modification is expanded in two ways. First, the procedure defined in Chapter 2 focuses on the linear model. The extension of the procedure to GLMs is non-trivial and is explored in Chapter 3. The second major contribution of Chapter 3 is to address the limitations of the α-Modification as a bias correction with a new alternative to the α-Modification, the α_ν-Modification. The α- and α_ν-Modifications are compared first with a simulation study and then by applying both techniques to a classification problem for Turkish raisins.

As Cross Validation is a crucial part of the approaches described in Chapters 2 and 3, it is important to acknowledge its computational limitations. Primarily, K-fold Cross Validation, which seeks to minimize a measurement of out-of-sample prediction accuracy, can be expensive to implement, particularly for large data sets and time consuming penalized estimation techniques. Chapter 4 proposes a sequential optimization approach for determining which λ to evaluate. The process utilizes a Gaussian process approximation to minimize the number of calculations necessary to find an optimal λ, and then to explore the neighboring space to induce sparsity. We evaluate how well Cross Validation can be approximated with a simulation study and implement the sequential approach to the EMG data application, showing that we can choose very similar models with far fewer calculations. Finally, Chapter 5 summarizes the most important contributions of this work and proposes a few avenues of future research inspired by it.

1.2 Background

Before the new contributions of this dissertation are presented, it is necessary to review popular penalized estimation methods, tuning parameter selection techniques, and some approaches to post-selection inference and bias reduction for penalized estimates. This review is not comprehensive, but it describes the most important approaches that inform the developments presented in Chapters 2, 3, and 4.

1.2.1 Classes of Penalized Estimators

Consider the class of penalties $P_\lambda(\boldsymbol{\beta}) = \lambda||\boldsymbol{\beta}||_q^q = \lambda \sum_j |\beta_j|^q$, $q > 0$, corresponding to the so-called bridge estimators (Frank and Friedman 1993). Knight and Fu (2000) show that for $q \leq 1$ and large enough λ, the penalized estimate will have some $\hat{\beta}_{\lambda,j} = 0$, yielding a continuous approach to the intractable exploration of all submodels. They went on to show that for the linear model under appropriate regularity conditions, the limiting distributions of such $\hat{\beta}_{\lambda,j}$ whose corresponding $\beta_j^* = 0$ can have positive probability mass at 0 when p is fixed, meaning that they are capable of support recovery. Huang et al. (2008) extended this work to prove the same result as p and n grow to infinity, under certain growth rate conditions. However, the large λ necessary for this to occur may cause $|\hat{\beta}_{\lambda,j}| << |\beta_j^*|$ for $j \in \mathcal{M}^*$, thereby inflating criteria commonly used for tuning parameter selection. That is, a λ yielding support recovery may be ignored by popular tuning parameter selection strategies.

The Lasso (Tibshirani 1996) is a bridge estimator where $q = 1$ so, $P_\lambda(\boldsymbol{\beta}) = \lambda||\boldsymbol{\beta}||_1 = \lambda \sum_{j=1}^p |\beta_j|$. It is one of the most commonly used penalized estimators and has been the subject of a large and growing body of theoretical research since its inception (Knight and Fu 2000; Efron et al. 2004; Zou et al. 2007; Chatterjee and Lahiri 2011; Lahiri 2021). A set of conditions that are relevant to theoretical results for the Lasso penalty are Karush-Kuhn-Tucker (KKT) conditions (Karush 1939; Kuhn and Tucker 1951). For these conditions, the optimization problem is framed as minimizing a function $f(\boldsymbol{\beta})$ under a set of L constraints requiring that $g_l(\boldsymbol{\beta}) \leq 0$ for all $l = 1, ..., L$. The Lagrangian that represents the constrained optimization problem is: $f(\boldsymbol{\beta}) + \sum_{l=1}^{L} \lambda_l \nabla g_l(\boldsymbol{\beta})$ where the nonnegative weights $\lambda_1, ..., \lambda_L$ are known as Lagrangian multipliers. Consider the optimal primal vector $\hat{\boldsymbol{\beta}}$ and the optimal Lagrangian multiplier vector, λ^*, also known as the dual vector. The KKT conditions are: (1) Primal feasibility: $g_l(\hat{\boldsymbol{\beta}}) \leq 0$ for all $l \in \{1, ..., L\}$; (2) Complementary slackness: $\lambda_l^* g_l(\hat{\boldsymbol{\beta}}) = 0$ for all $l \in \{1, ..., L\}$; (3) Lagrangian condition: $\hat{\boldsymbol{\beta}}$ and λ^* together satisfy:

$$0 = \nabla f(\hat{\boldsymbol{\beta}}) + \sum_{l=1}^{L} \lambda_l^* \nabla g_l(\hat{\boldsymbol{\beta}}) \ . \tag{1.7}$$

The KKT conditions are necessary and sufficient for $\hat{\boldsymbol{\beta}}$ to be a global optimum whenever the regularity condition of strong duality is satisfied (Hastie et al. 2017). For the Lasso objective function, there is a single constraint, which can be written $g(\boldsymbol{\beta}) = \sum_{j=1}^{p} |\beta_j| - t$ for some positive constant t, meaning that the problem seeks to minimize $f(\boldsymbol{\beta}) = L(\boldsymbol{\beta})$ while simultaneously requiring that $\boldsymbol{\beta}$ belong to an ℓ_1-ball of radius t.

There exists a unique solution to the Lasso problem for linear models and any differentiable, convex loss function when the columns of $\mathbf{X}$ are in general position, meaning that no k-dimensional subspace $L \subseteq \mathbb{R}^n$ for $k < \min(n, p)$ contains more than $k + 1$ elements of the set $\{\pm X_1, ..., \pm X_p\}$, excluding antipodal pairs (Tibshirani 2013). It is also known that there exists a closed form expression for the Lasso under the linear model when $\mathbf{X}^T\mathbf{X} = \mathbf{I}_p$:

$$\hat{\beta}_{\lambda,j} = \text{sign}(\hat{\beta}_{OLS,j})(|\hat{\beta}_{OLS,j}| - \lambda)^+ \ , \tag{1.8}$$

where $\hat{\beta}_{OLS,j}$ is the Ordinary Least Squares (OLS) estimate of the β_j^*, where $\hat{\boldsymbol{\beta}}_{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$, and $(z)_+ = \max(0, z)$. Under the linear model and for iid errors with constant variance, the Lasso has been proven to be model selection consistent, meaning there exists λ_n, a function of n, such that $P(\mathcal{M}_{\lambda_n} = \mathcal{M}^*) \to 1$ as $n \to \infty$ under a condition known as irrepresentability, described below (Zhao and Yu 2006).

A topic of great concern throughout this dissertation is the support recovery capabilities of the Lasso for a fixed n. Partition $\boldsymbol{\beta}^{*T} = (\boldsymbol{\beta}_1^{*T}, \boldsymbol{\beta}_2^{*T})$ where $\boldsymbol{\beta}_2^* = \mathbf{0}$ and partition $\mathbf{X}$ similarly. Let $\mathbf{C} = \frac{1}{n}\mathbf{X}^T\mathbf{X}$. There are a few important conditions that must be met to ensure that support recovery for the linear model has a high probability. First, the mutual incoherence condition, or irrepresentability (Fuchs 2004; Tropp 2006; Meinshausen and Bühlmann 2006; Zhao and Yu 2006) states that there exists some $\gamma > 0$ such that

$$\max_{j \in \mathcal{M}^{*c}} ||(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{X}_1^T x_j||_1 \leq 1 - \gamma \ . \tag{1.9}$$

Next suppose that the smallest eigenvalue of $\mathbf{C}$ is larger than some $C_{\min} > 0$ (Bickel et al. 2009). Finally, assume that the design matrix has normalized columns such that:

$$\max_{j=1,\ldots,p} \|\mathbf{x}_j\|_2/\sqrt{n} \leq K_{\text{clm}} \ . \tag{1.10}$$

Often, $\|\mathbf{x}_j\|_2 = \sqrt{n}$ and K_{clm} is taken to be 1. Under these three conditions and for $\boldsymbol{\epsilon}$ that is iid Normal with mean 0 and standard deviation σ^2, consider the Lasso estimates such that:

$$\lambda_n \geq \frac{8K_{\mathrm{clm}}\sigma}{\gamma}\sqrt{\frac{\log(p)}{n}} \ . \tag{1.11}$$

Then, Wainwright (2009) showed that, with probability greater than $1 - c_1 e^{-c_2 n \lambda_n^2}$, the following conditions are met: (1) $\hat{\boldsymbol{\beta}}_{\lambda_n}$ is unique; (2) $\mathcal{M}_{\lambda_n} \subseteq \mathcal{M}^*$; (3) $\|\hat{\boldsymbol{\beta}}_{\lambda_n}^{\mathcal{M}^*} - \boldsymbol{\beta}_1^*\|_\infty \leq \lambda_n\left(\frac{4\sigma}{\sqrt{C_{\min}}} + \|(\mathbf{X}_1^T\mathbf{X}_1/n)^{-1}\|_\infty\right) = \lambda_n B(\lambda_n, \sigma; \mathbf{X})$ where the infinity-norm of some matrix $\mathbf{A}_{M\times J}$ is given by $\|\mathbf{A}\|_\infty = \max\limits_{m=1,\ldots,M} \sum_{j=1}^{J} |A_{mj}|$ and $\hat{\boldsymbol{\beta}}_{\lambda_n}^{\mathcal{M}^*}$ are the coefficients in $\hat{\boldsymbol{\beta}}_{\lambda_n}$ indexed by $\mathcal{M}^*$; (4) $\hat{\boldsymbol{\beta}}_{\lambda_n}$ includes all j such that $\beta_j > B(\lambda_n, \sigma; \mathbf{X})$.

For the support recovery or model selection consistency results to be useful, we need to be able trust that the tuning parameter selection strategy used will be able to choose the best tuning parameters for a given data set. In short, it is useful to know that some tuning parameter exists such that $\mathcal{M}_\lambda = \mathcal{M}^*$ with high probability, and that $P(\mathcal{M}_\lambda = \mathcal{M}^*) \to 1$ as $n \to \infty$, but it is critical to be able to reliably identify such tuning parameters when they exist. This can be difficult, and is discussed for the Lasso as well as other penalties in Section 1.2.2. Because of its popularity, the Lasso will be the specific penalty used in many of the examples, simulations, and data applications throughout this dissertation.

Another familiar bridge estimator is Ridge Regression, where $q = 2$ (Hoerl and Kennard 1970). The Ridge penalty is cited as inspiration for the Lasso in Tibshirani (1996). In practice, the major difference between the two is that, because $q > 1$, the Ridge penalty is incapable of performing variable selection. For large values of λ, coefficient estimates will be close to, but never exactly, zero. The Ridge penalty is often used when $n < p$, but variable selection is not desired; it is also employed when complete or quasi-complete separation occurs in logistic regression and unpenalized estimates diverge toward negative or positive infinity (Albert and Anderson 1984; Sauter and Held 2016).

Non-bridge penalties, many of which perform variable selection, are also available. The Elastic Net Penalty (Zou and Hastie 2005) is a compromise between the Ridge and Lasso penalties: $P_\lambda(\boldsymbol{\beta}) = \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\boldsymbol{\beta}\|_2^2$. The degree to which the Ridge and Lasso penalties affect the estimates relies on the multidimensional tuning parameter, $\lambda = [\lambda_1, \lambda_2]^T$. It has been suggested that Elastic Net is preferable to the Lasso when the predictors are highly correlated with one another. The Lasso, in this case, tends to choose only one covariate from a group of highly correlated variables without much regard to which one. Elastic Net, on the other hand, is better able to distinguish among the correlated variables (Zou and Hastie 2005). Under the linear model and a regularity condition known as the Elastic Irrepresentable Condition, Jia and Yu (2010) showed that the Elastic Net penalty is variable selection consistent as n, p, and $p^* \to \infty$ under some conditions on the relationship between n, p, and p^* and on the values of λ_1 and λ_2.

The Smoothly Clipped Absolute Deviation (SCAD) (Fan and Li 2001) is a quadratic spline function with knots at two tuning parameters. It is most commonly expressed in terms of its derivative which, at a single point β_j, is:

$$P'_\lambda(\beta_j) = \lambda_1\Big[I(\beta_j \leq \lambda_1) + \frac{(\lambda_2\beta_j - \beta_j)_+}{(\lambda_2 - 1)\beta_j} I(\beta_j > \lambda_1)\Big]. \tag{1.12}$$

The tuning parameter $\lambda_2 > 0$ can be chosen simultaneously with λ_1 via traditional tuning parameter selection techniques. More commonly, $\lambda_2 > 0$ is simply set to 3.7 because this value minimizes Bayes risk (Fan and Li 2001). One of the benefits of this approach is that for large magnitude coefficients, the effect of the penalty is small or zero. Fan and Li (2001) also proved that for GLMs, the method is variable selection consistent, as long as $\lambda_n \to 0$ and $\sqrt{n}\lambda_n \to \infty$ as $n \to \infty$.

A similar non-convex penalty is the Minimax Concave penalty (MCP), which is continuous and nearly unbiased (Zhang 2010). Its derivative for a single coefficient, β_j, is:

$$P'_\lambda(\beta_j) = \begin{cases} \text{sign}(\beta_j)\Big(\lambda_1 - \frac{|\beta_j|}{\lambda_2}\Big) & \text{if } |\beta_j| \leq \lambda_1\lambda_2 \\ 0 & \text{otherwise} \end{cases} . \tag{1.13}$$

where $\lambda_1, \lambda_2 > 0$ comprise λ, the two-dimensional tuning parameter. Zhang (2010) note that as $\lambda_2 \to \infty$, the penalty converges to the Lasso penalty. Similar to SCAD, MCP relaxes the impact of the penalty as the magnitude of coefficient estimates increase, but MCP does so more quickly than SCAD. Zhang (2010) prove that lower bounds exist for the probability of the selection event given the MCP penalty for the linear model, although again these bounds are only achieved under constraints on the tuning parameter. The benefit of both SCAD and MCP is that they are less likely to have $|\hat{\beta}_{\lambda,j}| << |\beta^*_j|$—particularly for large magnitude coefficients—and can therefore lead to better variable selection. Their major limitation is that they require exploration of the tuning parameter space across multiple dimensions, so tuning parameter selection can be more arduous.

The Seamless ℓ_0 penalty (SELO) is a continuous approximation of the ℓ_0 penalty, which simply counts the number of non-zero elements in a vector (Dicker et al. 2013). The reason this approach was necessary is because the ℓ_0 norm is both non-differentiable and computationally inefficient to implement. The penalty is:

$$P_\lambda(\beta_j) = \frac{\lambda_1}{\log\big(2\big)}\log\Big(\frac{|\beta_j|}{|\beta_j| + \lambda_2} + 1\Big) . \tag{1.14}$$

In Li et al. (2012), an algorithm to use the SELO penalty for GLMs is proposed and the approach is proven to be variable selection consistent for this setting under some regularity conditions and assuming $\lambda_n/(\rho_n)^2 \to 0$ where $\rho_n = \min_{j \in \mathcal{M}^*} |\beta^*_j|$.

Another approach to prevent $|\hat{\beta}_{\lambda,j}| << |\beta^*_j|$ and promote accurate variable selection is to adjust bridge penalties. As discussed above, the Ridge penalty is not commonly used for variable selection because it does not shrink any coefficient estimates to exactly zero. A generalization of this penalty, however, where $P_\lambda(\boldsymbol{\beta}) = \sum_{j=1}^{p} \lambda_j \beta_j^2$, was proposed in Wu (2021) to allow some estimates to shrink to zero. This generalization allows for an adaptive step, letting one set of Ridge estimates to inform another, and results in variable selection. The Adaptive Lasso is a similar adjustment for the Lasso penalty (Zou 2006). It follows a two-stage process: first $\hat{\boldsymbol{\beta}}$—a consistent estimator for $\boldsymbol{\beta}^*$ such as OLS—is calculated. Then, with $\gamma > 0$, Adaptive Lasso estimates are found using the penalty $P_\lambda(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} |\hat{\beta}_j|^{-\gamma}|\beta_j|$. When $\gamma = 1$ and $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}_{OLS}$, the objective function reduces to the Nonnegative Garotte (Breiman 1995). This approach is appealing because it is variable selection consistent for both the linear model and GLMs when λ_n satisfies $\lambda_n/\sqrt{n} \to 0$ and $\lambda_n n^{(\gamma-1)/2} \to \infty$ as $n \to \infty$. The Adaptive Lasso is easily generalized to non-Lasso penalties, however the selection of γ and consistent estimation of $\boldsymbol{\beta}^*$ for high dimensional data can be difficult.

The Relaxed Lasso (Meinshausen 2007) is summarized by the following objective function for the linear model where $\lambda_2 \in (0, 1]$ and $\lambda_1 > 0$ are scalar tuning parameters:

$$\frac{1}{2n}||\mathbf{y} - \mathbf{X}(\boldsymbol{\beta} \circ \mathbf{1}_{\mathcal{M}_{\lambda_1}})||_2^2 + \lambda_1\lambda_2||\boldsymbol{\beta}||_1 \ . \tag{1.15}$$

Here, $\boldsymbol{\beta} \circ \mathbf{1}_{\mathcal{M}_{\lambda_1}}$ is the Hadamard product of $\boldsymbol{\beta}$ with the support vector of the Lasso solution for λ_1 under active subset $\mathcal{M}_{\lambda_1}$. For GLMs, the first part of the expression is $-\ell(\boldsymbol{\beta} \circ \mathbf{1}_{\mathcal{M}_{\lambda_1}})$ where $\ell(\cdot)$ is the loglikelihood under the given parameter. In simulations, the Relaxed Lasso returns sparse models with low bias (Meinshausen 2007). For their theoretical results, Meinshausen (2007) assume that the design matrix, $\mathbf{X}$ is a random variable with a Gaussian distribution. Then $X = (X^1, ..., X^{p_n})$ is a p_n-dimensional realization of a Normal distribution with mean 0 and covariance Σ_X and the response variable $Y = X^T\boldsymbol{\beta}^* + \epsilon$

where $\epsilon \sim N(0, \sigma^2)$. For the linear model, loss of the Relaxed Lasso estimator under the tuning parameters λ_1 and λ_2, $\hat{\boldsymbol{\beta}}_{\lambda_1,\lambda_2}$, is $L(\lambda_1, \lambda_2) = E[Y - \mathbf{X}^T\hat{\boldsymbol{\beta}}_{\lambda_1,\lambda_2}]^2 - \sigma^2$. When the number of predictor variables p_n is growing fast with the number of observations, and assuming there exists some $d < 1$ such that Σ_X and Σ_X^{-1} are diagonally dominant at d for all n, then:

$$\inf_{\lambda_1,\lambda_2} L(\lambda_1, \lambda_2) = O_p(n^{-1}) \tag{1.16}$$

whenever λ_1 is sufficiently large. Meinshausen (2007) shows that this rate of convergence is faster than the ordinary Lasso under the sparse high dimensional setting. That is, while the regular Lasso is affected by a large number of noise variables, the Relaxed Lasso is not. This result has not, to our knowledge, been extended to GLMs. Although the Relaxed Lasso performs well both in terms of variable selection and low-bias coefficient estimation, it is not as predominant as a regular Lasso approach because extensions to more complicated penalties and models are often computationally intensive.

The Group Lasso (Yuan and Lin 2006) penalty assumes $E(y_i) = \sum_{j=1}^{p} \mathbf{x}_{ij}^T\boldsymbol{\beta}_j$ where each $\mathbf{x}_{ij}$ is a $d_j \times 1$ vector corresponding with the ith observation of the jth covariate group and shrinks coefficients at the group level. For the linear model, and under some regularity conditions, Nardi and Rinaldo (2008) proved that the Group Lasso is variable selection consistent as long as $\sqrt{n}\lambda\lambda_j \to \infty$ for all $j \notin \mathcal{M}^*$. The Group Lasso penalty is commonly used for GAMs (Model 1.4) and functional linear models (Model 1.5). For such models, the penalty is $P_\lambda(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} ||\boldsymbol{\beta}_j||_{K_j}$ where $\|\mathbf{z}\|_K = (\mathbf{z}^T\mathbf{K}\mathbf{z})^{\frac{1}{2}}$ and $\mathbf{K}_1, ..., \mathbf{K}_J$ are symmetric positive definite matrices. The group of coefficient estimates, $\hat{\boldsymbol{\beta}}_{\lambda,j}$, are then either all zero or all nonzero. In Stallrich et al. (2020), the Group Lasso approach for functional linear models introduced in Gertheiss et al. (2013) was expanded to an adaptive framework in their Sequential Adaptive Functional Estimation method (SAFE).

Some penalties specific to GAMs and functional linear models penalize both smoothness and sparsity (Meier et al. 2009) whereas others penalize only sparsity and impose a constraint on smoothness by the predetermined dimension of the basis expansion (Fan et al. 2015). The more challenging setting for tuning parameter selection is the smoothness-sparsity penalty:

$$P_\lambda(\beta) = \lambda_1 \sum_{j=1}^{p} \left(\|\beta_j\|^2 + \lambda_2\|\beta_j''(t)\|^2\right)^{1/2} \tag{1.17}$$

where $\|\beta_j\|^2 = \int\{\beta_j(t)\}^2 dt$ and $\|\beta_j''(t)\|^2 = \int\{\partial^2\beta_j(t)//\partial t\}^2 dt$. Gertheiss et al. (2013) propose a method of estimation for this penalty that exploits the Group Lasso penalty. Note that this type of penalty has aspects other than the magnitude of coefficients over which to regularize, which can make tuning parameter selection more difficult.

1.2.2 Tuning Parameter Selection Strategies

Desirable statistical properties of penalized estimators require certain tuning parameter values, making tuning parameter selection a pivotal step in the analysis. Even in simple circumstances, i.e. the linear model with only a single tuning parameter, tuning parameter selection can be difficult, as will be discussed here. For GAMs and functional linear models, where both sparsity and smoothness need to be accounted for in evaluating model complexity, tuning parameter selection can be even more complicated. The first challenge is to determine the set of tuning parameters, Λ, that are worthy of evaluation. For the Lasso penalty on linear models, when the entire solution path can be easily found, this challenge is simplified but for many penalties this is more difficult.

The most efficient approaches, such as information criteria, are evaluated over the full data set and do not require data-splitting. Information criteria take the form $IC(\lambda) = -2\ell(\hat{\boldsymbol{\beta}}_\lambda) + h(k_\lambda)$ where $\ell(\hat{\boldsymbol{\beta}}_\lambda)$ is the loglikelihood of the data under $\hat{\boldsymbol{\beta}}_\lambda$ and $h(\cdot)$ is a penalty to prevent overselection based on $k_\lambda = |\mathcal{M}_\lambda|$. Two well-known criteria are AIC, where $h(k_\lambda) = 2k_\lambda$ (Akaike 1974), and BIC, where $h(k_\lambda) = k_\lambda \log(n)$ (Schwarz 1978). AIC often overselects, particularly for small sample sizes, so a corrected AIC (AICc), where $h(k_\lambda) = 2k_\lambda + \frac{2k_\lambda^2+2k_\lambda}{n-k_\lambda-1}$ can be used for small n (Hurvich and Tsai 1989). BIC, unlike AIC, is variable selection consistent for the linear model when $\epsilon_i \overset{iid}{\sim} N(0,\sigma^2)$ (Nishii 1984).

For the linear model with Gaussian errors, $\ell(\hat{\boldsymbol{\beta}}_\lambda)$ involves σ^2 and for unknown σ^2, $-2\ell(\hat{\boldsymbol{\beta}}_\lambda) \propto n\log\left(\hat{\sigma}_\lambda^2\right)$ where $\hat{\sigma}_\lambda^2$ is the estimated model variance at λ. Buhlmann and van de Geer (2011) point out this can lead to overselection when $\hat{\sigma}_\lambda^2 < 1$. When possible, σ^2 is substituted with the mean squared error from a presumed low bias model (Hastie et al. 2017), but this may be challenging to identify for high-dimensional data. In short AIC and BIC can struggle for penalized estimators because they were developed for unpenalized maximum likelihood estimation. To address this, Hui et al. (2015) proposed the Extended Regularized Information Criterion (ERIC) specifically for tuning parameter selection of penalized estimators, where $h(k_\lambda) = 2\upsilon k_\lambda \log(n\phi/\lambda)$ and $\upsilon > 0$. ERIC outperformed popular tuning parameter selection approaches in their simulations for the Adaptive Lasso, but υ controls the balance between fit and complexity and its selection is subjective.

Another information criterion is the Extended BIC (EBIC) developed for the linear model in Chen and Chen (2008) as a response to overselection using BIC. The definition has since been expanded to include GLMs, although model selection consistency has only been proven in the linear model with Normal errors (Chen and Chen 2012). For this criterion, $h(k_\lambda) = k_\lambda \log(n) + 2k_\lambda \gamma \log(p)$ where $\gamma > 0$. The authors recommend $\gamma = 0.5$, but no theoretical justification is given for the recommendation.

Cross Validation (CV), is the process of splitting data into training and validation sets, often repeatedly and is, perhaps, the most common approach to tuning parameter selection in penalized estimation. It takes many forms, but K-fold CV (Geisser 1975; Allen 1974; Stone 1974) is arguably the most common. In K-fold CV, the data are partitioned into K sets, or folds, of size n_k each. Estimates are made using $K-1$ of the folds and then predictions are made for the remaining fold, denoted $\hat{\mathbf{y}}_{\lambda,k}$. This is repeated using each fold as the holdout set for all λ. Prediction errors are found and are averaged across the K folds to give the average prediction error (APE) for each λ. For the linear model, prediction error is calculated $\frac{1}{n_k}||\mathbf{y}_k - \hat{\mathbf{y}}_{\lambda,k}||_2^2$; for GLMs, the deviance, twice the negative difference in the loglikelihood for a model fit with parameter λ and the "saturated" model where $\hat{y} = y_i$, can be used as prediction error. The λ with the minimum APE is selected, or a 1SE rule—which chooses the largest λ within one standard error of the minimum APE—is implemented to induce sparsity. The use of a 1SE rule is most common for a one-dimensional λ, but Stallrich et al. (2020) proposed a multidimensional extension where the chosen $\hat{\boldsymbol{\beta}}_\lambda$ lies within one standard error of the minimum and also minimizes some measure of model complexity. K-fold CV still has a tendency towards overselection, even with a 1SE rule (Krstajic et al. 2014).

Feng and Yu (2013) propose a method of CV they call Consistent CV (CCV), meant to address some of the overselection concerns of K-fold CV. This approach uses the whole dataset to find the solution path. Along the solution path, the sequence of active subsets is defined. K-fold CV then splits the data into K folds. Each fold is set aside, using the other $K - 1$ folds to generate unpenalized maximum likelihood solutions according to the active sequence generated on the whole dataset and the APE is found using these unpenalized estimates. The active subset found to have the smallest APE is the optimal and unpenalized maximum likelihood estimates from the full dataset are found for the optimal subset. This process has been shown to be variable selection consistent for GLMs under certain conditions regarding the behavior of $\mathbf{X}$ and $\boldsymbol{\beta}^*$ (Feng and Yu 2013).

Another variation on CV is Percentile CV, proposed in Roberts and Nowak (2014). In this approach, K-fold CV is carried out across many randomly generated sets of folds, the minimum APE λ is selected for each, and then some percentile of those λ values is chosen as the optimal. The authors suggest that a percentile of 0.75 or larger corresponds to a practice that generally overselects. This practice appears similar to a 1SE Rule in approach. However, because repeated K-fold CV is required, it is a computational burden.

Generalized Cross Validation (GCV) is an efficient approximation of n-fold Cross Validation (Craven and Wahba 1978; Golub et al. 1979). GCV is appropriate when the estimation procedure admits linear predictions $\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$ for some matrix $\mathbf{S}$ (Hastie et al. 2017). For example, in Ridge Regression on the linear model, $\mathbf{S} = \mathbf{X}((\mathbf{X}^T\mathbf{X})^{-1} + \lambda\mathbf{I})\mathbf{X}^T$. GCV minimizes a function that, for the linear model, divides $L(\hat{\boldsymbol{\beta}}_\lambda) = ||\mathbf{y} - \hat{\mathbf{y}}_\lambda||_2^2$ by a function of k_λ, the effective degrees of freedom. However, just like with other information criteria, GCV has sometimes been shown to lead to overselection (Homrighausen and McDonald 2018).

1.2.3 Post-selection Inference

Because many of the most popular methods of tuning parameter selection tend to overselect, it is common to perform post-selection inference to carry out further variable selection. Post-selection inference techniques have been the subject of a great deal of recent research (van de Geer et al. 2014; Javanmard and Montanari 2013; Taylor and Tibshirani 2015; Lee et al. 2016; Shi et al. 2020). In this dissertation, focus is given to two methods, the Debiased Lasso and the Covariance Test, because of their relationship with the new method proposed in Chapter 2.

The Debiased Lasso is used for hypothesis testing and to calculate confidence intervals for coefficient estimates in Model 1.1 when errors are Gaussian and variance constant. When $\hat{\boldsymbol{\beta}}_\lambda$ are Lasso estimates, Debiased Lasso solutions are found by adding $\frac{1}{n}\boldsymbol{\Theta}\mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_\lambda)$, where $\boldsymbol{\Theta}$ is an estimate of $(\mathbf{X}^T\mathbf{X})^{-1}$, to $\hat{\boldsymbol{\beta}}_\lambda$ (Zhang and Zhang 2014). This method focuses on estimation bias rather than direct variable selection, so the use of an additive bias correction is coherent; estimates for zero effect coefficients are small and hence their bias is small. Moreover, the point of the Debiased Lasso correction is to define the distribution of Lasso estimates and carry out variable selection in multiple stages though inference. van de Geer et al. (2014) proposed a Debiased Lasso for GLMs, however, Xia et al. (2020) found through extensive simulations that bias for these models remains high, leading to true coverage probabilities that are much lower than their nominal levels

The Covariance Test for the Lasso (Lockhart et al. 2014) under the linear model takes advantage of the LARS algorithm, which constructs the Lasso solution path by adding variables one at a time (Efron et al. 2004). Suppose the knots found through LARS are $\lambda_1 > \lambda_2 > ...\lambda_L$ and $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$. To test the significance of the predictor which first enters the model at λ_l, we refit the Lasso, setting $\lambda = \lambda_{l+1}$ and using $\mathcal{M}_{\lambda_{l-1}}$ as predictors. This gives coefficient estimates that lead to predictions for the response, $\hat{\mathbf{y}}^*$. The test statistic is:

$$F_{2,n-p} = \frac{1}{\hat{\sigma}^2}\Big(\mathbf{y}^T\hat{\mathbf{y}}_{\lambda_{l+1}} - \mathbf{y}^T\hat{\mathbf{y}}^*\Big) \tag{1.18}$$

where $\hat{\sigma}^2$ is the mean squared error of a model including all parameters. The Covaraince Test for the Lasso is distinctive because it assesses fit using covariance rather than squared error loss. Although it requires estimation of σ^2 and its extension to non-Lasso penalties is not well studied, it provides precedence for the use of correlation in the evaluation of model fit for variable selection, a topic further explored in Chapter 2.

Finally, there exists a more general strategy for post-selection inference that is not as tied to the Lasso in particular, although many of its applications are related to the Lasso. When the selection event can be written $\{\mathbf{Ay} \leq \mathbf{b}\}$ for some matrix $\mathbf{A}$ and vector $\mathbf{b}$, there exists a general scheme for post-selection inference that gives exact confidence intervals and p-values for Model 1.1 with Gaussian errors. This is the case for the LARS algorithm (Efron et al. 2004), forward stepwise regression, and the Lasso when λ is fixed (Hastie et al. 2015). Choose $\boldsymbol{\eta}$ such that inference about $\boldsymbol{\eta}^T E[\mathbf{y}]$ is of interest. Let

$\mathcal{V}^0(\mathbf{y}) = \min_{j:a_j=0}(b_j - (\mathbf{A}\mathbf{y})_j)$ and let:

$$\mathcal{V}^-(\mathbf{y}) = \max_{j:a_j<0} \frac{b_j - (\mathbf{A}\mathbf{y})_j + a_j \boldsymbol{\eta}^T \mathbf{y}}{a_j} \qquad \mathcal{V}^+(\mathbf{y}) = \min_{j:a_j>0} \frac{b_j - (\mathbf{A}\mathbf{y})_j + a_j \boldsymbol{\eta}^T \mathbf{y}}{a_j} \tag{1.19}$$

where $\mathbf{a} = \mathbf{A}\boldsymbol{\eta}/\|\boldsymbol{\eta}\|_2^2$. Lee et al. (2016) and Lockhart et al. (2014) show what is known as the polyhedral lemma for Gaussian errors: $\{\mathbf{A}\mathbf{y} \leq \mathbf{b}\}$ equals $\{\mathcal{V}^-(\mathbf{y}) \leq \boldsymbol{\eta}^T\mathbf{y} \leq \mathcal{V}^+(\mathbf{y}); \mathcal{V}^0(\mathbf{y}) \geq 0\}$. Where $F_{\mu,\sigma^2}^{c,d}$ is the cumulative distribution function of a truncated normal distribution with support $[c, d]$:

$$F_{\boldsymbol{\eta}^T E[\mathbf{y}],\sigma^2\|\boldsymbol{\eta}\|_2^2}^{\mathcal{V}^-(\mathbf{y}),\mathcal{V}^+(\mathbf{y})}\big(\boldsymbol{\eta}^T\mathbf{y}\big) \mid \{\mathbf{A}\mathbf{y} \leq \mathbf{b}\} \sim U(0,1) \ , \tag{1.20}$$

assuming constant variance σ^2. This fact can be used to perform conditional inference about any linear combination of the expected value of the response. This is useful because it allows for inference upon each step of the LARS algorithm or any Lasso solution for a fixed tuning parameter λ. When used for successive steps of LARS, it is known as the Spacing Test (Tibshirani et al. 2014) and is a non-asymptotic version of the Covariance Test.

1.3 Optimal EMG Placement for a Robotic PC

The complex variable selection problem introduced in Stallrich et al. (2020) is an example of a data application where tuning parameter selection can be challenging. The shrinkage of large magnitude coefficients and practical limitations inhibiting comprehensive exploration of the tuning parameter space can both contribute to the overspecification of models.

The goal of the application is to efficiently translate signals from the residual muscle of transradial amputees into the movement of a robotic hand. The process of decoding EMG

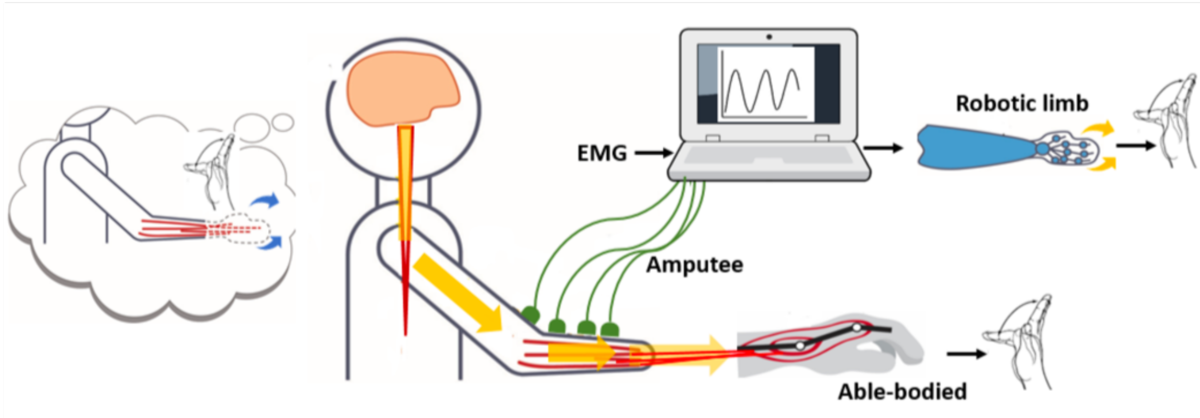

Figure 1.1: The process of hand movement for an able-bodied subject and a transradial amputee.

signals into limb movement for both a transradial amputee and an able-bodied subject is demonstrated in Figure 1.1. State-of-the-art PC decoders require time consuming training sessions, model movement with a large number of EMG sensors, and are prone to poor prediction. Biomechanical models have suggested that only a few of these EMG signals are necessary to predict certain movements, so selecting which signals should be involved in a predictive model will improve prediction accuracy and make the device more comfortable and easier to re-calibrate (Stallrich et al. 2020).

The data consist of a response, hand position for an able bodied person, and 16 EMG signals as predictors, denoted $X_1, ..., X_{16}$; one of these, X_9, is an external signal, unrelated to the movement. Figure 1.2 demonstrates the placement of the EMG sensors. Data has been thinned, as in Stallrich et al. (2020), to reduce the autocorrelation of the responses. The EMG signals are normalized such that 0 indicates no contraction and 1 indicates maximal contraction, based on data from a previous training session. Six data sets are analyzed in the following Chapters, corresponding to three consistent finger movement patterns (FC1, FC2, FC3) and three random patterns (FR1, FR2, FR3). As data were collected from an able-bodied subject, it was known that three of the 16 sensors, denoted

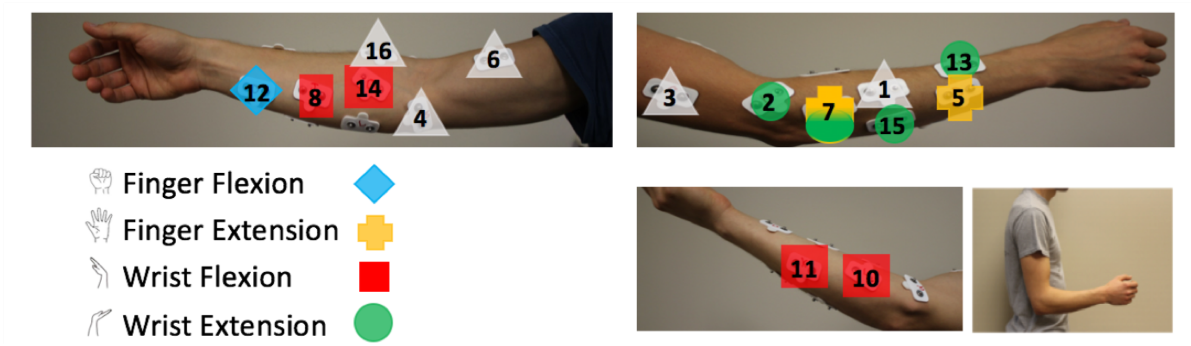

Figure 1.2: EMG sensor placement along the arm of a subject, differentiated by movement degree of freedom contribution. The posture for data collection is shown in the bottom right.

X_5, X_7, X_{12}, targeted muscles known to fully explain finger movement. Sensors X_5 and X_7 collected information from the same muscle contraction, so only one of the pair is necessary to predict finger position. An ideal model would thus include X_{12} and either X_5 or X_7, but recovery of all three sensors is also acceptable. Additional information regarding the data collection and processing can be found in Stallrich et al. (2020).

The data are modeled as functional predictors—the EMG signals—of a scalar response—the hand position. One of the reasons this is a challenging problem is that the EMG signals are highly correlated with one another, making variable selection through penalized estimation difficult. Furthermore, known biomechanical forces of hand movement—e.g. physical restraints and passive forces—mean that an activated predictive EMG signal does not always indicate movement and an inactive predictive EMG signal does not always indicate rest. This is demonstrated in Figure 1.3 for X_7 and X_{12}, two signals known to contribute to finger flexion and extension. For these reasons, Stallrich et al. (2020) model velocity rather than predicting hand position directly.

The primary objective is to efficiently identify EMG sensors that are useful predictors

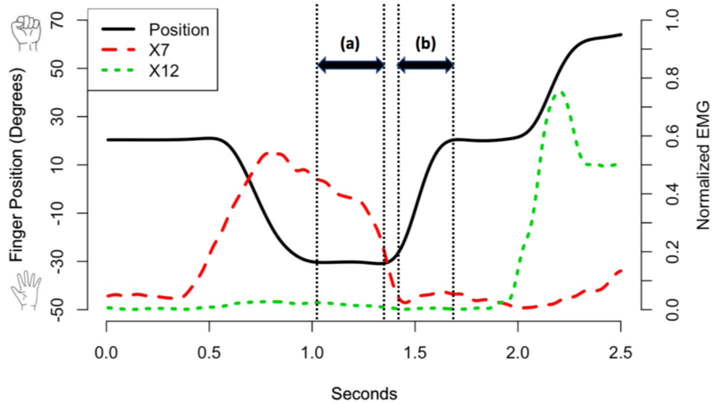

Figure 1.3: Normalized EMG signals 7 and 12, which correspond to finger flexion and finger extension. Time period (a) demonstrates the physical constraint of maximal finger extension and time period (b) demonstrates the passive forces of the hand returning to rest.

of hand movement, using the model:

$$y_i = \sum_{j=1}^{p} \int_{-\delta}^{0} X_{ij}(t)\gamma_j(t, z_i)dt + \epsilon_i \tag{1.21}$$

where y_i is the velocity, $X_{ij}(t)$ represents past EMG signals, t is time with respect to δ, a pre-determined historical length, and z_i is the position. The linear approximations used for $\gamma_j(\cdot)$ may be found in Appendix A.2. Following Gertheiss et al. (2013), Stallrich et al. (2020) proposed a penalized estimation procedure where the penalty accounted for both sparsity and smoothness of the $\gamma_j(\cdot, \cdot)$ functions:

$$P_\lambda(\gamma_j) = \lambda_1(f_k||\gamma_j||_2^2 + g_k\lambda_t||\gamma''_{j,t}||_2^2 + h_k\lambda_z||\gamma''_{j,t}||_2^2)^{1/2} \tag{1.22}$$

where f_k, g_k, and h_k are adaptive weights, initially set to 1. These adaptive weights are introduced in Stallrich et al. (2020) to formulate a multi-stage penalized estimation procedure known as Sequential Adaptive Functional Estimation (SAFE), which introduces adaptive stages to the Group Lasso approach to functional regression developed in

Gertheiss et al. (2013). SAFE is extremely successful in choosing appropriately sparse models, but it uses as many as five stages to do so.

This can be a time consuming process so exploration via K-fold CV across a dense grid of the three tuning parameters is not always practical. Stallrich et al. (2020) reduced the computation time somewhat by evaluating a relatively sparse set of the smoothing penalties, λ_t and λ_z. This improves the efficiency of the procedure, but the authors note that it is not clear whether the tuning parameter space is sufficiently explored. This problem will be addressed in two ways: first in Chapter 2 by introducing two new Cross Validation procedures that can efficiently perform variable selection; next, in Chapter 4, a sequential design approach to choosing tuning parameters is proposed to allow for thorough exploration of the tuning parameter space without enormous computational expense.

CHAPTER

2

TUNING PARAMETER SELECTION FOR PENALIZED ESTIMATION VIA R^2

2.1 Motivation

This chapter proposes a new K-fold Cross Validation strategy for Model 1.1 that assesses the predictive quality of $\hat{\boldsymbol{\beta}}_\lambda$ via squared prediction correlation rather than squared prediction error, with a particular focus on penalized estimators, such as the Lasso, that perform variable selection. The invariance of correlation to the scale of $\hat{\boldsymbol{\beta}}_\lambda$ reduces the impact of the potential shrinkage when estimating large β_j^* that would otherwise burgeon squared prediction error.

As a demonstration, consider a simulated dataset from Model 1.1: $\{\mathbf{y}, \mathbf{X}\}$ with $n = p = 100$, $p^* = 5$, and $\sigma^2 = 1$, where the entire solution path of the Lasso has been generated. $\mathbf{X}$ were independently sampled from a standard Normal distribution and the active coefficients in $\boldsymbol{\beta}^*$ were $\{2.13, 1.81, -2.46, -1.89, -2.51\}$. Many λ values had $\mathcal{M}_\lambda = \mathcal{M}^*$, so a wide range were optimal for variable selection. Figure 2.1 plots $\mathbf{y}$ against their in-sample predictions, $\hat{\mathbf{y}}_\lambda$, under four different λ values, and summarizes the models' number of false positives (FP), average prediction error (APE) from 10-fold CV, and in-sample squared prediction correlation (R^2). Figure 2.1(a) corresponds to the λ determined by a Cross Validation 1SE Rule (CV 1SE) and 2.1(b) corresponds to the λ that minimizes $||\mathbf{X}\boldsymbol{\beta}^* - \mathbf{X}\hat{\boldsymbol{\beta}}_\lambda||_2$ (Min PB). Among the λ having $\mathcal{M}_\lambda = \mathcal{M}^*$, we consider the largest and smallest in Figures 2.1(c) and 2.1(d), respectively.

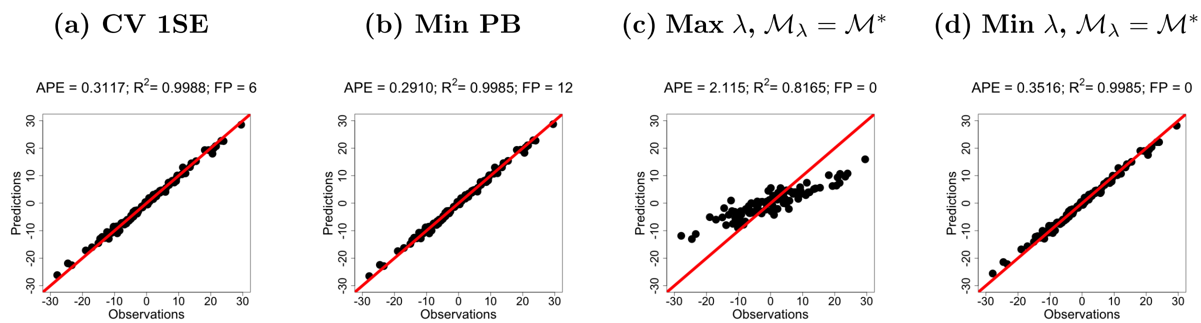

Figure 2.1: In-sample observations (y_i) versus predictions ($\hat{y}_i$) for Lasso estimator at four values of λ for a data set with $n = 100$, $p = 100$, and $p^* = 5$. We also report average prediction error (APE) under 10-fold CV, in-sample squared prediction correlation (R^2), and the number of false positives (FP). The minimum APE plus-or-minus one standard error is 0.2844 ± 0.0262 and all estimates have $\mathcal{M}^* \subseteq \mathcal{M}_\lambda$.

All models shown in Figure 1 include the 5 important variables. Both the CV 1SE and Min PB models have small APE and large R^2, but many false positives. The model in Figure 2.1(c) has no false positives, but has relatively large APE. The model under Figure 2.1(d) also has no false positives and its in-sample R^2 approximately equals that for the Min PB estimate. This motivates a tuning parameter selection strategy based on an R^2 metric rather than APE to compromise between prediction and variable selection.

The use of an R^2 metric for assessing model fit highlights a relationship between the metric and a multiplicative adjustment on $\hat{\boldsymbol{\beta}}_\lambda$, the α-Modification. For $\hat{\boldsymbol{\beta}}_\lambda$ with certain statistical properties, the adjustment can reduce the bias of $\hat{\boldsymbol{\beta}}_\lambda$ thereby improving its predictive potential. Notably, the α-Modification for the Lasso bears similarities to the Nonnegative Garrote (Breiman 1995) and Relaxed Lasso Meinshausen (2007). However, unlike these two methods, the α-Modification can be applied to any penalized least-squares estimator, including non-convex penalties, without additional computational complexity.

This chapter is organized as follows. Section 2.2 justifies the value of the R^2 metric and provides statistical properties for a general class of penalized estimators. Finite-sample properties are then derived for the α-Modification for the Lasso in Section 2.3. Section 2.4 presents a simulation study of the new approaches and Section 2.5 applies the new methods to the EMG data of Stallrich et al. (2020). Section 2.6 provides a discussion on the implications of the new framework for evaluating model fit and propose avenues of future research. Proofs of all results may be found in Section 2.A, with additional numerical results given in Appendix A.1 and A.2.

2.2 Methodology

2.2.1 Why Correlation Over Average Prediction Error?

Distinguish the magnitude of $\boldsymbol{\beta}^*$, denoted $\alpha^* = ||\boldsymbol{\beta}^*||_2$, from its direction, $\boldsymbol{\xi}^* = \boldsymbol{\beta}^*/\alpha^*$. Then $\boldsymbol{\xi}^*$ contains all information about $\mathcal{M}^*$. The same summaries may be computed from a penalized estimate, $\hat{\boldsymbol{\beta}}_\lambda$, denoted by $\hat{\alpha}_\lambda$ and $\hat{\boldsymbol{\xi}}_\lambda$. Comparing $\mathcal{M}_\lambda$ to $\mathcal{M}^*$ is equivalent to comparing the supports of $\hat{\boldsymbol{\xi}}_\lambda$ and $\boldsymbol{\xi}^*$. A tuning parameter selection strategy based on squared prediction error, however, concerns both magnitude and direction. Let $\{\mathbf{y}, \mathbf{X}\}$ denote a holdout sample where $\mathbf{y}$ and $\mathbf{X}$ have been centered. The squared prediction error for $y_i = \mathbf{x}_i^T \boldsymbol{\beta}^* + \epsilon_i$ is

$$\sum_i (y_i - \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)^2 = \sum_i (\varepsilon_i + \mathbf{x}_i^T(\alpha \boldsymbol{\xi}^* - \hat{\alpha}_\lambda \hat{\boldsymbol{\xi}}_\lambda))^2 \ . \tag{2.1}$$

In the ideal situation where $\hat{\boldsymbol{\xi}}_\lambda = \boldsymbol{\xi}^*$, (2.1) will be inflated when $\hat{\alpha}_\lambda \neq \alpha^*$. Indeed, for penalized estimators typically $\hat{\alpha}_\lambda < \alpha^*$. Therefore, it is possible for an estimate having $\hat{\boldsymbol{\xi}}_\lambda = \boldsymbol{\xi}^*$ to have a large APE and so would be unlikely to be chosen by an APE-based tuning parameter selection strategy.

Consider now the correlation between $\mathbf{y}$ and its prediction $\hat{\mathbf{y}}_\lambda = \mathbf{X}\hat{\boldsymbol{\beta}}_\lambda$. After some simplification, we get the expression

$$\mathrm{Corr}(\mathbf{y}, \hat{\mathbf{y}}_\lambda) = \frac{(\mathbf{X}\boldsymbol{\xi}^* + \tilde{\boldsymbol{\epsilon}})^T \mathbf{X}\hat{\boldsymbol{\xi}}_\lambda}{||\mathbf{X}\boldsymbol{\xi}^* + \tilde{\boldsymbol{\epsilon}}||_2 \ ||\mathbf{X}\hat{\boldsymbol{\xi}}_\lambda||_2} \tag{2.2}$$

where $\tilde{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}/\alpha^*$ is a scaled error vector that does not depend on $\hat{\boldsymbol{\beta}}_\lambda$. The $\hat{\alpha}_\lambda$ has no influence over this summary so this measure better compares the $\boldsymbol{\xi}^*$ and $\hat{\boldsymbol{\xi}}_\lambda$, and hence better assesses support recovery than squared prediction error. This is especially true for penalized estimators that struggle to estimate α^*. The proposed tuning parameter selection strategy, called AR2 CV, employs K-fold CV with folds $\{\mathbf{y}_k, \mathbf{X}_k\}$ but replaces APE with

$$\mathrm{AR2} = \frac{1}{K} \sum_{k=1}^{K} 1 - \mathrm{Corr}(\mathbf{y}_k, \hat{\mathbf{y}}_{\lambda,k})^2 \ . \tag{2.3}$$

The optimal λ may be chosen as the one that minimizes AR2, but significant improvements in support recovery have been found under an analogous 1SE Rule.

Applying AR2 CV with a 1SE rule to the toy example in Section 2.1, the optimal λ is that given in Figure 2.1(d). The AR2 CV estimator was then able to compromise between support recovery and prediction error, while the APE-based CV prioritized prediction error. The coefficient estimates for the $j \in \mathcal{M}^*$ under the APE CV model exhibit less bias than those of the AR2 CV model. A potential drawback then of AR2 CV is that its ignorance to $\hat{\alpha}_\lambda$ may lead to a $\hat{\boldsymbol{\beta}}_\lambda$ that exhibits more shrinkage than is desired. A potential remedy is to proceed selection of $\mathcal{M}_\lambda$ with unpenalized estimation for only predictors in $\mathcal{M}_\lambda$. Section 2.2.2 discusses an alternative strategy that is related to AR2 CV and adjusts the shrinkage of $\hat{\boldsymbol{\beta}}_\lambda$.

2.2.2 The α-Modification

Consider now the in-sample $\mathbf{y}$ and their predictions, $\hat{\mathbf{y}}_\lambda$. If $\hat{\mathbf{y}}_\lambda \neq \mathbf{0}$, calculate the least-squares estimate $\hat{\alpha}_\lambda = \arg\min_\alpha ||\mathbf{y} - \alpha\hat{\mathbf{y}}_\lambda||_2^2 = (\hat{\mathbf{y}}_\lambda^T\hat{\mathbf{y}}_\lambda)^{-1}\hat{\mathbf{y}}_\lambda^T\mathbf{y}$. Note this $\hat{\alpha}_\lambda$ likely differs from $||\hat{\boldsymbol{\beta}}_\lambda||_2$. By definition, the predictions $\hat{\alpha}_\lambda\hat{\mathbf{y}}_\lambda$ will be as close or closer to $\mathbf{y}$ as $\hat{\mathbf{y}}_\lambda$. Note these $\hat{\alpha}_\lambda\hat{\mathbf{y}}_\lambda$ are equivalent to prediction under the adjusted penalized estimate, $\hat{\alpha}_\lambda\hat{\boldsymbol{\beta}}_\lambda$. This estimate results from minimizing the biconvex objective function

$$\frac{1}{2n}||\mathbf{y} - \alpha\mathbf{X}\boldsymbol{\xi}||_2^2 + P_\lambda(\boldsymbol{\xi}) \tag{2.4}$$

by first fixing $\alpha = 1$ to estimate $\hat{\boldsymbol{\xi}} = \hat{\boldsymbol{\beta}}_\lambda$ and then minimizing the function for α given $\hat{\boldsymbol{\xi}}$. Note that as $\lambda \to 0$, $\hat{\alpha}_\lambda \to 1$ because the objective function focuses most of its attention on the loss function. Thus the impact of the α-Modification is more noticeable for larger values of λ. For such λ, the hope is that the α-Modified estimate will correct the bias of $\hat{\boldsymbol{\beta}}_\lambda$ due to shrinkage. We now study the properties of these α-Modified estimates.

The α-Modification is similar to many other existing methods. First, one can view the modification as reversing the process of calculating the Nonnegative Garrote estimator, which starts with OLS estimates and then performs penalization (Breiman 1995). Zou and Hastie (2005) also recommended a multiplicative adjustment to the Elastic Net estimator, although the adjustment only involves one of the tuning parameters. For penalties that satisfy $P(\alpha\boldsymbol{\xi}) = \alpha P(\boldsymbol{\xi})$ for $\alpha > 0$, we may rewrite the penalty in (2.4) as $\lambda\alpha^{-1}P(\alpha\boldsymbol{\xi})$ which resembles the Relaxed Lasso penalty, so long as $\alpha^{-1} \in (0, 1]$. Finally, the Debiased Lasso attempts to reduce bias through an additive adjustment, but this causes $\hat{\beta}_{j,\lambda} = 0$ to become nonzero. The multiplicative adjustment proposed here does not change the support of $\hat{\boldsymbol{\beta}}_\lambda$.

Penalized estimates are typically shrunk towards zero so the α-Modification will correct this type of bias only if $\hat{\alpha}_\lambda \geq 1$. This property is guaranteed for common penalties:

Theorem 1. *Suppose $P_\lambda(\boldsymbol{\beta}) = \sum_{\ell=1}^{L} \lambda_\ell g_\ell(\boldsymbol{\beta})$ where $g_\ell(\boldsymbol{\beta})$ is convex and minimized at $\boldsymbol{0}_p$. Then $\hat{\alpha}_\lambda \geq 1$ when $\hat{\boldsymbol{\beta}}_\lambda \neq \boldsymbol{0}$.*

Amplifying penalized estimates does not necessarily decrease bias. To evaluate the α-Modification as a bias-reduction tool, consider the following result.

Lemma 1. *If there exists a λ where $\hat{\boldsymbol{\xi}}_\lambda = \boldsymbol{\xi}^*$ as defined in Section 2.2.1, then $E[\hat{\alpha}_\lambda \hat{\boldsymbol{\beta}}_\lambda] = \boldsymbol{\beta}^*$.*

Lemma 1 conditions on an event that may have probability 0. The following lemma considers a broader condition, whereby the penalized estimate recovers the direction of an OLS estimate under the submodel $\mathcal{M}_\lambda$, denoted by $\hat{\boldsymbol{\beta}}_{OLS}^{\mathcal{M}_\lambda}$. Here $\hat{\beta}_{OLS,j}^{\mathcal{M}_\lambda} = \hat{\beta}_{OLS,j}$ when $j \in \mathcal{M}_\lambda$ and zero otherwise.

Lemma 2. *If $\hat{\boldsymbol{\beta}}_\lambda \propto \hat{\boldsymbol{\beta}}_{OLS}^{\mathcal{M}_\lambda}$ then $\hat{\alpha}_\lambda \hat{\boldsymbol{\beta}}_\lambda = \hat{\boldsymbol{\beta}}_{OLS}^{\mathcal{M}_\lambda}$.*

There are multiple examples that satisfy the condition of Lemma 2. The OLS estimator itself qualifies as a scaled OLS estimator, where $\hat{\alpha}_\lambda = 1$. Ridge estimates when $\mathbf{X}^T\mathbf{X} = n\mathbf{I}_p$ also take this form, having modified estimates $\frac{1}{1+\lambda}\hat{\boldsymbol{\beta}}_{OLS}$. Lemma 2 also applies whenever $\hat{\boldsymbol{\beta}}_\lambda$ contains exactly one non-zero entry. Finally, note the Relaxed Lasso always includes $\hat{\boldsymbol{\beta}}_{\lambda_1,\lambda_2} = \hat{\boldsymbol{\beta}}_{OLS}^{\mathcal{M}_\lambda}$ among its solutions by setting $\lambda_2 = 0$. Lemma 2 shows that this is can sometimes occur for the α-Modified estimates as well.

The α-Modification serves to improve predictions under a given $\hat{\boldsymbol{\beta}}_\lambda$ through a positive, multiplicative adjustment and so its ability to improve estimation depends on the properties of $\hat{\boldsymbol{\beta}}_\lambda$. While this paper is mainly concerned with tuning parameter selection for finite sample sizes, properties of $\hat{\boldsymbol{\beta}}_\lambda$ are easier to study as $n \to \infty$ and the same is true for α-Modified estimators. Concerning model selection consistency, since the support of $\hat{\alpha}_\lambda\hat{\boldsymbol{\beta}}_\lambda$ equals the support of $\hat{\boldsymbol{\beta}}_\lambda$, the α-Modified estimator is model selection consistent if and only if $\hat{\boldsymbol{\beta}}_\lambda$ is model selection consistent. The following theorem establishes estimation consistency of α-Modified estimators.

Theorem 2. *Fix p and suppose there exists a positive definite matrix $\boldsymbol{C}$ where $\frac{1}{n}\boldsymbol{X}_n^T\boldsymbol{X}_n = \frac{1}{n}\boldsymbol{C}_n \to \boldsymbol{C}$ as $n \to \infty$. For any $P_\lambda(\cdot)$, if there exists a λ_n where $\hat{\boldsymbol{\beta}}_{\lambda_n} \to_p c\xi^*$ for some $c > 0$, then $\hat{\alpha}_{\lambda_n}\hat{\boldsymbol{\beta}}_{\lambda_n} \to_p \boldsymbol{\beta}^*$.*

We hypothesize that the rate of convergence for the α-Modified estimators is generally faster than the unmodified versions, but leave the proof for future investigation.

2.2.3 α-Modified Cross Validation

In addition to AR2 CV, we propose α-Modified CV based on average prediction error under the α-Modified estimates. Returning to the simulated data example from Section 2.1, the λ chosen through α-Modified CV with a 1SE Rule returns a model with $\mathcal{M}_\lambda = \mathcal{M}^*$. Its APE is 0.3821, which is marginally larger than that of AR2 CV. This demonstrates the possibility that the two CV strategies may point to different optimal λ.

One benefit of α-Modified CV over traditional CV surprisingly derives from the potential drawbacks of the modification. As the modification unshrinks the $\hat{\boldsymbol{\beta}}_\lambda$ without changing its direction, the modification may exacerbate an estimate of poor quality, causing its corresponding APE to be greater than APE based on $\hat{\boldsymbol{\beta}}_\lambda$. Similarly, when $\hat{\boldsymbol{\beta}}_\lambda$ recovers $\boldsymbol{\xi}^*$ but has small magnitude, the modified APE will decrease significantly over the traditional APE. Theorem 3 gives a theoretical justification for the latter situation.

Theorem 3. *Suppose that $\hat{\boldsymbol{\beta}}_\lambda$ recovers the direction of $\boldsymbol{\beta}^*$ and assume all ϵ_i are independent with constant variance σ^2. Then the expected value of the α-Modified APE is less than the expected value of the traditional APE when:*

$$\frac{\alpha^2}{E[(\|\hat{\boldsymbol{\beta}}_\lambda\|_2 - \alpha)^2]} \leq \frac{\boldsymbol{\beta}^{*T}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}^*}{\sigma^2} \ . \tag{2.5}$$

The α-Modified APE is then expected to be smaller than the traditional APE provided the signal-to-noise ratio, represented by $\boldsymbol{\beta}^{*T}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}^*/\sigma^2$, is sufficiently large.

2.3 α-Modified Lasso Estimates

The methods in Section 2.2 generalize to many types of penalties, but to better understand their properties we must focus on a specific penalty. Due to its popularity, we explore the properties for the Lasso penalty and finite n. First, note that the α-Modified Lasso may be thought of as a reverse Nonnegative Garotte in that it starts with shrunken estimates and uses a least squares modification to readjust and “un-shrink" them. The use of correlation to assess model quality is also consistent with the premise of the Covariance Test. The α-Modified Lasso is most similar to the Relaxed Lasso. The α-Modification penalty is $\lambda\alpha^{-1}||\alpha\boldsymbol{\xi}||_1$ and the Relaxed Lasso penalty is $\lambda_1\lambda_2||\boldsymbol{\beta}||_1$ for $\lambda_2 \in (0, 1]$. Theorem 1 establishes $\alpha^{-1} \in (0, 1]$ but α is estimated while the Relaxed Lasso treats λ_2 as a tuning parameter. While estimating $\hat{\alpha}_\lambda$ reduces computations, the possible estimators are less flexible than the Relaxed Lasso. Specifically, the α-Modification cannot change the direction of $\hat{\boldsymbol{\beta}}_\lambda$.

Figure 2.2 illustrates the distinction between the α-Modified and Relaxed Lasso estimator for an orthogonal $\mathbf{X}$ with $n = 100$, $p = 50$ and $p^* = 2$ where $(\beta_1^*, \beta_2^*) = (-6, 6)$. The α-Modified Lasso and Relaxed Lasso solutions were generated for a set of 100 λ_1 and 20 λ_2 values. Figure 2.2(a) simultaneously shows the original Lasso estimates and the α-Modified estimates for (β_1^*, β_2^*) across all λ values. Figure 2.2(b) shows the Relaxed Lasso estimates for (β_1^*, β_2^*) at a fixed λ_1 and across the λ_2 values. Note how the α-Modified estimates amplify the Lasso estimates in the same direction, while the Relaxed Lasso is able to change the direction of the original Lasso estimate. However, the α-Modified estimates do seem to approximate the Relaxed Lasso estimates as λ_1 is varied.

(a) Lasso estimates (solid) and α-Modified estimates (dashed) for multiple λ_1.

(b) Relaxed Lasso estimates for multiple values of λ_2 at a fixed $\lambda_1 = 0.429$.

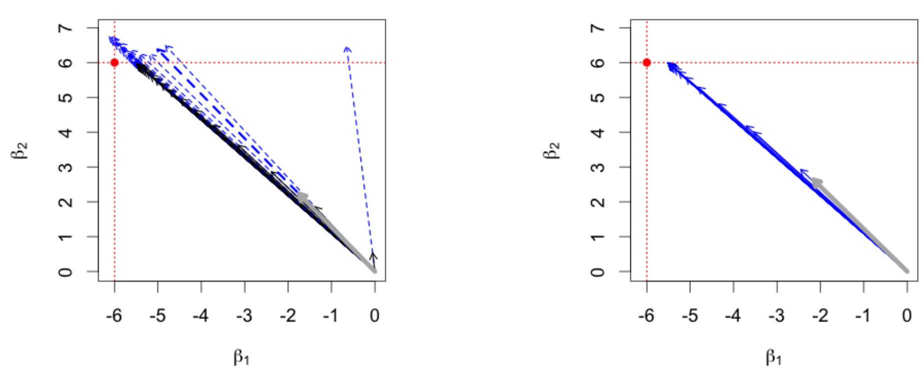

Figure 2.2: Lasso, Modified Lasso, and Relaxed Lasso estimates for two active coefficients in simulated data. The dot indicates the true values of the coefficients, and estimates at $\lambda_1 = 0.429, \lambda_2 = 1$ drawn in grey. Shorter vectors correspond to larger tuning parameters.

We now derive properties of the α-Modified estimator for orthogonal $\mathbf{X}$. The Lasso estimator for such $\mathbf{X}$ is $\hat{\beta}_{\lambda,j} = \text{sign}(\hat{\beta}_{OLS,j})(|\hat{\beta}_{OLS,j}| - \lambda)_+$, where $\hat{\beta}_{OLS,j}$ is the OLS estimate of the β_j^* and $(z)_+ = \max(0, z)$. If the Lasso estimate for a j^* is non-zero, the closed form expression for $\hat{\alpha}_\lambda \hat{\beta}_{\lambda,j^*}$ may be derived.

Lemma 3. *When $\boldsymbol{X}$ is orthogonal, the α-Modified Lasso estimator is:*

$$\hat{\alpha}_\lambda \hat{\beta}_{\lambda,j^*} = w_1 \hat{\beta}_{OLS,j^*} + (1 - w_1)\hat{\beta}_{j^*,\lambda} + w_2 \hat{\beta}_{j^*,\lambda} \tag{2.6}$$

where $w_1 = \frac{d_{j^*}^2}{\sum_{j=1}^p d_j^2}$, $w_2 = \frac{\lambda \sum_{j \neq j^*} d_j}{\sum_{j=1}^p d_j^2}$, and $d_j = (|\hat{\beta}_{OLS,j}| - \lambda)_+$.

If $d_{j^*} = 0$, then $\hat{\alpha}_\lambda \hat{\beta}_{\lambda,j^*} = \hat{\beta}_{\lambda,j^*} = 0$. If $\hat{\alpha}_\lambda \hat{\beta}_{\lambda,j} = 0$ for all $j \neq j^*$ and $d_{j^*} > 0$ then $\hat{\alpha}_\lambda \hat{\beta}_{\lambda,j^*} = \hat{\beta}_{OLS,j}$ which is consistent with Lemma 2. When $|\mathcal{M}_\lambda| > 1$, (2.6) involves a convex combination of the OLS and Lasso estimates, as well as an additive term, $w_2 \hat{\beta}_{j^*,\lambda}$, that may overcorrect the α-Modified Lasso beyond the OLS estimate. This behavior is illustrated for a simple example in Figure 2.3. In that example, for $\lambda < 10$ there are potential values of $\hat{\beta}_{OLS,j^*}$ where the α-Modified Lasso exceeds $\hat{\beta}_{OLS,j^*}$. To better understand this behavior, Theorem 4 provides an upper bound for $|\hat{\alpha}_\lambda \hat{\beta}_{\lambda,j^*} - \hat{\beta}_{OLS,j^*}|$.

Theorem 4. *Suppose $\boldsymbol{X}$ is orthogonal and consider a given predictor, j^*, and λ where at least one $j \neq j^*$ has $|\hat{\beta}_{OLS,j}| > \lambda$. Then across all possible $\hat{\beta}_{OLS,j^*}$,*

$$|\hat{\alpha}_\lambda \hat{\beta}_{\lambda,j^*} - \hat{\beta}_{OLS,j^*}| \leq \lambda \times \max\left(1, \frac{\sqrt{u^2 v + v^2} - v}{2v}\right) \tag{2.7}$$

where $u = \sum_{j \neq j^*} d_j$ and $v = \sum_{j \neq j^*} d_j^2$. Moreover, $\lim_{|\hat{\beta}_{OLS,j^*}| \to \infty} |\hat{\alpha}_\lambda \hat{\beta}_{j^*,\lambda} - \hat{\beta}_{OLS,j^*}| \to 0$.

Based on Theorem 4, we see that when $(\sqrt{u^2 v + v^2} - v)/2v > 1$ there exists some $\hat{\beta}_{OLS,j^*}$ in which the α-Modified estimate blows up $\hat{\beta}_{\lambda,j^*}$, thereby introducing bias in the opposite direction. This occurs as u^2/v increases, specifically when $u^2/v > 8$, as can be

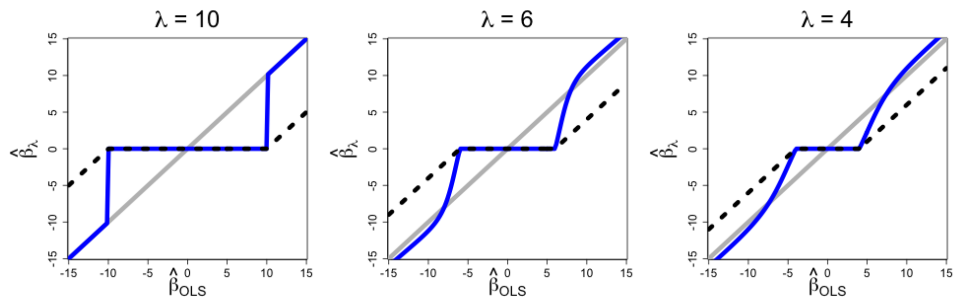

Figure 2.3: α-modified estimates of $\alpha\beta_1$ as $\hat{\beta}_{OLS,1}$ changes, all other parameters being fixed. $\hat{\beta}_{OLS,j} = [-8, 5, -3, 1]$ for $j = [2, ..., 5]$. The dotted line gives the Lasso estimates and the solid line gives the α-modified estimates.

easily shown. When u^2/v is large, multiple d_j are nonzero but close to 0. This will likely occur for relatively large λ values and certain patterns of $\boldsymbol{\beta}^*$.

To generalize Theorem 4 to an arbitrary $\mathbf{X}$ we condition on the event that λ recovers the sign vector of $\boldsymbol{\beta}^*$. Theorem 5 shows that the absolute difference between the α-Modified Lasso estimate and the OLS estimate generally do not approach 0 as $|\hat{\boldsymbol{\beta}}_{OLS,j^*}| \to \infty$. Rather, they approach a non-zero constant that is always smaller than $|\hat{\beta}_{\lambda,j^*} - \hat{\beta}_{OLS,j^*}|$, indicating the α-Modified estimate improves $\hat{\beta}_{\lambda,j^*}$.

Theorem 5. *Suppose the Lasso recovers the correct sign vector of $\boldsymbol{\beta}^*$. Let $\boldsymbol{s}_1$ be the sign vector of active coefficients, and let $\boldsymbol{X}_1$, the active subset of $\boldsymbol{X}$, have full column rank. Let $G = n\lambda\Big(s_{j^*} - [(\boldsymbol{X}_1^T\boldsymbol{X}_1)^{-1}]_{j^*}\boldsymbol{s}_1\Big)$. As $|\hat{\beta}_{OLS,j^*}| \to \infty$ for $\hat{\beta}_{\lambda,j^*} \neq 0$:*

$$|\hat{\alpha}_\lambda \hat{\beta}_{\lambda,j^*} - \hat{\beta}_{OLS,j^*}| \to G < |\hat{\beta}_{\lambda,j^*} - \hat{\beta}_{OLS,j^*}| \ . \tag{2.8}$$

Theorems 4 and 5 confirm that the α-Modification cannot generally guarantee bias reduction. However, a poor bias adjustment may be detected by CV. To illustrate, a simulation study was performed for the $\mathbf{X}$ used in Figure 2.1, considering two $\boldsymbol{\beta}^*$ vectors. For each $\boldsymbol{\beta}^*$, 500 $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_n)$ were generated and the CV results were averaged across the replications, including APE, α-Modified APE, and AR2. The results are presented in

(a) $\boldsymbol{\beta}^* = (4.3, 2.9, -5.8, -3.3, -6.1, 0, ..., 0)^T$.   **(b)** $\boldsymbol{\beta}^* = (50, 50, 50, 50, 50, 0, ..., 0)^T$.

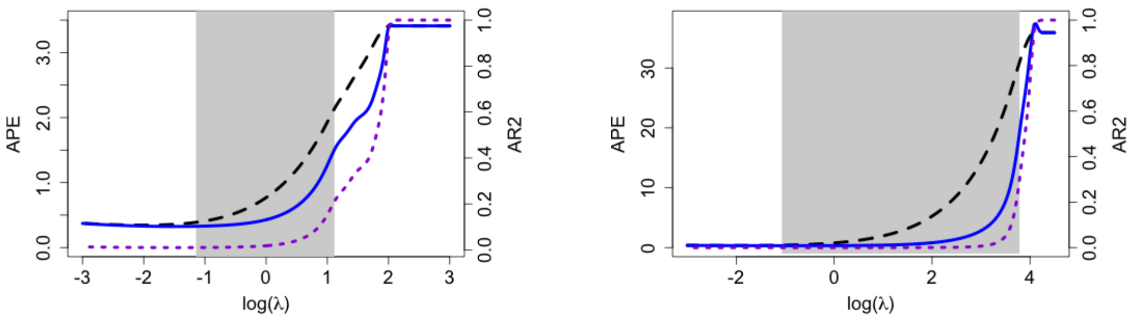

Figure 2.4: Average 10-fold Cross Validation results across 500 replications of linear data. $\mathbf{X}$ was taken from the example in Section 1. Average values of λ where the correct submodel was selected are represented by the grey box, the dashed line gives the APE, the solid line gives α-Modified APE, and the dotted line gives Average R^2. Standard errors from these simulations were too small to be depicted.

Figure 2.4. In Figure 2.4(a) the α-Modified APE is less than or equal to APE for all λ, and especially so for larger λ that recover $\mathcal{M}^*$. The optimal λ according to the minimum for both AR2 and α-Modified APE result in sparser models than the minimum APE. With a 1SE rule, traditional CV recovered the support in only 24.4% of the replications, whereas α-Modified and AR2 CV did so in 91.6% and 93.6% of the replications, respectively.

Figure 2.4(b) corresponds to $\boldsymbol{\beta}^*$ with larger and equal magnitude coefficients. Generally the α-Modified APE is equal to or smaller than regular APE except for $\log(\lambda) \approx 4$. These models exhibit false negatives and $\hat{\alpha}_\lambda$ tends to overamplify the few active coefficients in the wrong direction, leading to poor prediction. Fortunately, the α-Modified APE highlights this issue and will not choose such λ. The support recovery percentages were 19.8% for APE CV, 98.2% for α-Modified CV, and 98.8% for AR2 CV. Surprisingly, the support recovery percentage decreased for APE CV despite a higher signal to noise ratio.

2.4 Numerical Results

To evaluate the new methods for the Lasso, a simulation study consisting of 100 replications of data from Model (1.1) was conducted. Following Meinshausen (2007), the rows of $\mathbf{X}$ were drawn from a multivariate Normal distribution with mean $\mathbf{0}$ and covariance $\mathbf{\Sigma}_X$. For each $\mathbf{X}$, nonzero elements of $\boldsymbol{\beta}^*$ were drawn from a Gamma distribution with shape 10 and scale 0.25 with negative and positive signs randomly assigned with equal probability. For each $\mathbf{X}$ and $\boldsymbol{\beta}^*$, two independent error vectors were generated from $N(0, \sigma^2)$ distributions, with σ determined by a fixed signal-to-noise ratio SNR $= \boldsymbol{\beta}^{*T}\mathbf{\Sigma}\boldsymbol{\beta}^*/\sigma^2$.

Each simulated data set was analyzed using 10-fold CV with and without a 1SE rule under traditional APE (CV Min, CV 1SE), AR2 (AR2 Min, AR2 1SE), α-Modified APE (Mod Min, Mod 1SE), and the Relaxed Lasso. Meinshausen (2007) recommended choosing tuning parameters for the Relaxed Lasso based on the minimum APE. This was implemented as well as a 1SE Rule based on Stallrich et al. (2020), wherein the optimal combination of λ_1 and λ_2 is the model with the smallest $||\hat{\boldsymbol{\beta}}_{\lambda_1,\lambda_2}||_1$ among all models within one standard error of the minimum. Relaxed Lasso estimates were found using a coordinate descent algorithm, capable of admitting more than $n-1$ covariates into models, unlike LARS which was used by Meinshausen (2007). In the following results, variable selection is evaluated using the Hamming Distance (HD) between $\boldsymbol{\beta}^*$ and $\hat{\boldsymbol{\beta}}_\lambda$, which is the number of false negatives plus the number of false positives. Additional metrics including the false discovery rate (FDR), average number of false positives and negatives, and the average prediction bias, where prediction bias is $||\mathbf{X}\boldsymbol{\beta}^* - \mathbf{X}\hat{\boldsymbol{\beta}}_\lambda||_2$ or $||\mathbf{X}\boldsymbol{\beta}^* - \mathbf{X}\hat{\alpha}_\lambda\hat{\boldsymbol{\beta}}_\lambda||_2$ for the α-Modification, can be found in the supplementary material.

2.4.1 Simulation Study for Independent Predictors

We first consider the case of $\boldsymbol{\Sigma}_X = \boldsymbol{I}_p$. Tables 2.1, 2.2, and 2.3 give the average Hamming distance between $\boldsymbol{\beta}^*$ and $\hat{\boldsymbol{\beta}}_\lambda$ for the traditional APE CV technique with a 1SE Rule, AR2 CV with a 1SE rule, and the Relaxed Lasso with a minimum APE CV approach. AR2 CV and Mod APE CV perform fairly similarly; the latter results may be found in the supplementary materials. In general, AR2 CV has better variable selection than APE CV and, for small n and large p^*, the Relaxed Lasso.

Table 2.1: Average Hamming Distance for 100 replications of CV 1SE. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100	3.1	4.3	6	7.8	9.1	2.2	4	7.5	9.3	10.7
	(0.3)	(0.4)	(0.6)	(1.1)	(1.2)	(0.2)	(0.4)	(1.2)	(1)	(1.2)
500	0.6	1	1.3	1.9	2.9	0.9	0.9	1.7	1.7	1.5
	(0.1)	(0.2)	(0.2)	(0.5)	(0.5)	(0.2)	(0.1)	(0.3)	(0.3)	(0.3)
1000	0.2	0.4	0.4	0.6	0.9	0.1	0.4	0.5	0.7	0.9
	(0)	(0.1)	(0.1)	(0.1)	(0.2)	(0)	(0.1)	(0.1)	(0.1)	(0.2)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100	31.7	43.5	50.1	51.1	53.2	5	32	47	53.8	54.5
	(1.3)	(0.6)	(0.5)	(0.4)	(1.1)	(0.4)	(0.6)	(0.7)	(1.1)	(1.1)
500	1.6	18.2	33.3	48.6	57.4	0 (0)	16.8	35.3	58.4	81.2
	(0.2)	(0.5)	(0.8)	(1.3)	(1.8)		(0.5)	(0.9)	(1.6)	(2)
1000	0.4	14.5	28.5	43.5	57.6	0 (0)	14.2	28	44.8	64.2
	(0.1)	(0.4)	(0.9)	(1.4)	(1.6)		(0.5)	(0.8)	(1.3)	(1.8)

Results from increasing values of p^* were considered for $n = \{100, 500, 1000\}$ and $p = 100$ in Figure 2.5. Additional methods and results may be found in the appendix. For $n = 100$ all methods perform comparably, except for the Relaxed Lasso with a minimum APE rule, which tends to have a higher number of false positives than the other methods,

Table 2.2: Average Hamming Distance for 100 replications of AR2 CV 1SE. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100		2.1 (0.3)	2.8 (0.3)	3.3 (0.3)	4.9 (0.7)	5 (0.5)	0.6 (0.1)	0.6 (0.1)	1.9 (0.8)	1.8 (0.4)	2.6 (0.4)
500		0.1 (0)	0.2 (0)	0.1 (0)	0.2 (0)	0.4 (0.1)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
1000		0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100		23.6 (1.3)	40.4 (0.5)	51.4 (0.8)	55.1 (1.2)	54.8 (1.2)	4.2 (0.4)	31.9 (0.5)	47.9 (0.8)	54.6 (1.3)	53.5 (0.9)
500		1.8 (0.2)	17.1 (0.5)	28.7 (0.8)	42.3 (1.3)	51.9 (2.2)	0.2 (0)	9.1 (0.4)	17.8 (0.7)	30 (1)	46.6 (1.3)
1000		0.7 (0.1)	9.1 (0.4)	15.4 (0.6)	23.1 (0.9)	32.5 (1.1)	0 (0)	3.9 (0.3)	6.3 (0.4)	11.1 (0.6)	16.9 (0.8)

leading to larger average HD. The result of this higher rate of false positives is a lower rate of false negatives; the balance however, as represented by the average Hamming Distance, does not favor the Relaxed Min. As n increases, α-Modified and AR2 CV appear to perform comparably to the Relaxed Lasso with a 1SE Rule, and have a consistently smaller average Hamming Distance than traditional CV with a 1SE Rule. The new methods of CV consistently perform similarly to Relaxed 1SE and out performs the CV 1SE in terms of variable selection.

Unlike the Relaxed Lasso, the α-Modification is straightforward to apply to other penalties. The α-Modified Lasso was compared with the non-convex penalties SCAD and MCP, both traditionally and with an α-Modification. As expected, the α-Modification had a minimal effect on the predictive models. MCP and SCAD estimates were prone to underselection whereas the α-Modified Lasso estimates were more prone to overselection. Details may be found in Appendix A.1.

Table 2.3: Average Hamming Distance for 100 replications of Relaxed Lasso. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100	3.7	5.4	7.4	18.4	28.1	1.1	1.8	4.5	6.6	7.9
	(0.4)	(0.8)	(1.1)	(3.4)	(4.2)	(0.3)	(0.5)	(1.7)	(2.3)	(2.6)
500	0.8	1.2	0.5	0.7	1.5	1.2	0.8	0.8	0.3	0.1
	(0.3)	(0.3)	(0.2)	(0.4)	(0.6)	(0.3)	(0.2)	(0.4)	(0.1)	(0)
1000	0.6	0.5	0.7	0.4	1.9	0.6	0.8	0.4	0.4	0.4
	(0.2)	(0.2)	(0.3)	(0.1)	(1.5)	(0.2)	(0.3)	(0.2)	(0.2)	(0.2)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100	19.5	39.9	57.6	70.8	67.9	2.1	34.1	75.9	87.6	93.6
	(1.3)	(0.5)	(1.7)	(3.1)	(3.2)	(0.3)	(0.5)	(2.5)	(3.3)	(3.8)
500	0.9	22.9	39.6	66.2	74.5	0.1	7.2	14.8	25.4	36.8
	(0.1)	(0.9)	(1.8)	(3.4)	(4.2)	(0)	(0.5)	(1.2)	(2.3)	(2.7)
1000	0.2	12.3	17.5	27.9	32.7	0 (0)	2.4	3.1	4.9	5.5
	(0)	(0.9)	(1.4)	(2.7)	(2.9)		(0.3)	(0.3)	(0.5)	(0.3)

2.4.2 Simulation Study for Correlated Predictors

Here the $\mathbf{\Sigma}_X$ satisfied $\Sigma_{ii} = 1$ and $\Sigma_{ij} = 0.75$ whenever $i \neq j$. Figure 2.6 gives the average Hamming distance for 100 replications with $p = 100$ and $n = \{100, 500, 1000\}$, but further results can be found in Appendix A.1. The two new methods of CV are highly competitive with the Relaxed 1SE, whereas Relaxed Min has a tendency to overselect as p^* increases, particularly for $n = 100$. Similarly to the results from independent predictors, it is clear that as n increases, the average Hamming distance for the Relaxed Lasso and the two new methods of CV decreases for all considered p^*. The Relaxed approach with a minimum APE rule struggles when n is small, even for a high signal-to-noise ratio, primarily due to overselection. Once again the new methods of CV generally have a smaller average Hamming distance than CV with a 1SE Rule and are similar in variable selection to the Relaxed Lasso with a 1SE Rule.

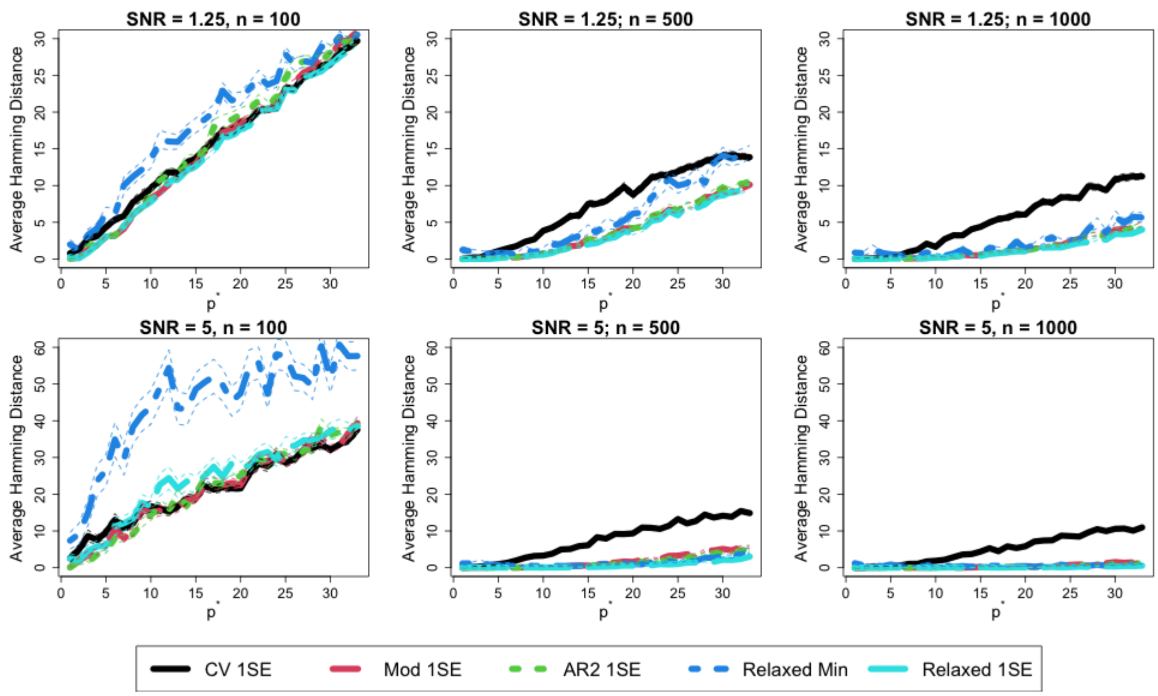

Figure 2.5: Average Hamming Distance between $\boldsymbol{\beta}^*$ and $\hat{\boldsymbol{\beta}}_\lambda$ from 100 replications of simulated data with $n = \{100, 500, 1000\}$ and $p = 100$ with independent predictors for both SNR $= 1.25$ and SNR $= 5$. Thin dotted lines represent the mean $\pm$ one standard error.

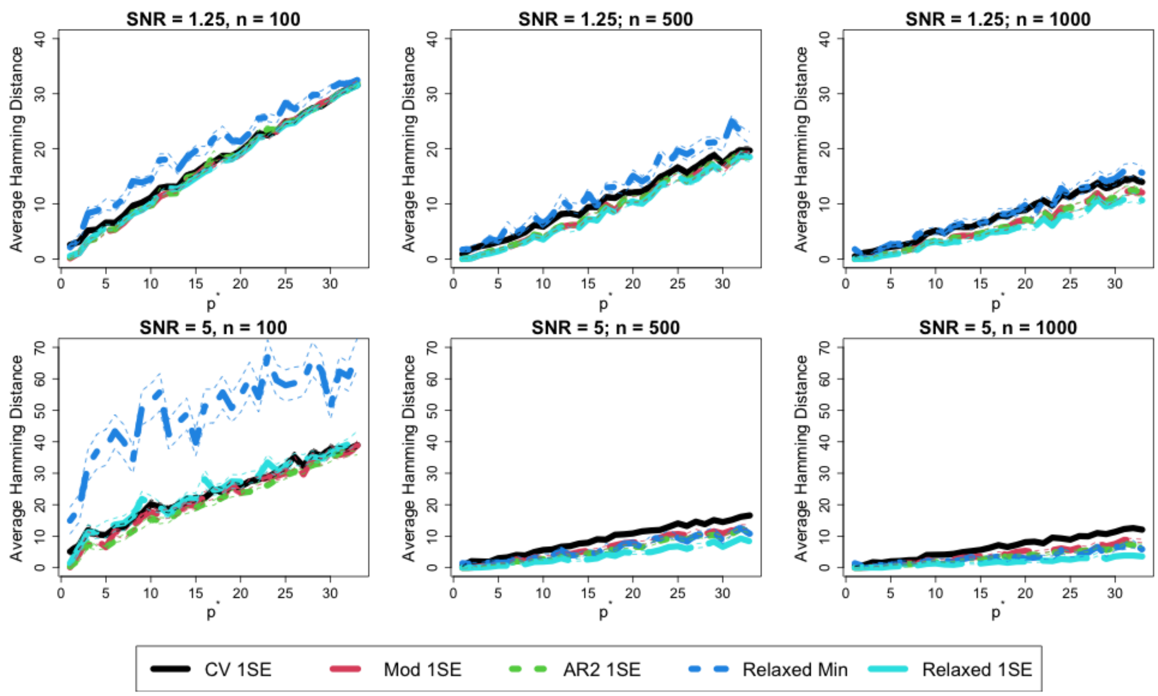

Figure 2.6: Average Hamming Distance between $\boldsymbol{\beta}^*$ and $\hat{\boldsymbol{\beta}}_\lambda$ from 100 replications of simulated data with $n = \{100, 500, 1000\}$ and $p = 100$ for correlated predictors. Thin dotted lines represent the mean $\pm$ one standard error.

2.5 Application to EMG Data

The goal of this application, introduced in Section 1.3, is to compare the new CV methods of tuning parameter selection to traditional CV in translating EMG signals into the movement of a prosthetic limb. Recall that redundancies in the information provided by the predictor variables mean the ideal model size is only two (X_{12} and ether X_5 or X_7), but models containing X_5, X_7, and X_{12} are also considered acceptable. The model used for the application is the scalar-on-function linear model given in (1.21) and the penalty for smootheness and sparsity is shown in (1.22). Recall that the high correlation among predictor variables can make variable selection difficult.

Table 2.4: Variable selection results for EMG finger movements without adaptive weighting. FP indicates the total number of false positives in the model, and Size is the total number of EMG signals contained in the model.

		FC1	FC2	FC3	FR1	FR2	FR3	Mean
APE	FP	2	1	2	2	1	1	1.5
	Size	4	3	4	5	4	4	4
AR2	FP	0	1	3	0	1	1	1
	Size	2	3	6	2	4	3	3.33
Mod	FP	1	1	2	2	1	1	1.33
	Size	3	3	4	5	4	4	3.83

Table 2.4 gives the variable selection results for the three CV methods based on the initial stage; results from subsequent stages of SAFE can be found in Appendix A.2. AR2 CV and Mod APE generally give smaller models than traditional APE CV, with one exception: the AR2 method has a large model size for FC3. That said, however, AR2 CV is also able to perform ideal variable selection without any adaptive weighting for both FC1 and FR1.

On average, both new methods have fewer false positives and smaller model sizes.

Mod APE gives very similar results to APE, suggesting that the α-Modified CV approach requires further study for the Group Lasso. Although AR2 CV is not perfect in this application, in general it reduces model size and decreases false positives at no additional computational cost.

2.6 Discussion

In this chapter, AR2 CV was proposed as a method of tuning parameter selection to balance support recovery and prediction performance. This led to the α-Modification, a multiplicative adjustment to predictions from penalized estimates which can also be used for α-Modified CV. The α-Modification is simple and efficient, making it an attractive option when less flexible approaches are unavailable. A simulation study on the capabilities of AR2 and α-Modified CV found that their variable selection results were highly competitive with—or, in some cases, better than—the Relaxed Lasso. In order to ensure fair comparison, a 1SE Rule for the Relaxed Lasso was introduced. Finally, the approaches were applied to a functional data problem in a demonstration of their flexibility.

The α-Modification and the tuning parameter selection methods proposed here inspire several research questions. First, further theoretical analysis of the methods is of interest. Because the two new methods of tuning parameter selection are CV-based approaches, the theoretical properties of CV are central. Theoretical justifications for the use of CV for penalized estimators are relatively new and still evolving. Chetverikov et al. (2021) may be extended to show that α-Modified and AR2 CV lead to estimates that are low bias and appropriately sparse. A second area of future research is the theoretical properties of α-Modified estimates themselves. In Theorem 2, it was shown that any consistent estimator will still be consistent with the α-Modification. We posit that the rates of convergence for α-Modified estimators are faster than unmodified, but proof of this conjecture is the work of future research. Similarly, many of the results from this paper assume finite sample sizes. Further study is needed to determine more of the asymptotic properties of α-Modified estimators and to adapt the specific results given for the Lasso penalty to other penalties.

There are also a few extensions and adaptations of the α-Modification that may prove fruitful. Expanding the α-Modification to Generalized Linear Models (GLMs) is the subject of Chapter 3. This extension requires an iterative algorithm to find estimates of α because closed form solutions do not exist and an accommodation for the inclusion of an intercept term will be necessary. Additionally, as noted in some of the theoretical results in this paper, the α-Modification does not always reduce bias. Chapter 3 also explores a further penalty on α itself to ensure bias reduction. Finally, the calculation of $\hat{\alpha}_\lambda$ described here uses in-sample predictions and observations. As overspecification is a particular concern, the question of whether out-of-sample data can be used to find estimates of α is another subject of further research.

2.A Proofs for Theoretical Results

This section includes the proofs of all theoretical results presented in Chapter 2.

Proof of Theorem 1. Let $\hat{\boldsymbol{\beta}}_\lambda \neq \mathbf{0}$ and $g(\boldsymbol{\beta}) = \sum_{\ell=1}^{L} \lambda_\ell g_\ell(\boldsymbol{\beta})$. Denote the subgradient vector of $g(\boldsymbol{\beta})$ by $\nabla \mathbf{g}^*$. Then the KKT conditions give $\frac{1}{n}\mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_\lambda) = \nabla \mathbf{g}^*$ and there exists a generalized inverse $(\mathbf{X}^T\mathbf{X})^-$ where $\hat{\boldsymbol{\beta}}_\lambda = (\mathbf{X}^T\mathbf{X})^-(\mathbf{X}^T\mathbf{y} - n\nabla \mathbf{g}^*)$. Therefore:

$$\hat{\mathbf{y}}_\lambda^T \mathbf{y} = \hat{\mathbf{y}}_\lambda^T \hat{\mathbf{y}}_\lambda + n\nabla \mathbf{g}^{*T}(\mathbf{X}^T\mathbf{X})^-\mathbf{X}^T\mathbf{y} - n^2\nabla \mathbf{g}^{*T}(\mathbf{X}^T\mathbf{X})^-\nabla \mathbf{g}^* \ . \tag{2.9}$$

Given this definition, $\hat{\alpha}_\lambda$ may be written:

$$\begin{aligned}
\hat{\alpha}_\lambda &= \frac{\hat{\mathbf{y}}_\lambda^T \hat{\mathbf{y}}_\lambda + n\nabla \mathbf{g}^{*T}(\mathbf{X}^T\mathbf{X})^-(\mathbf{X}^T\mathbf{y} - n\nabla \mathbf{g}^*)}{\hat{\mathbf{y}}_\lambda^T \hat{\mathbf{y}}_\lambda} && (2.10)\\
&= \frac{\hat{\mathbf{y}}_\lambda^T \hat{\mathbf{y}}_\lambda + n\nabla \mathbf{g}^{*T}\hat{\boldsymbol{\beta}}_\lambda}{\hat{\mathbf{y}}_\lambda^T \hat{\mathbf{y}}_\lambda} && (2.11)\\
&= 1 + \frac{n\nabla \mathbf{g}^{*T}\hat{\boldsymbol{\beta}}_\lambda}{\hat{\mathbf{y}}_\lambda^T \hat{\mathbf{y}}_\lambda}. && (2.12)
\end{aligned}$$

Because the denominator of the second term is greater than 0, as is n, this expression is greater than or equal to 1 whenever $\nabla \mathbf{g}^{*T}\hat{\boldsymbol{\beta}}_\lambda$ is greater than or equal to zero. The vector $\nabla \mathbf{g}^*$ the subgradient of a convex function at $\hat{\boldsymbol{\beta}}_\lambda$, therefore for any $\boldsymbol{\beta}$:

$$g(\boldsymbol{\beta}) \geq g(\hat{\boldsymbol{\beta}}_\lambda) + \nabla \mathbf{g}^{*T}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_\lambda) \tag{2.13}$$

$$\nabla \mathbf{g}^{*T}\hat{\boldsymbol{\beta}}_\lambda \geq g(\hat{\boldsymbol{\beta}}_\lambda) - g(\boldsymbol{\beta}) + \nabla \mathbf{g}^{*T}\boldsymbol{\beta}. \tag{2.14}$$

Because this is true for all $\boldsymbol{\beta}$ vectors, it must be the case when $\boldsymbol{\beta} = \mathbf{0}$. Hence, we have:

$$\nabla \mathbf{g}^{*T}\hat{\boldsymbol{\beta}}_\lambda \geq g(\hat{\boldsymbol{\beta}}_\lambda) - g(\mathbf{0}) \geq 0. \tag{2.15}$$

Thus $\hat{\alpha}_\lambda \geq 1$. $\square$

Proof of Lemma 1. When $\|\hat{\boldsymbol{\beta}}_\lambda\|_2^{-1}\hat{\boldsymbol{\beta}}_\lambda = \boldsymbol{\xi}^*$, $\hat{\boldsymbol{\beta}}_\lambda = \|\hat{\boldsymbol{\beta}}_\lambda\|_2\boldsymbol{\xi}^*$. Expand the response $\mathbf{y} = \alpha\mathbf{X}\boldsymbol{\xi}^* + \boldsymbol{\epsilon}$, to define:

$$\hat{\alpha}_\lambda = \frac{\alpha\boldsymbol{\xi}^{*T}\mathbf{X}^T\mathbf{X}\boldsymbol{\xi}^* + \boldsymbol{\xi}^*\mathbf{X}^T\boldsymbol{\epsilon}}{\|\hat{\boldsymbol{\beta}}_\lambda\|_2\boldsymbol{\xi}^{*T}\mathbf{X}^T\mathbf{X}\boldsymbol{\xi}^*} \ . \tag{2.16}$$

Therefore:

$$\begin{aligned}
\hat{\alpha}_\lambda\hat{\boldsymbol{\beta}}_\lambda &= \hat{\alpha}_\lambda\|\hat{\boldsymbol{\beta}}_\lambda\|_2\boldsymbol{\xi}^* &(2.17)\\
&= \frac{1}{\|\hat{\boldsymbol{\beta}}_\lambda\|_2}\Big(\alpha + \frac{\boldsymbol{\xi}^{*T}\mathbf{X}^T\boldsymbol{\epsilon}}{\boldsymbol{\xi}^{*T}\mathbf{X}^T\mathbf{X}\boldsymbol{\xi}^*}\Big)\|\hat{\boldsymbol{\beta}}_\lambda\|_2\boldsymbol{\xi}^* &(2.18)\\
&= \alpha\boldsymbol{\xi}^* + \Big(\frac{\boldsymbol{\xi}^{*T}\mathbf{X}^T\boldsymbol{\epsilon}}{\boldsymbol{\xi}^{*T}\mathbf{X}^T\mathbf{X}\boldsymbol{\xi}^*}\Big)\boldsymbol{\xi}^* \ . &(2.19)
\end{aligned}$$

Thus $E[\hat{\alpha}_\lambda\hat{\boldsymbol{\beta}}_\lambda] = \alpha\boldsymbol{\xi}^* = \boldsymbol{\beta}^*$ and the estimator is unbiased. $\square$

Proof of Lemma 2. Let $\hat{\boldsymbol{\beta}}_\lambda = a\hat{\boldsymbol{\beta}}_{OLS}^{\mathcal{M}_\lambda}$ where $\hat{\boldsymbol{\beta}}_{OLS}^{\mathcal{M}_\lambda}$ is the OLS estimate for submodel $\mathcal{M}_\lambda$. Let $\mathbf{X}_{\mathcal{M}_\lambda}$ be the submatrix of $\mathbf{X}$ containing only the columns indexed by $\mathcal{M}_\lambda$. Then $\hat{\boldsymbol{\beta}}_{OLS}^{\mathcal{M}_\lambda} = (\mathbf{X}_{\mathcal{M}_\lambda}^T\mathbf{X}_{\mathcal{M}_\lambda})^{-1}\mathbf{X}_{\mathcal{M}_\lambda}^T\mathbf{y}$. Now:

$$\hat{\alpha}_\lambda = \frac{a\mathbf{y}^T\mathbf{X}_{\mathcal{M}_\lambda}(\mathbf{X}_{\mathcal{M}_\lambda}^T\mathbf{X}_{\mathcal{M}_\lambda})^{-1}\mathbf{X}_{\mathcal{M}_\lambda}^T\mathbf{y}}{a^2\mathbf{y}^T\mathbf{X}_{\mathcal{M}_\lambda}(\mathbf{X}_{\mathcal{M}_\lambda}^T\mathbf{X}_{\mathcal{M}_\lambda})^{-1}\mathbf{X}_{\mathcal{M}_\lambda}^T\mathbf{X}_{\mathcal{M}_\lambda}(\mathbf{X}_{\mathcal{M}_\lambda}^T\mathbf{X}_{\mathcal{M}_\lambda})^{-1}\mathbf{X}_{\mathcal{M}_\lambda}^T\mathbf{y}} = \frac{1}{a} \ . \tag{2.20}$$

Therefore:

$$\hat{\alpha}_\lambda\hat{\boldsymbol{\beta}}_\lambda = \frac{1}{a} \times a\hat{\boldsymbol{\beta}}_{OLS}^{\mathcal{M}_\lambda} = \hat{\boldsymbol{\beta}}_{OLS}^{\mathcal{M}_\lambda} \ . \tag{2.21}$$

$\square$

Proof of Theorem 2. Consider $\hat{\alpha}_{\lambda_n}$:

$$\hat{\alpha}_{\lambda_n} = \frac{\frac{1}{n}\hat{\boldsymbol{\beta}}_{\lambda_n}^T \mathbf{X}_n^T \mathbf{y}}{\frac{1}{n}\hat{\boldsymbol{\beta}}_{\lambda_n}^T \mathbf{X}_n^T \mathbf{X}_n \hat{\boldsymbol{\beta}}_{\lambda_n}} \ . \tag{2.22}$$

By Continuous Mapping Theorem, the denominator of this expression converges in probability to $c^2 \boldsymbol{\xi}^{*T} \mathbf{C} \boldsymbol{\xi}^*$. The numerator is equal to:

$$\frac{1}{n}\hat{\boldsymbol{\beta}}_{\lambda_n}^T \mathbf{X}_n^T \mathbf{X}_n \boldsymbol{\beta} + \hat{\boldsymbol{\beta}}_{\lambda_n}^T \mathbf{X}_n^T \boldsymbol{\epsilon} \to_p \alpha c \boldsymbol{\xi}^{*T} \mathbf{C} \boldsymbol{\xi}^* \ . \tag{2.23}$$

Then $\hat{\alpha}_{\lambda_n} \to_p \frac{\alpha}{c}$ and, once again by Continuous Mapping Theorem, $\hat{\alpha}_{\lambda_n} \hat{\boldsymbol{\beta}}_{\lambda_n} \to \frac{\alpha}{c} c \boldsymbol{\xi}^* = \alpha \boldsymbol{\xi}^* = \boldsymbol{\beta}^*$. $\square$

Proof of Theorem 3. Average Prediction Error for K-fold CV is calculated as:

$$\frac{1}{K}\sum_k \frac{1}{n_k}\sum_i (y_{k_i} - \mathbf{x}_{k_i}^T \hat{\boldsymbol{\beta}}_\lambda)^2. \tag{2.24}$$

To show that the expected value of α-Modified APE is less than that of regular APE, it is sufficient to show that $E[ModPE] < E[PE]$ where $PE = \sum_i (y_{k_i} - \mathbf{x}_{k_i}^T \hat{\boldsymbol{\beta}}_\lambda)^2$ and $ModPE = \sum_i (y_{k_i} - \hat{\alpha}_\lambda \mathbf{x}_{k_i}^T \hat{\boldsymbol{\beta}}_\lambda)^2$. When the direction of $\boldsymbol{\beta}^*$ has been captured by $\hat{\boldsymbol{\beta}}_\lambda$, we can write the estimate as $\|\hat{\boldsymbol{\beta}}_\lambda\| \boldsymbol{\xi}^*$. Expand $y_{k_i} = \alpha \mathbf{x}_{k_i}^T \boldsymbol{\xi}^* + \epsilon_{k_i}$ to find:

$$PE = \sum_i \left(\epsilon_{k_i} - \mathbf{x}_{k_i}^T \boldsymbol{\xi}^* (\|\hat{\boldsymbol{\beta}}_\lambda\|_2 - \alpha)\right)^2 \ . \tag{2.25}$$

Similarly,

$$ModPE = \sum_i \left(\epsilon_{k_i} - \mathbf{x}_{k_i}^T \boldsymbol{\xi}^* \Big(\frac{\boldsymbol{\xi}^{*T} \mathbf{X}^T \boldsymbol{\epsilon}}{\boldsymbol{\xi}^{*T} \mathbf{X}^T \mathbf{X} \boldsymbol{\xi}^*}\Big)\right)^2 \ . \tag{2.26}$$

Let Σ be the covariance matrix for $\boldsymbol{\epsilon}$, the error vector of the observations used to find $\hat{\boldsymbol{\beta}}_\lambda$. Therefore, $\Sigma = \sigma^2\mathbf{I}$

$$
\begin{aligned}
E[PE] &= n_k\sigma^2 + \sum_i (\mathbf{x}_{k_i}^T\boldsymbol{\xi}^*)^2 E[(\|\hat{\boldsymbol{\beta}}_\lambda\|_2 - \alpha)^2] && (2.27)\\
E[ModPE] &= E\left[\sum_i \left(\epsilon_{k_i} - \mathbf{x}_{k_i}^T\boldsymbol{\xi}^*\Big(\frac{\boldsymbol{\xi}^{*T}\mathbf{X}^T\boldsymbol{\epsilon}}{\boldsymbol{\xi}^{*T}\mathbf{X}^T\mathbf{X}\boldsymbol{\xi}^*}\Big)\right)^2\right] && (2.28)\\
&= n_k\sigma^2 + \sum_i \frac{(\mathbf{x}_{k_i}^T\boldsymbol{\xi}^*)^2 E[\boldsymbol{\epsilon}^T\mathbf{X}\boldsymbol{\xi}^*\boldsymbol{\xi}^{*T}\mathbf{X}^T\boldsymbol{\epsilon}]}{(\boldsymbol{\xi}^{*T}\mathbf{X}^T\mathbf{X}\boldsymbol{\xi}^*)^2} && (2.29)\\
&= n_k\sigma^2 + \sum_i \frac{(\mathbf{x}_{k_i}^T\boldsymbol{\xi}^*)^2 (tr(\boldsymbol{\xi}^{*T}\mathbf{X}^T\Sigma\mathbf{X}\boldsymbol{\xi}^*) + E[\boldsymbol{\epsilon}^T]\mathbf{X}\boldsymbol{\xi}^*\boldsymbol{\xi}^{*T}\mathbf{X}^T E[\boldsymbol{\epsilon}])}{(\boldsymbol{\xi}^{*T}\mathbf{X}^T\mathbf{X}\boldsymbol{\xi}^*)^2} && (2.30)\\
&= n_k\sigma^2 + \sum_i \frac{(\mathbf{x}_{k_i}^T\boldsymbol{\xi}^*)^2\sigma^2(\boldsymbol{\xi}^{*T}\mathbf{X}^T\mathbf{X}\boldsymbol{\xi}^*)}{(\boldsymbol{\xi}^{*T}\mathbf{X}^T\mathbf{X}\boldsymbol{\xi}^*)^2} && (2.31)\\
&= n_k\sigma^2 + \frac{\sigma^2}{\boldsymbol{\xi}^{*T}\mathbf{X}^T\mathbf{X}\boldsymbol{\xi}^*}\sum_i (\mathbf{x}_{k_i}^T\boldsymbol{\xi}^*)^2 && (2.32)
\end{aligned}
$$

Therefore $E[PE] \geq E[ModPE]$ whenever

$$
\begin{aligned}
E[(\|\hat{\boldsymbol{\beta}}_\lambda\|_2 - \alpha)^2] &\geq \frac{\sigma^2}{\boldsymbol{\xi}^{*T}\mathbf{X}^T\mathbf{X}\boldsymbol{\xi}^*} && (2.33)\\
\frac{\alpha^2}{E[(\|\hat{\boldsymbol{\beta}}_\lambda\|_2 - \alpha)^2]} &\leq \frac{\boldsymbol{\beta}^{*T}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}^*}{\sigma^2} && (2.34)
\end{aligned}
$$

because $\boldsymbol{\beta}^* = \alpha\boldsymbol{\xi}^*$. Therefore we have the result. $\square$

Proof of Lemma 3. When $\mathbf{X}$ is orthogonal,

$$
\hat{\alpha}_\lambda = \frac{\hat{\boldsymbol{\beta}}_\lambda^T\hat{\boldsymbol{\beta}}_{OLS}}{\hat{\boldsymbol{\beta}}_\lambda^T\hat{\boldsymbol{\beta}}_\lambda} = \frac{\sum_{j=1}^p |\hat{\beta}_{OLS,j}|(|\hat{\beta}_{OLS,j}| - \lambda)_+}{\sum_{j=1}^p (|\hat{\beta}_{OLS,j}| - \lambda)_+^2} \ . \qquad (2.35)
$$

So following expression arises:

$$\hat{\alpha}_\lambda \hat{\beta}_{\lambda,j^*} = \left(\frac{\sum_{j=1}^p |\hat{\beta}_{OLS,j}|(|\hat{\beta}_{OLS,j}| - \lambda)_+}{\sum_{j=1}^p (|\hat{\beta}_{OLS,j}| - \lambda)_+^2} \right) \times \operatorname{sign}(\hat{\beta}_{OLS,j^*}) \times (|\hat{\beta}_{OLS,j^*}| - \lambda)_+ \ . \tag{2.36}$$

Letting $d_j = (|\hat{\beta}_{OLS,j}| - \lambda)_+$ and $s_{j*} = \operatorname{sign}(\hat{\beta}_{OLS,j^*})$, the expression becomes:

$$\begin{aligned} \hat{\alpha}_\lambda \hat{\beta}_{\lambda,j^*} &= s_{j*} |\hat{\beta}_{OLS,j*}| \frac{d_{j*}^2}{\sum_{j=1}^p d_j^2} + s_{j*} d_{j*} \frac{\sum_{j \neq j*} |\hat{\beta}_{OLS,j}| d_j}{\sum_{j=1}^p d_j^2} && (2.37) \\ &= w_1 \hat{\beta}_{OLS,j*} + \hat{\beta}_{\lambda,j^*} \frac{\sum_{j \neq j*} (d_j + \lambda) d_j}{\sum_{j=1}^p d_j^2} \ , && (2.38) \end{aligned}$$

which uses $s_{j*}|\hat{\beta}_{OLS,j*}| = \hat{\beta}_{OLS,j*}$, $w_1 = d_{j*}^2 / \sum_j d_j^2$, $\hat{\beta}_{\lambda,j^*} = s_{j*} d_{j*}$, and $|\hat{\beta}_{OLS,j}| = (|\hat{\beta}_{OLS,j}| - \lambda)_+ + \lambda$ whenever $d_j > 0$. Therefore:

$$\begin{aligned} \hat{\alpha}_\lambda \hat{\beta}_{\lambda,j^*} &= w_1 \hat{\beta}_{OLS,j*} + \hat{\beta}_{\lambda,j^*} \left(\frac{\sum_{j \neq j*} d_j^2}{\sum_{j=1}^p d_j^2} + \lambda \frac{\sum_{j \neq j*} d_j}{\sum_{j=1}^p d_j^2} \right) && (2.39) \\ &= w_1 \hat{\beta}_{OLS,j*} + (1 - w_1) \hat{\beta}_{\lambda,j^*} + w_2 \hat{\beta}_{\lambda,j^*} \ , && (2.40) \end{aligned}$$

where $w_2 = \lambda \frac{\sum_{j \neq j*} d_j}{\sum_{j=1}^p d_j^2}$. Hence we have the result. □

Proof of Theorem 4. For a given λ we are allowing $\hat{\beta}_{OLS,j^*}$ to vary. For $|\hat{\beta}_{OLS,j^*}| \leq \lambda$, we have $\hat{\alpha}_\lambda \hat{\beta}_{\lambda,j^*} = 0$ so $|\hat{\alpha}_\lambda \hat{\beta}_{\lambda,j^*} - \hat{\beta}_{OLS,j^*}| = |\hat{\beta}_{OLS,j^*}| \leq \lambda$ so the theorem is satisfied for this range of values. Next we consider $\hat{\beta}_{OLS,j^*} > \lambda$. By Lemma 3,

$$\begin{aligned} \hat{\alpha}_\lambda \hat{\beta}_{\lambda,j^*} - \hat{\beta}_{OLS,j^*} &= (w_1 - 1) \hat{\beta}_{OLS,j^*} + (1 - w_1 + w_2) \hat{\beta}_{\lambda,j^*} && (2.41) \\ &= w_2 \hat{\beta}_{OLS,j^*} + (1 - w_1 + w_2)(-\lambda) && (2.42) \\ &= \frac{\lambda (\sum_{j \neq j^*} d_j) \hat{\beta}_{OLS,j^*} - \lambda (\sum_{j \neq j^*} d_j^2 + \lambda \sum_{j \neq j^*} d_j)}{(\hat{\beta}_{OLS,j^*} - \lambda)^2 + \sum_{j \neq j^*} d_j^2} && (2.43) \end{aligned}$$

Let $u = \sum_{j \neq j^*} d_j$, $v = \sum_{j \neq j^*} d_j^2$, and $x = \hat{\beta}_{OLS,j^*}$. Expression (2.43) can be written as a function, $f(x)$, that is differentiable for $x \geq \lambda$:

$$f(x) = \frac{\lambda u x - \lambda(v + \lambda u)}{(x-\lambda)^2 + v} \tag{2.44}$$

$$\frac{\partial f(x)}{\partial x} = \frac{u\lambda}{(x-\lambda)^2 + v} - \frac{2(x-\lambda)(u\lambda x - \lambda(u\lambda + v))}{((x-\lambda)^2+v)^2} \ . \tag{2.45}$$

The derivative $\frac{\partial f(x)}{\partial x}$ is equal to zero whenever $x = x^*$, where x^* is defined:

$$x^* = \frac{u\lambda + v \pm \sqrt{u^2 v + v^2}}{u} \ . \tag{2.46}$$

Note that $x^* < \lambda$ when the negative part of the square root is taken, therefore the root is restricted to $x^* = (u\lambda + v + \sqrt{u^2v + v^2})/u$. The second derivative of $f(x)$ is:

$$\frac{\partial^2 f(x)}{\partial x^2} = \frac{2\lambda[-3v(x-\lambda)(u + x - \lambda) + u(x-\lambda)^3 + v^2]}{(v + (x-\lambda)^2)^3} \ . \tag{2.47}$$

When $x = x^*$, the second derivative equals:

$$-\frac{u^4\lambda(u^2+v)(\sqrt{u^2v+v^2}+v)}{2v^2(\sqrt{u^2v+v^2}+u^2+v)^3} < 0 \tag{2.48}$$

meaning that x^* gives the maximum of $f(x)$ when $x > \lambda$. Therefore, between $x = \lambda$ and $x = x^*$, $f(x)$ is an increasing function bounded below by $-\lambda$ and bounded above by

$$\frac{\lambda}{2}\Big(\sqrt{\frac{u^2}{v} + 1} - 1\Big) \ . \tag{2.49}$$

For $x > x^*$, $f(x)$ is decreasing so we consider $\lim_{x\to\infty} f(x)$, which is easily shown to equal $0 > -\lambda$. Therefore,

$$|f(x)| \leq \lambda \times \max\left(1,\ \frac{1}{2}\Big(\sqrt{\frac{u^2}{v}+1}-1\Big)\right)\ , \tag{2.50}$$

and $\lim_{x\to\infty}|f(x)| = 0$. For $\hat{\beta}_{OLS,j^*} < -\lambda$, note that $\hat{\alpha}_\lambda\hat{\beta}_{\lambda,j^*} - \hat{\beta}_{OLS,j^*} = -(\hat{\alpha}_\lambda|\hat{\beta}_{\lambda,j^*}| - |\hat{\beta}_{OLS,j^*}|)$ and so analogous arguments to the case of $\hat{\beta}_{OLS,j^*}$ also apply. This completes the proof.□

Proof of Theorem 5. When the sign vector of the true model, $\mathbf{s}_1$, is known there exists a closed form expression for the Lasso solutions: $\hat{\boldsymbol{\beta}}_\lambda = (\hat{\boldsymbol{\beta}}_{OLS} - n\lambda(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1)$ where $\mathbf{X}_1$ is the covariate matrix including all variables in the support of the true model. The following is also known:

$$\hat{\alpha}_\lambda\hat{\boldsymbol{\beta}}_\lambda = \Big(\frac{\hat{\boldsymbol{\beta}}_\lambda^T\mathbf{X}_1^T\mathbf{y}}{\hat{\boldsymbol{\beta}}_\lambda^T\mathbf{X}_1^T\mathbf{X}_1\hat{\boldsymbol{\beta}}_\lambda}\Big)\hat{\boldsymbol{\beta}}_\lambda. \tag{2.51}$$

Now, we can find definitions of these components in terms of $\hat{\boldsymbol{\beta}}_{OLS}$, supposing $\hat{\beta}_{\lambda,j^*} \neq 0$:

$$\begin{aligned}
\hat{\mathbf{y}}_\lambda^T\mathbf{y} &= \hat{\boldsymbol{\beta}}_\lambda\mathbf{X}_1^T\mathbf{y} && (2.52)\\
&= \hat{\boldsymbol{\beta}}_{OLS,1}^T\mathbf{X}_1^T\mathbf{X}_1\hat{\boldsymbol{\beta}}_{OLS,1} - n\lambda\mathbf{s}_1^T\hat{\boldsymbol{\beta}}_{OLS,1} && (2.53)\\
&= \big(\sum_{i=1}^n x_{ij^*}^2\big)\hat{\beta}_{OLS,j^*}^2 + \big(2\sum_{i=1}^n x_{ij^*}\big(\sum_{j\neq j^*} x_{ij}\hat{\beta}_{OLS,j}\big)\big) - n\lambda s_{j^*}\big)\hat{\beta}_{OLS,j^*} && (2.54)\\
&\quad - n\lambda\sum_{j\neq j^*}|\hat{\beta}_{OLS,j}| + \sum_{i=1}^n\big(\sum_{j\neq j^*}\hat{\beta}_{OLS,j}x_{ij}\big)^2 && (2.55)
\end{aligned}$$

$$\hat{\mathbf{y}}_\lambda^T\hat{\mathbf{y}}_\lambda = \hat{\boldsymbol{\beta}}_{OLS,1}^T\mathbf{X}_1^T\mathbf{X}_1\hat{\boldsymbol{\beta}}_{OLS,1} - 2n\lambda\mathbf{s}_1^T\hat{\boldsymbol{\beta}}_{OLS,1} + n^2\lambda^2\mathbf{s}_1^T(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1 \tag{2.56}$$

$$= \big(\sum_{i=1}^n x_{ij^*}^2\big)\hat{\beta}_{OLS,j^*}^2 + \big(2\sum_{i=1}^n x_{ij^*}\big(\sum_{j\neq j^*} x_{ij}\hat{\beta}_{OLS,j}\big) - 2n\lambda s_{j^*}\big)\hat{\beta}_{OLS,j^*} \tag{2.57}$$

$$- 2n\lambda\sum_{j\neq j^*}|\hat{\beta}_{OLS,j}| + \sum_{i=1}^n\big(\sum_{j\neq j^*}\hat{\beta}_{OLS,j}x_{ij}\big)^2 + n^2\lambda^2\mathbf{s}_1^T(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1. \tag{2.58}$$

Here, x_{ij} is the ith element of the jth column of $\mathbf{X}_1$ Note that:

$$\hat{\mathbf{y}}_\lambda^T\mathbf{y} - \hat{\mathbf{y}}_\lambda^T\hat{\mathbf{y}}_\lambda = n\lambda s_{j^*}\hat{\beta}_{OLS,j^*} + n\lambda\sum_{j\neq j^*}|\hat{\beta}_{OLS,j}| - n^2\lambda^2\mathbf{s}_1^T(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1. \tag{2.59}$$

Hence:

$$\hat{\alpha}_\lambda\hat{\beta}_{\lambda,j^*} - \hat{\beta}_{OLS,j^*} = \hat{\alpha}_\lambda\big(\hat{\beta}_{OLS,j^*} - n\lambda[(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1]_{j^*}\big) - \hat{\beta}_{OLS,j^*} \tag{2.60}$$

$$= (\hat{\alpha}_\lambda - 1)\hat{\beta}_{OLS,j^*} - \hat{\alpha}_\lambda n\lambda[(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1]_{j^*} \tag{2.61}$$

$$= \frac{\hat{\mathbf{y}}_\lambda^T\mathbf{y}(\hat{\beta}_{OLS,j^*} - n\lambda[(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1]_{j^*})}{\hat{\mathbf{y}}_\lambda^T\hat{\mathbf{y}}_\lambda} - \frac{\hat{\mathbf{y}}_\lambda^T\hat{\mathbf{y}}_\lambda\hat{\beta}_{OLS,j^*}}{\hat{\mathbf{y}}_\lambda^T\hat{\mathbf{y}}_\lambda} \tag{2.62}$$

$$= \frac{(\hat{\mathbf{y}}_\lambda^T\mathbf{y} - \mathbf{y}_\lambda^T\hat{\mathbf{y}}_\lambda)\hat{\beta}_{OLS,j^*} - n\lambda\hat{\mathbf{y}}_\lambda^T\mathbf{y}[(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1]_{j^*}}{\hat{\mathbf{y}}_\lambda^T\hat{\mathbf{y}}_\lambda} \tag{2.63}$$

Now, using the expression for $\hat{\mathbf{y}}_\lambda^T\mathbf{y} - \hat{\mathbf{y}}_\lambda^T\hat{\mathbf{y}}_\lambda$ derived above, it can be shown that as $|\hat{\beta}_{OLS,j^*}| \to \infty$:

$$\hat{\alpha}_\lambda\hat{\beta}_{\lambda,j^*} - \hat{\beta}_{OLS,j^*} \to \frac{n\lambda s_{j^*}\hat{\beta}_{OLS,j^*}^2 - n\lambda[(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1]_{j^*}\big(\sum_{i=1}^n x_{ij^*}^2\big)\hat{\beta}_{OLS,j^*}^2}{\big(\sum_{i=1}^n x_{ij^*}^2\big)\hat{\beta}_{OLS,j^*}^2} \tag{2.64}$$

$$\to \frac{n\lambda s_{j^*} - n\lambda[(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1]_{j^*}\big(\sum_{i=1}^n x_{ij^*}^2\big)}{\big(\sum_{i=1}^n x_{ij^*}^2\big)} \tag{2.65}$$

$$= n\lambda\Big(\frac{s_{j^*}}{\big(\sum_{i=1}^n x_{ij^*}^2\big)} - [(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1]_{j^*}\Big). \tag{2.66}$$

If we assume that the columns of $\mathbf{X}$ have been centered and scaled, we can say:

$$\hat{\alpha}_\lambda \hat{\beta}_{\lambda,j^*} - \hat{\beta}_{OLS,j^*} \to n\lambda\Big(s_{j^*} - [(\mathbf{X}_1^T\mathbf{X}_1)^{-1}]_{j^*}\mathbf{s}_1\Big) = G \ . \tag{2.67}$$

By the definition of the Lasso estimate:

$$\hat{\beta}_{\lambda,j^*} - \hat{\beta}_{OLS,j^*} = -n\lambda[(\mathbf{X}_1^T\mathbf{X}_1)^{-1}]_{j^*}\mathbf{s}_1 \ , \tag{2.68}$$

Because accurate sign recovery has been assumed, when $s_{j^*} > 0$, $-n\lambda[(\mathbf{X}_1^T\mathbf{X}_1)^{-1}]_{j^*}\mathbf{s}_1 < 0$ and when $s_{j^*} < 0$, $-n\lambda[(\mathbf{X}_1^T\mathbf{X}_1)^{-1}]_{j^*}\mathbf{s}_1 > 0$. Now for proof by contradiction, assume $|G| > |\hat{\beta}_{\lambda,j^*} - \hat{\beta}_{OLS,j^*}|$. The absolute value inequality can be broken into two cases, first where $s_{j^*} > 0$ and then where $s_{j^*} < 0$. Begin with $s_{j^*} > 0$:

$$G < -n\lambda[(\mathbf{X}_1^T\mathbf{X}_1)^{-1}]_{j^*}\mathbf{s}_1 \tag{2.69}$$

$$n\lambda\Big(s_{j^*} - [(\mathbf{X}_1^T\mathbf{X}_1)^{-1}]_{j^*}\mathbf{s}_1\Big) < -n\lambda[(\mathbf{X}_1^T\mathbf{X}_1)^{-1}]_{j^*}\mathbf{s}_1 \tag{2.70}$$

$$n\lambda s_{j^*} < 0 \ . \tag{2.71}$$

This is a contradiction. Now consider $s_{j^*} < 0$:

$$G > -n\lambda[(\mathbf{X}_1^T\mathbf{X}_1)^{-1}]_{j^*}\mathbf{s}_1 \tag{2.72}$$

$$n\lambda\Big(s_{j^*} - [(\mathbf{X}_1^T\mathbf{X}_1)^{-1}]_{j^*}\mathbf{s}_1\Big) > -n\lambda[(\mathbf{X}_1^T\mathbf{X}_1)^{-1}]_{j^*}\mathbf{s}_1 \tag{2.73}$$

$$n\lambda s_{j^*} > 0 \ , \tag{2.74}$$

giving us another contradiction. Note that $G = |\hat{\beta}_{\lambda,j^*} - \hat{\beta}_{OLS,j^*}|$ if and only if $n\lambda s_{j^*} = 0$, which is another contradiction. Therefore, $|G| < |\hat{\beta}_{\lambda,j^*} - \hat{\beta}_{OLS,j^*}|$. Thus the absolute value of the difference between the α-Modified Lasso estimate and the OLS estimate approaches a constant that is always less than the absolute value of the difference between the unmodified Lasso estimate and the OLS estimate as the OLS estimate approaches infinity.

□

CHAPTER

3

EXTENSIONS OF THE α-MODIFICATION

3.1 Motivation

In Chapter 2, the α-Modification was introduced as a bias reduction technique for penalized estimators of the linear model inspired by an approach to tuning parameter selection that emphasized correlation rather than squared error loss. The α-Modification is efficient and generalizable to all penalties, but its extension to GLMs is non-trivial. This chapter addresses some of the challenges of extending the modification to Model (1.2) and provides an algorithm to find $\hat{\alpha}_\lambda$. Furthermore, as discussed in Chapter 2, even for the linear model the α-Modification is not guaranteed to reduce the bias of penalized estimates for all λ. A penalized α-Modification, the α_ν-Modification, is proposed as an answer to the bias concerns of the preceding chapter.

This chapter discusses both the linear model (1.1) and GLM (1.2). Penalized estimation may be necessary either of these models for a variety of reasons: to reduce high-dimensional data, separate out features of a data set that are important to predict some response, or to arrive at estimates when unpenalized approaches are either unavailable or inadvisable. For example, in logistic regression, unregularized maximum likelihood estimates (ML) can suffer from the somewhat counter-intuitive problem of complete separation (Albert and Anderson 1984). If the estimated decision boundary of responses correctly classifies all observations, then increasing the scale of those predictions only pushes them further towards the correct classification of 0 or 1. If coefficient estimates are found via gradient descent, they can grow in magnitude without bounds. When this happens, an imposed constraint like the Ridge or Lasso penalty, can stabilize estimates and improve interpretability, whether or not variable selection is of interest.

As discussed in Chapter 1, penalties that can be used for the linear model are generally available to GLMs as well. The solution paths for the Lasso and Elastic Net for GLMs have been derived (Park and Hastie 2007; Tay et al. 2021) and the Ridge penalty has long been used to stabilize GLM coefficient estimates (Segerstedt 1992). It is reasonable to conclude that many sparsity-inducing penalties will suffer from the same shrinkage problem in GLMs as they do for the linear model. That is, in order for some coefficients to be necessarily shrunk to exactly zero, active coefficient estimates shrink as well, making tuning parameter selection for the purpose of variable selection difficult. The SCAD, MCP, and SELO penalties have all been used to combat this issue for GLMs (Fan and Li 2001; Zhang 2010; Li et al. 2012).

The Relaxed Lasso and, more generally the Relaxed approach to any penalty, can be applied to GLMs. Although Meinshausen (2007) do not describe the application in much detail, the concept of relaxing penalized estimates is not altered by the use of a GLM as opposed to a linear model. The convergence results for the Relaxed Lasso discussed in Chapter 1 have not yet been extended to include GLMs. In contrast, recall that the Adaptive Lasso has been proven to be variable selection consistent for GLMs under some mild regularity conditions and assuming $\lambda_n/\sqrt{n} \to 0$ and $\lambda_n n^{(\gamma-1)/2} \to \infty$ as $n \to \infty$. The primary drawback of both these approaches is their computational expense and, for the Adaptive Lasso, the fact that finding consistent estimates for GLMs to use as weights can sometimes be a challenge.

Regardless of the penalty used, a reliable method of tuning parameter selection is still necessary. Information criteria like AIC and BIC are prone to overselection for GLMs (Chen and Chen 2012), as is K-fold CV (Feng and Yu 2013; Roberts and Nowak 2014), especially for the Lasso penalty. A Debiased Lasso exists for GLMs (van de Geer et al. 2014), though recent work suggests that the bias for these estimates remains high (Xia et al. 2020). Furthermore, the debiased framework has not, to our knowledge, been extended to non-Lasso penalties. The other post-selection inference techniques described in Chapter 1 are not useful for GLMs.

The α-Modification can be a computationally efficient bolster to penalized estimates. It has the advantage of easy applicability to any type of penalized estimator. The extension of the α-Modification to GLMs is non-trivial for three reasons. First, most GLMs do not have closed form solutions for even unpenalized maximum likelihood estimates, let alone penalized estimates. This means that a closed form expression for $\hat{\alpha}_\lambda$ will not be possible for many GLMs, including logistic and Poisson regression problems, and an iterative algorithm is necessary to reach estimates of α. Second, for the linear model the responses can be centered such that the intercept term is simply zero; this is not the case for most GLMs with non-identity link functions. As a result, the α-Modified estimation process must optimize over both the intercept and α in order to find the best estimates. The lack of a closed form solution and the need to account for an intercept term are not unique challenges to the α-Modification. Both are necessary hurdles for approaches such as the Relaxed and Debiased Lasso as well. Third, and more specifically, consider the problem of complete separation for logistic regression. The introduction of a scalar α term in front of the coefficient estimates will exacerbate this problem when it arises. That is, an algorithm for finding $\hat{\alpha}_\lambda$ may struggle to converge as $\hat{\alpha}_\lambda$ increases to positive or negative infinity or negative.

This final challenge inspired, in part, the second major contribution of this chapter: a penalized version of the α-Modification, the α_ν-Modification. Simply put, imposing a penalty upon $\hat{\alpha}_\lambda$ will keep it from exploding when complete separation issues arise in logistic regression. Further investigation of the penalized α-Modification proved that the approach has merit apart from logistic regression as well. For linear regression using the Lasso penalty, it is proven in this chapter that there are conditions under which there always exists a tuning parameter ν such that the α_ν-Modification reduces the distance between the penalized estimate and the OLS estimate, thereby reducing the bias when accurate sign recovery has occurred. As a result, the α_ν-Modification is an answer to the bias problems that the unpenalized α-Modification sometimes has, as discussed in Chapter 2.

This chapter is organized as follows. Section 3.2 describes the process of using the α-Modification for GLMs and gives an algorithm to find estimates. Theoretical results of the modification for any penalty under GLMs is also discussed. In Section 3.3, the α_ν-Modification is presented and several theoretical results are given to show that the penalized modification is a useful bias reduction tool for the Lasso on linear models. Section 3.4 discusses the results of simulation studies on the Lasso penalty for linear, logistic, and Poisson regression using both the α- and α_ν-Modifications to enhance estimates. Section 3.5 explores the new methods of tuning parameter selection in an application involving the classification of Turkish raisins and Section 3.6 explores areas of future research. The proofs for all theoretical results are provided in Section 3.A.

3.2 α-Modification for Generalized Linear Models

Chapter 2 introduced a reformulation of the linear model to separate the coefficient vector, $\boldsymbol{\beta}^*$ into two components: $\boldsymbol{\xi}^*$, the direction of the estimates with respect to one another, and α, a scaling constant that represents the magnitude of the direction. The α-Modification then uses $\hat{\boldsymbol{\beta}}_\lambda$ to estimate $\boldsymbol{\xi}^*$ and then optimizes the likelihood to find an estimate of α. Consider this reformulation for Model (1.2), allowing $\boldsymbol{\beta}^* = \alpha^*\boldsymbol{\xi}^*$. The expected value of the response is:

$$E[\mathbf{y}_{n\times 1}|\mathbf{X}_{n\times p}] = g^{-1}(\beta_0^*\mathbf{1}_n + \alpha^*\mathbf{X}_{n\times p}\boldsymbol{\xi}^*_{p\times 1}) \ . \tag{3.1}$$

Unlike for the linear model, a closed form expression for $\hat{\alpha}_\lambda$ is no longer achievable. An additional complication to the generalization is the intercept term. For the linear model, it is reasonable to assume that the response, $\mathbf{y}$ is centered at 0, making it unnecessary to estimate β_0^*. For Model (1.2), estimating the intercept term is crucial to making accurate predictions. Accounting for the intercept term is necessary to ensure the α-Modification is useful for GLMs, but complicates the α estimation process.

Let $-\ell(\beta_0, \alpha, \boldsymbol{\xi} \,|\, \mathbf{y}, \mathbf{X})$ be the negative log-likelihood from Model (1.2) in terms of α and $\boldsymbol{\xi}$. A more general formulation of the adjustment proposed in Chapter 2 is summarized by the objective function:

$$h(\beta_0, \alpha, \boldsymbol{\xi}) = -\ell(\beta_0, \alpha, \boldsymbol{\xi}|\mathbf{y}, \mathbf{X}) + P_\lambda(\boldsymbol{\xi}) \ . \tag{3.2}$$

Because this expression is biconvex, attempts to minimize it must be carried out in two stages, similar to the linear model. First, fix $\alpha = 1$ and find penalized estimates $\hat{\beta}_{0,\lambda}$

and $\hat{\boldsymbol{\beta}}_\lambda$. Note that the minimizers vary according to λ, so it is necessary to perform the optimization across all λ of interest. This step is well studied, as it is simply penalized estimation for generalized linear models (Segerstedt 1992; Park and Hastie 2007; Friedman et al. 2010; Li et al. 2012). Next, fix $\boldsymbol{\xi} = \hat{\boldsymbol{\beta}}_\lambda$ and minimize the likelihood with respect to α and β_0. When the derivative of $-\ell(\beta_0, \alpha | \boldsymbol{\xi} = \hat{\boldsymbol{\beta}}_\lambda, \mathbf{y}, \mathbf{X})$ is set to zero, the following expression emerges:

$$\sum_i y_i \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda = \sum_i b'(\beta_0 + \alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda) \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda \ . \tag{3.3}$$

Under a canonical link function, the matrix form expression is:

$$\mathbf{y}^T \mathbf{X} \hat{\boldsymbol{\beta}}_\lambda = g^{-1}(\beta_0 \mathbf{1}_n + \alpha \mathbf{X} \hat{\boldsymbol{\beta}}_\lambda)^T \mathbf{X} \hat{\boldsymbol{\beta}}_\lambda \ . \tag{3.4}$$

It is clear that, when $\beta_0 = 0$ and $g^{-1}(z) = z$, $\hat{\alpha}_\lambda = \frac{\mathbf{y}^T \mathbf{X} \hat{\boldsymbol{\beta}}_\lambda}{\hat{\boldsymbol{\beta}}_\lambda^T \mathbf{X}^T \mathbf{X} \hat{\boldsymbol{\beta}}_\lambda}$. This is the expression used for $\hat{\alpha}_\lambda$ for the linear model in Chapter 2. Under the generalized formulation, the optimization problem requires an iterative algorithm to solve. Algorithm 1 describes an implementation of Newton's Method that can be used to find $\hat{\beta}_{0,\lambda}^{\alpha}$, the estimated intercept when $\hat{\boldsymbol{\xi}} = \hat{\boldsymbol{\beta}}_\lambda$, and $\hat{\alpha}_\lambda$, the estimated scaling factor. Details of the objective function h and its gradient can be found in Appendix B.1.

Chapter 2 proved that $\hat{\alpha}_\lambda \geq 1$ for all λ under convex penalties minimized at $\mathbf{0}$ for the linear model. This result is important in that it shows that the modification pushes estimates away from zero, rather than shrinking them further. When generalized to GLMs, we find a similar result under reasonable conditions on the penalty and log-likelihood.

Theorem 6. *If $\hat{\boldsymbol{\beta}}_\lambda$ is the unique minimizer of $-\ell(\beta_0, \boldsymbol{\beta}) + P_\lambda(\boldsymbol{\beta})$ where $-\ell(\boldsymbol{\beta})$ is convex, and $P_\lambda(\boldsymbol{\beta})$ is a norm function, then $\hat{\alpha}_\lambda \geq 1$.*

Although this result shows that the α-Modification for norm penalties never shrinks

Algorithm 1 α-Modification for GLM

Given: Let $\hat{\boldsymbol{\beta}}_\lambda$ be the penalized coefficient estimates at a fixed λ. Now $h(\mathbf{z}) = h(\begin{bmatrix}\beta_0 & \alpha\end{bmatrix}^T) = h(\beta_0, \alpha|\hat{\boldsymbol{\beta}}_\lambda)$. Let $\mathbf{z}_0 = \begin{bmatrix}\hat{\beta}_{0,\lambda} & 1\end{bmatrix}^T$ where $\hat{\beta}_{0,\lambda}$ is the estimated slope for $\hat{\boldsymbol{\beta}}_\lambda$. Set $k = 1$ and tolerance $\epsilon > 0$.
while $\psi^2 > \epsilon$ and $\delta > 0$ **do**
  Compute $\Delta\mathbf{z}_k \leftarrow \nabla^2 h(\mathbf{z}_{k-1})^{-1}\nabla h(\mathbf{z}_{k-1})$
  Find $\psi^2 \leftarrow \nabla h(\mathbf{z}_{k-1})^T \nabla^2 h(\mathbf{z}_{k-1})^{-1}\nabla h(\mathbf{z}_{k-1})$
  $\mathbf{z}_k \leftarrow \boldsymbol{z}_{k-1} - t\Delta\mathbf{z}_k$
  $\delta \leftarrow h(\mathbf{z}_k) - h(\mathbf{z}_{k-1})$ ▷ To ensure each step improves penalized likelihood
end while
Then $\begin{bmatrix}\hat{\beta}^\alpha_{0,\lambda} & \hat{\alpha}_\lambda\end{bmatrix}^T \leftarrow \mathbf{z}_k^T$.

estimates towards zero, Theorem 6 does not show that the α-Modification always reduces the bias of the original penalized estimates. For instance, when the λ selected gives rise to a sparse but inaccurate submodel, coefficients that should be zero will be amplified using the α-Modification. That is, the α-Modification is only as good as the tuning parameter and the penalized estimator it modifies.

An additional concern with the α-Modification with respect to GLMs is that of complete separation in logistic regression, discussed in Section 3.1. Complete separation and its weaker form, quasi-complete separation, occur when $\mathbf{X}$ predicts all or a subset of the response perfectly and often leads estimates to diverge (Albert and Anderson 1984; Sauter and Held 2016). The α-Modification will also struggle to converge under these conditions and increasingly large values of $\hat{\alpha}_\lambda$ may have no effect on predictions.

3.3 α_ν-Modification

As an answer to potential convergence issues and to the fact that the α-Modification cannot reduce bias for every possible λ, this section introduces the α_ν-Modification. The α_ν-Modification imposes a penalty on α, shrinking estimates of the scaling factor to stabilize the procedure and protect the modification against exacerbating the bias of the penalized estimates.

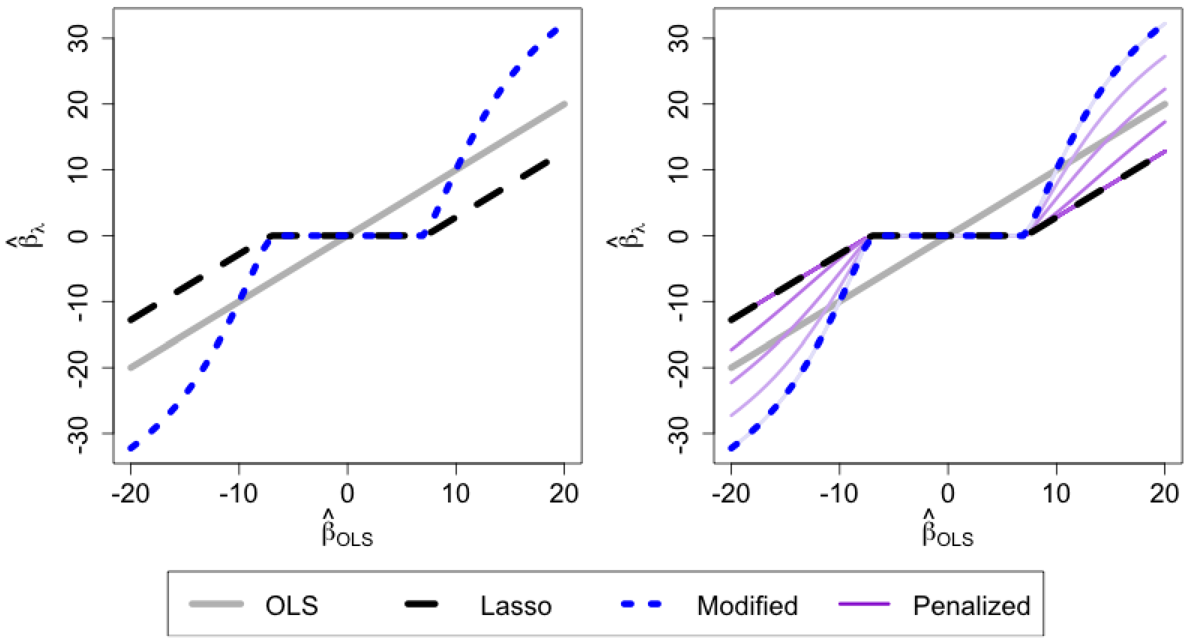

Figure 3.1: Lasso ($\hat{\beta}_{\lambda',1}$), α-Modified Lasso ($\hat{\alpha}_{\lambda'}\hat{\beta}_{\lambda',1}$), and α_ν-Modified Lasso ($\hat{\alpha}_{\lambda',\nu}\hat{\beta}_{\lambda',1}$) solutions for the linear model as $\hat{\beta}_{OLS,1}$ varies when $\lambda' = 7.2$. This tuning parameter value results in perfect variable selection. $\hat{\beta}_{OLS,j} = 10$ for $j = \{2, ..., 20\}$. Larger ν result in solutions closer to $\hat{\beta}_{\lambda',1}$ and smaller ν give solutions closer to $\hat{\alpha}'_{\lambda}\hat{\beta}_{\lambda',1}$.

Consider the Lasso for a linear model, depicted in Figure 3.1. The leftmost plot in the figure shows Lasso and α-Modified Lasso solutions for a single coefficient at a single value of λ, $\lambda' = 7.2$, as the OLS estimate varies. In this example, $p = 20$ and $\hat{\boldsymbol{\beta}}^T_{OLS} = \{\hat{\beta}_{OLS,1}, 10, ..., 10\}$. All twenty coefficients are non-zero in $\hat{\boldsymbol{\beta}}_{\lambda'}$. The $\mathbf{X}$ matrix is orthogonal, so Theorem 4 from Chapter 2 applies and the bound on the absolute difference between $\hat{\alpha}_{\lambda'}\hat{\beta}_{\lambda',1}$ and $\hat{\beta}_{OLS,1}$ is 12.49. That is, $\hat{\alpha}_{\lambda'}\hat{\beta}_{\lambda',1}$ may be larger in magnitude than $\hat{\beta}_{OLS,1}$ by as much as 12.49, whereas the unmodified Lasso estimate, $\hat{\beta}_{\lambda',1}$ will only ever be $\lambda' = 7.2$ smaller than the OLS estimate. Figure 3.1 shows that the distance between $\hat{\beta}_{\lambda',1}$

and $\hat{\beta}_{OLS,1}$ is smaller than the distance between $\hat{\alpha}_{\lambda'}\hat{\beta}_{\lambda',1}$ and $\hat{\beta}_{OLS,1}$ when $\hat{\beta}_{OLS,1} \approx 15$.

Although Theorem 4 proved that as $|\hat{\beta}_{OLS,1}|$ approaches infinity, $|\hat{\alpha}_{\lambda}\hat{\beta}_{\lambda,1} - \hat{\beta}_{OLS,1}|$ approaches 0, it is clear that the α-Modification can fail to reduce bias for some situations. Under the conditions of Theorem 4, this can happen whenever:

$$\frac{1}{2}\left(\sqrt{\frac{u^2}{v}+1}-1\right) > 1 \ , \tag{3.5}$$

where $u = \sum_{j \neq j^*} d_j$ and $v = \sum_{j \neq j^*} d_j^2$. The difference between the OLS and α-Modified estimate becomes large as $\frac{u^2}{v}$ becomes large, which happens when many d_j are nonzero, but close to zero. In Chapter 2, this possibility is for the most part ignored. It is reasonable to conclude, after all, that α-Modified K-fold Cross Validation will never choose λ where this has occurred. A more robust response to the possibility, however, is warranted.

Now consider the rightmost plot in Figure 3.1. Pictured here, in addition to the Lasso and α-Modified Lasso estimates are the new set of α_ν-Modified Lasso estimates. The α_ν-Modified estimates range from $\hat{\beta}_{\lambda',1}$ to $\hat{\alpha}_{\lambda'}\hat{\beta}_{\lambda',1}$, representing a compromise between unmodified and α-Modified Lasso estimates. This compromise can lead estimates to be closer to $\hat{\beta}_{OLS,1}$ than either the unmodified or the fully modified solutions.

The penalty on α used for the α_ν-Modification is $\nu|\alpha - 1|$, resulting in the following expression for $\hat{\alpha}_{\lambda,\nu}$:

$$\hat{\alpha}_{\lambda,\nu} = \arg\min_{\alpha}\big[-\ell(\beta_0, \alpha, \boldsymbol{\xi}|\mathbf{y}, \mathbf{X}) + P_\lambda(\boldsymbol{\xi}) + \nu|\alpha - 1|\big] \ . \tag{3.6}$$

Note that, in (3.6), $P_\lambda(\boldsymbol{\xi})$ does not affect the value of $\hat{\alpha}_{\lambda,\nu}$ and can be removed from the expression. Across all possible values of ν, $\hat{\alpha}_{\lambda,\nu}$ ranges from 1 (when ν is large) to $\hat{\alpha}_\lambda$

(when $\nu = 0$). Algorithm 1 finds $\hat{\alpha}_{\lambda,\nu}$ for any model of the form described in (1.3) when:

$$h(\beta_0, \alpha) = -\frac{\sum_i y_i(\beta_0 + \alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)} + \frac{\sum_i b(\beta_0 + \alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)} + \nu|\alpha - 1| \ . \tag{3.7}$$

Appendix B.1 gives the gradient for the objective function to assist in implementation.

The additional tuning parameter, $\nu \geq 0$, determines the scope of the multiplicative correction, giving options as to how large the estimated scaling factor is allowed to be. The ν tuning parameter space should be explored as completely as possible before a final value of ν is chosen. In this chapter, K-fold CV is used to select the optimal combination of λ—from the penalized estimator—and ν—from the Penalized α-Modification. When the set of ν values is chosen such that 1 is the smallest $\hat{\alpha}_{\lambda,\nu}$ and $\hat{\alpha}_\lambda = \hat{\alpha}_{\lambda,0}$ is the largest, both the fully α-Modified penalized estimate and the un-modified penalized estimate are considered. Intuitively, the α_ν-Modification allows one to choose how much of an effect—if any—the modification will have on estimates based on information from K-fold CV. The minimum APE can be used to choose λ and ν, or a 1SE Rule may be employed.

The unpenalized α-Modification was framed in Chapter 2 as a bridge to the Relaxed penalized estimation method (Meinshausen 2007); the addition of a penalty on α strengthens this connection because the second penalty guides the impact of the bias adjustment. The direction of penalized estimates remains fixed, so α_ν-adjusted solutions will still not be identical to Relaxed solutions. However, the inclusion of a penalty on α allows for informed decisions about how much of an impact the multiplicative adjustment has.

3.3.1 Specifics for the Linear Model

For the linear model, estimates for α_ν are found through a process that is nearly identical to the traditional Lasso for a single variable. Because of this, closed form expressions are easily derived. Under the linear model, g^{-1} is the identity and $\mathbf{y}$ is assumed to be centered, so $\beta_0 = 0$. Because the derivative of $|\alpha - 1|$ is undefined at one, consider the subdifferential definition:

$$\frac{\partial h(\alpha)}{\partial \alpha} = -\mathbf{y}^T \mathbf{X} \hat{\boldsymbol{\beta}}_\lambda + \alpha \hat{\boldsymbol{\beta}}_\lambda^T \mathbf{X}^T \mathbf{X} \hat{\boldsymbol{\beta}}_\lambda + \nu \mathbf{s}_1 \tag{3.8}$$

where $\mathbf{s}_1$ is defined below, with the understanding that $\alpha \geq 1$:

$$\mathbf{s}_1 = \begin{cases} 1 & \alpha > 1 \\ [-1, 1] & \alpha = 1 \end{cases}. \tag{3.9}$$

Assume that KKT conditions are met and the subdifferential must contain zero. Therefore:

$$\hat{\alpha}_{\lambda,\nu} = \frac{\mathbf{y}^T \mathbf{X} \hat{\boldsymbol{\beta}}_\lambda - \nu \mathbf{s}_1}{\hat{\boldsymbol{\beta}}_\lambda^T \mathbf{X}^T \mathbf{X} \hat{\boldsymbol{\beta}}_\lambda}. \tag{3.10}$$

Allow $\hat{\alpha}_{\lambda,0}$ to be the unpenalized estimate of α, $\hat{\alpha}_{\lambda,0} = \hat{\alpha}_\lambda = \frac{\mathbf{y}^T \hat{\mathbf{y}}_\lambda}{\hat{\mathbf{y}}_\lambda^T \hat{\mathbf{y}}_\lambda}$. Then:

$$\hat{\alpha}_{\lambda,\nu} = \begin{cases} \hat{\alpha}_{\lambda,0} - \frac{\nu}{\hat{\boldsymbol{\beta}}_\lambda \mathbf{X}^T \mathbf{X} \hat{\boldsymbol{\beta}}_\lambda} & \hat{\alpha}_{\lambda,0} - \frac{\nu}{\hat{\boldsymbol{\beta}}_\lambda^T \mathbf{X}^T \mathbf{X} \hat{\boldsymbol{\beta}}_\lambda} > 1 \\ 1 & \hat{\alpha}_{\lambda,0} - \frac{\nu}{\hat{\boldsymbol{\beta}}_\lambda^T \mathbf{X}^T \mathbf{X} \hat{\boldsymbol{\beta}}_\lambda} \leq 1 \end{cases}. \tag{3.11}$$

The expression given in (3.11) offers a closed form solution for the α_ν-Modification for any penalty on estimates under the linear model. To name a few, the Lasso, Elastic Net (Zou and Hastie 2005), SCAD (Fan and Li 2001), MCP (Zhang 2010), and SELO (Dicker et al. 2013) penalties can all utilize (3.11) to efficiently find estimates for α_ν. This calculation is trivial because it simply involves subtracting a fixed value from $\hat{\alpha}_{\lambda,0}$, although many ν must be explored to find an optimal value.

3.3.2 α_ν-Modification for the Linear Lasso

In order to more thoroughly examine theoretical properties of the modification, a single penalty must be specified. We choose the Lasso because of its popularity and simplicity and because the finite sample size theoretical results of the unpenalized α-Modification provided in Chapter 2 focus on the Lasso penalty. Again, although the fitting algorithm proposed in Section 3.2 can be applied to any GLM, the following finite-sample theoretical results assume a linear model.

Theorem 7. *Suppose $\boldsymbol{X}$ is orthogonal and fix λ such that the Lasso estimate of the linear model $\hat{\beta}_{\lambda,j^*} \neq 0$. There exists some ν_{j^*} such that $\alpha_\nu \hat{\beta}_{\lambda,j^*}$ is a convex combination of $\hat{\beta}_{OLS,j^*}$ and $\hat{\beta}_{\lambda,j^*}$.*

Corollary 7.1. *There exists some ν_{j^*} such that: $|\hat{\beta}_{OLS,j^*} - \hat{\alpha}_{\lambda,\nu}\hat{\beta}_{\lambda,j^*}| \leq |\hat{\beta}_{OLS,j^*} - \hat{\beta}_{\lambda,j^*}|$.*

Theorem 7 and Corollary 7.1 show that for every fixed λ and j^*, it is possible to find a value for ν such that the α_ν-Modified Lasso estimate for coefficient j^* is closer to the OLS estimate than the un-modified Lasso estimate was. This is an encouraging result, but insufficient in practice. Consider a fixed λ. Corollary 7.1 shows that there exists some ν such that $\hat{\alpha}_{\lambda,\nu}\hat{\beta}_{\lambda,j^*}$ is closer to $\hat{\beta}_{OLS,j^*}$ than $\hat{\beta}_{\lambda,j^*}$, but recall that this adjustment is applied to the entire coefficient vector, $\hat{\boldsymbol{\beta}}_\lambda$. The more probing question to ask, then, is whether it is possible to reduce the distance between $\hat{\boldsymbol{\beta}}^1_{OLS}$ and $\hat{\boldsymbol{\beta}}^1_\lambda$ where $\hat{\boldsymbol{\beta}}^1_{OLS}$ is the vector of OLS estimates found using the covariates indexed by $\mathcal{M}_\lambda$ and $\hat{\boldsymbol{\beta}}^1_\lambda$ are the non-zero elements of $\hat{\boldsymbol{\beta}}_\lambda$.

Theorem 8. *Under the linear model, for an orthogonal $\boldsymbol{X}$ matrix and any fixed $\lambda > 0$, there exists some $\nu > 0$ such that:*

$$||\hat{\boldsymbol{\beta}}^1_{OLS} - \hat{\boldsymbol{\beta}}^1_\lambda||_2^2 > ||\hat{\boldsymbol{\beta}}^1_{OLS} - \hat{\alpha}_{\lambda,\nu}\hat{\boldsymbol{\beta}}^1_\lambda||_2^2 \tag{3.12}$$

where $\hat{\boldsymbol{\beta}}^1_{OLS}$ *and* $\hat{\boldsymbol{\beta}}^1_{\lambda}$ *are the components of* $\hat{\boldsymbol{\beta}}_{OLS}$ *and* $\hat{\boldsymbol{\beta}}_{\lambda}$ *respectively that are non-zero in* $\hat{\boldsymbol{\beta}}_{\lambda}$*, the Lasso estimate.*

The vector $\hat{\boldsymbol{\beta}}^1_{OLS}$ in Theorem 8 is not necessarily the oracle OLS estimator, because this result applies when $\hat{\boldsymbol{\beta}}_{\lambda}$ has incorrectly specified the sign vector. A special case of Theorem 8 occurs when $\hat{\boldsymbol{\beta}}_{\lambda}$ recovers the correct sign vector. Therefore, for an orthogonal covariate matrix when $\hat{\boldsymbol{\beta}}_{\lambda}$ has captured the true sign vector, there exists a ν such that $\hat{\alpha}_{\lambda,\nu}\hat{\boldsymbol{\beta}}_{\lambda}$ is closer to $\hat{\boldsymbol{\beta}}^*_{OLS}$, the unbiased oracle OLS estimator, than $\hat{\boldsymbol{\beta}}_{\lambda}$.

The above results are specific to an orthogonal $\mathbf{X}$ matrix, which simplifies Lasso and OLS calculations, but is not generally practical in real data applications. The stronger result, shown in Theorem 9, replicates Theorem 8, for a non-orthogonal $\mathbf{X}$ matrix. The compromise that must be made, however, is the assumption that $\hat{\boldsymbol{\beta}}_{\lambda}$ has recovered the true sign vector, $\mathbf{s}$. This is not a guarantee, but it is a necessary condition of the bias reduction capabilities of the α_ν-Modification. As stated above, the modification is only capable of amplifying the direction of the penalized estimates, not changing it. If the sign vector is incorrect, the α_ν-Modification will not necessarily be able to reduce the overall bias of the estimates.

Theorem 9. *Let* $\boldsymbol{\beta}^*_{OLS}$ *be the oracle OLS estimator for the linear model and let* $\boldsymbol{X}_1$*, the submatrix of* $\boldsymbol{X}$ *including only the active coefficients in* $\boldsymbol{\beta}^*$*, be non-singular. There exists a value* ν *for any fixed* λ *such that*

$$||\hat{\boldsymbol{\beta}}^*_{OLS} - \hat{\boldsymbol{\beta}}^1_{\lambda}||_2^2 > ||\hat{\boldsymbol{\beta}}^*_{OLS} - \hat{\alpha}_{\lambda,\nu}\hat{\boldsymbol{\beta}}^1_{\lambda}||_2^2 \tag{3.13}$$

whenever the Lasso estimate, $\hat{\boldsymbol{\beta}}_{\lambda}$*, has recovered the true sign vector,* $\boldsymbol{s}$*.*

The conclusion from Theorem 9 is that, for the linear model, there will always exist a ν that allows the α_ν-Modification to reduce the bias of the Lasso for a fixed λ, as long as that λ leads to a Lasso estimate that recovers the sign vector of the true model. To increase the chances of finding this combination of ν and λ, the tuning parameter space must be thoroughly explored through K-fold CV.

3.4 Numerical Results

This section discusses the results of simulation studies on the Lasso penalty for linear, logistic, and Poisson regression. The primary methods of estimation considered are the traditional Lasso (Lasso), α-Modified Lasso (α-Mod), α_ν-Modified Lasso (α_ν-Mod), and Relaxed Lasso (Relaxed). Tuning parameters are chosen via 5-fold CV.

For all methods, both a minimum APE approach and a 1SE Rule were implemented. For the linear regression simulations and large sample size logistic and Poisson regression simulations, only the 1SE results are reported. For logistic and Poisson regression with a smaller sample size, $n = 100$, only the minimum APE is reported. Implementation of a 1SE Rule for the multi-dimensional λ of the α_ν and Relaxed methods can be complicated, even for the linear model. In Chapter 2, the strategy used was that of Stallrich et al. (2020), where the the APEs for a dense grid of λ values are calculated, and all sets of tuning parameters that have an APE less than or equal to the minimum plus one standard error are set aside. The combination that gave the estimates with the smallest ℓ_1-norm is taken as the optimal. This approach appears suitable for the linear model for all sample sizes considered, but when used for GLM coefficient estimates for smaller sample sizes, it gave unpredictable results. The question of how to evaluate the model complexity of GLM estimates is still an open one, so the results of the 1SE Rule for smaller sample sizes are reported in Appendix B.2.

The average coefficient bias (CB) was recorded where coefficient bias is calculated as $\|(\beta_0^*, \boldsymbol{\beta}^{*T})^T - (\hat{\beta}_0, \hat{\boldsymbol{\beta}}^T)^T\|_2$ where, for α- and α_ν-Modified approaches, $(\hat{\beta}_0, \hat{\boldsymbol{\beta}}^T)^T =$

$(\hat{\beta}^{\alpha}_{0,\lambda}, \hat{\alpha}_{\lambda,\nu}\hat{\boldsymbol{\beta}}^T_{\lambda})^T$. The average deviance (Dev) is also reported, where deviance is found to be: $\frac{2\phi}{a(\phi)}\sum_{i=1}^{n}(y_i(g(y_i) - b'(\hat{y}_{\lambda,i}) - b(g(y_i)) + \hat{y}_{\lambda,i})$. Once again, for α- and α_ν-Modified methods, $\hat{y}_i = \hat{\beta}^{\alpha}_{0,\lambda} + \hat{\alpha}_{\lambda,\nu}\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_{\lambda}$. Size represents the average number of active coefficients—other than the intercept—included in the model; the average number of false negatives (FN) are the number of coefficients active in $\boldsymbol{\beta}^*$, but not active in $\hat{\boldsymbol{\beta}}_{\lambda}$.

A few comparative models were included in these results to provide context for the difficulty of the estimation problem. The Oracle ML estimator is the unpenalized maximum likelihood estimator using only the truly active coefficients—knowledge only available because the data is simulated. It is reported here as a comparison tool for the other methods of estimation. The average Coefficient Bias and Deviance for these estimates are reported in the following tables. As an evaluation of the difficulty of the variable selection problem for Logistic and Poisson Regression, the model size and false positives for two Lasso solutions are reported: Lasso FPFN and Lasso FNFP. To find Lasso FPFN and Lasso FNFP, the Lasso solutions for all $\lambda \in \Lambda$ were found and evaluated in terms of their model selection results. Of the subset of Λ giving the smallest number of falsely included parameters, the Lasso FPFN estimates had the smallest number of false negatives. The Lasso FNFP model selection results are found by choosing the λ that results in the fewest number of false positives among the models with the fewest number of false negatives.

3.4.1 Linear Regression With Independent Predictors

The following is a simulation study of linear data where $p = 50$, $p^* = 5$, $n = \{100, 500\}$, and $\mathbf{X}$ was drawn from a multivariate Normal distribution with mean $\mathbf{0}$ and covariance $\Sigma = \sigma^2\mathbf{I}_p$. $\boldsymbol{\beta}^*$ was generated from a Gamma distribution with shape 5 and scale 0.25 where negative and positive signs were randomly assigned with equal probability. For each data set, σ was determined by a fixed Signal-to-Noise ratio (SNR) of 5 where SNR is calculated: SNR $= (\boldsymbol{\beta}^{*T}\Sigma\boldsymbol{\beta}^{*})/(\sigma^2)$ (Homrighausen and McDonald 2018). The results of this simulation study are given in Table 3.1.

Both the α- and α_ν-Modifications result in lower average coefficient bias and smaller average model sizes than the Lasso with a 1SE Rule. The α_ν-Modification tends to be more conservative than the α-Modification, with larger average model sizes but fewer average false negatives. It is also of note that the average value of the α estimate for both these approaches is very similar. That is, the α_ν-Modification seems to choose relatively small values of ν and the penalty for α may not be especially necessary for these types of problems. The average coefficient bias for the two new methods are similar to the Relaxed Lasso with a 1SE Rule, though the Relaxed Lasso has a tendency to return sparser models. Consistent with the simulation study results from Chapter 2, the α-Modification appears highly competitive with the Relaxed Lasso in terms of variable selection and coefficient bias, however the α-Modification has an advantage over the Relaxed Lasso in computational efficiency.

Notice that the α- and α_ν-Modified results are nearly identical. This is unsurprising, given the conclusions from Chapter 2. It was hypothesized that any λ such that the α-Modification resulted in a higher APE than the unmodified APE would also underselect and be unlikely to be chosen as the optimal model. Indeed, this seems to be the case and the ν chosen using α_ν-Modified CV appears very close to 0, meaning that the penalty on α is unnecessary. In terms of model size and false negatives, it is clear that the α_ν-Modification results in models that are a compromise between the regular Lasso and the α-Modified Lasso.

Table 3.1: Average results from 500 replications of linear regression data with $p^* = 5, p = 50$. Standard errors given in parentheses.

		α	CB	Dev	Size	FN	Time
$n = 100$	Lasso (Min)		0.6	1.52	16.37	0.03	0.10
			(0.2)	(0.74)	(5.9)	(0.18)	(0.02)
	α-Mod (Min)	1.19	0.49	1.61	10.18	0.07	0.24
		(0.07)	(0.2)	(0.77)	(5.34)	(0.28)	(0.01)
	α_ν-Mod (Min)	1.18	0.49	1.61	10.26	0.07	7.69
		(0.07)	(0.19)	(0.77)	(5.01)	(0.28)	(0.25)
	Relaxed (Min)		0.44	1.67	6.29	0.2	1.44
			(0.22)	(0.79)	(3.77)	(0.45)	(0.16)
	Lasso (1SE)		0.71	2.00	8.23	0.1	0.1
			(0.24)	(0.94)	(2.96)	(0.32)	(0.02)
	α-Mod (1SE)	1.39	0.59	1.91	5.77	0.21	0.24
		(0.16)	(0.23)	(0.89)	(1.82)	(0.49)	(0.01)
	α_ν-Mod (1SE)	1.14	0.62	1.94	6.8	0.18	7.69
		(0.12)	(0.23)	(0.9)	(2.38)	(0.45)	(0.25)
	Relaxed (1SE)		0.62	2.00	5.18	0.31	1.44
			(0.25)	(0.94)	(1.29)	(0.56)	(0.16)
	Oracle OLS		0.29	1.74			
			(0.12)	(0.79)			
$n = 500$	Lasso (Min)		0.25	1.81	15.4	0.00	0.11
			(0.08)	(0.78)	(5.71)	(0.04)	(0.03)
	α-Mod (Min)	1.09	0.19	1.84	8.26	0.00	0.28
		(0.03)	(0.07)	(0.79)	(3.73)	(0.04)	(0.03)
	α_ν-Mod (Min)	1.09	0.19	1.84	8.45	0.00	7.89
		(0.03)	(0.07)	(0.79)	(3.92)	(0.06)	(0.25)
	Relaxed (Min)		0.15	1.84	5.45	0.01	1.79
			(0.07)	(0.79)	(2.16)	(0.12)	(0.21)
	Lasso (1SE)		0.37	1.97	5.87	0.01	0.11
			(0.1)	(0.84)	(1.39)	(0.12)	(0.03)
	α-Mod (1SE)	1.26	0.32	1.94	5.00	0.05	0.28
		(0.11)	(0.11)	(0.84)	(0.37)	(0.24)	(0.03)
	α_ν-Mod (1SE)	1.17	0.32	1.94	5.13	0.03	7.89
		(0.08)	(0.11)	(0.84)	(0.63)	(0.18)	(0.25)
	Relaxed (1SE)		0.34	1.96	5.02	0.03	1.79
			(0.12)	(0.84)	(0.29)	(0.17)	(0.21)
	Oracle OLS		0.13	1.85			
			(0.05)	(0.79)			

3.4.2 Logistic Regression With Independent Predictors

The model for the following results is:

$$E[y_i|\mathbf{x}_i] = p_i = \frac{e^{\beta_0+\mathbf{x}_i^T\boldsymbol{\beta}}}{1+e^{\beta_0+\mathbf{x}_i^T\boldsymbol{\beta}}} \tag{3.14}$$

where $y_i \sim \text{Bernoulli}(p_i)$. These results represent replications of logistic regression data with $p^* = 5, p = 20$, and n varying from 100 to 1000. All coefficients were set such that

$$\sum_{i=0}^{p^*} |\beta_i| = 1 \quad \text{and} \quad |\beta_0| = |\beta_1| = ... = |\beta_{p^*}|\ , \tag{3.15}$$

as in Li and Lederer (2019). The elements of $\mathbf{X}$ are independently drawn from a Normal distribution with mean 0 and standard deviation 0.5. Results for $n = 100$ are displayed in Table 3.2 and for $n = \{500, 1000\}$ in Table 3.3. For $n = 100$, it seems clear that a 1SE Rule is particularly unstable in logistic regression, so the tuning parameters for each method are chosen using 5-fold CV with a minimum APE rule. For $n = \{500, 1000\}$, a 1SE Rule is more reliable so tuning parameters are chosen using 5-fold CV with a 1SE Rule.

The results in Table 3.2 demonstrate, first of all, that this is a challenging variable selection problem. Lasso FPFN tends to underselect and Lasso FNFP overselects by an average of nearly 4 coefficients. Intriguingly, the α-Modification returns variable selection results that are the most similar to Lasso FPFN, giving the fewest number of false negatives among the models with the fewest number of false positives. The α_ν-Modification, in contrast, appears able to reduce the number of false negatives without overselecting as much as the unmodified Lasso. The α_ν-Modification also gives an average coefficient bias that is smaller than that of the α-Modification, and comparable to the Relaxed Lasso.

For the larger sample size, α- and the α_ν-Modifications reduce both the average

Table 3.2: Average results from 500 replications of logistic regression data with $p^* = 5, p = 20$, and $n = 100$. Optimal models chosen with 5-fold CV using the minimum APE. Standard errors given in parentheses.

	α	CB	Dev	Size	FN	Time
Lasso		1.43	2.87	9.46	0.45	0.24
		(0.31)	(0.15)	(3.38)	(0.91)	(0.2)
α-Mod	3.55	1.96	2.99	3.67	2.43	1.87
	(4.89)	(0.46)	(0.2)	(3.39)	(1.94)	(0.58)
α_ν-Mod	1.45	1.5	2.9	8.37	0.77	43.94
	(1.56)	(0.35)	(0.16)	(3.95)	(1.33)	(2.15)
Relaxed		1.51	2.89	6.63	1.13	12.47
		(0.39)	(0.16)	(3.9)	(1.39)	(10.35)
Oracle ML		1.01	2.89			
		(0.46)	(0.11)			
Lasso FPFN				3.16	1.86	
				(1.25)	(1.24)	
Lasso FNFP				8.54	0.00	
				(3.46)	(0.00)	

coefficient bias and average deviance of the unmodified Lasso. When compared to the α-Modification, the α_ν-Modification reliably results in smaller average coefficient bias (CB), often similar to the Relaxed Lasso, though generally larger. Despite the smaller average coefficient bias, the α_ν-Modification often has a similar model size to the α-Modification. That is, the α_ν-Modification is able to reduce the overall coefficient bias of the α-Modification and maintain its accurate variable selection properties. The α_ν-Modification improves upon the Relaxed Lasso in terms of average variable selection results for both $n = 500$ and $n = 1000$, suggesting that the α_ν-Modification is not only competitive with, but often better than the Relaxed Lasso in this setting.

Table 3.3: Average results from 500 replications of logistic regression data with $p^* = 5, p = 20$. Optimal models chosen with 5-fold CV using a 1SE Rule. Standard errors given in parentheses.

		α	CB	Dev	Size	FN	Time
	Lasso		1.1	2.99	5.81	0.00	0.36
			(0.18)	(0.05)	(1.2)	(0.00)	(0.02)
	α-Mod	3.29	0.82	2.96	4.97	0.08	5.36
		(0.75)	(0.25)	(0.05)	(0.43)	(0.29)	(0.25)
	α_ν-Mod	1.01	0.58	2.94	5.02	0.06	126.15
		(0.06)	(0.23)	(0.05)	(0.48)	(0.27)	(3.66)
$n = 500$	Relaxed		0.55	2.95	4.78	0.24	12.29
			(0.32)	(0.05)	(0.5)	(0.45)	(1.17)
	Oracle ML		0.41	2.94			
			(0.14)	(0.05)			
	Lasso FPFN				4.99	0.01	
					(0.09)	(0.09)	
	Lasso FNFP				5.01	0.00	
					(0.09)	(0.00)	
	Lasso		0.91	2.98	5.53	0.00	0.58
			(0.14)	(0.03)	(0.97)	(0.00)	(0.03)
	α-Mod	3.42	0.61	2.96	5.01	0.00	15.02
		(0.75)	(0.18)	(0.03)	(0.18)	(0.04)	(0.85)
	α_ν-Mod	1.01	0.36	2.95	5.00	0.00	342.41
		(0.06)	(0.12)	(0.03)	(0.06)	(0.04)	(17.12)
$n = 1000$	Relaxed		0.35	2.95	4.92	0.09	17.34
			(0.23)	(0.04)	(0.3)	(0.28)	(1.41)
	Oracle ML		0.29	2.95			
			(0.09)	(0.03)			
	Lasso FPFN				5.00	0.00	
					(0.00)	(0.00)	
	Lasso FNFP				5.00	0.00	
					(0.00)	(0.00)	

3.4.3 Logistic Regression With Correlated Predictors

In this section, we tackle the more difficult problem of correlated predictors, once again using Model (3.14). $\mathbf{X}$ is drawn from a multivariate Normal distribution with mean $\mathbf{0}$ and covariance Σ_X, where $\Sigma_{X,ii} = 0.5$ and $\Sigma_{X,ij} = \rho \times 0.5$ whenever $i \neq j$. The correlation between columns of $\mathbf{X}$, ρ, was set to 0.5. This is a more challenging variable selection problem, as evidenced by the fact that all methods have more false negatives on average than in the uncorrelated predictors setting.

A similar pattern to independent predictors is shown for correlated predictors when

$n = 100$ in Table 3.4. The α- and α_ν-Modifications are both able to reduce the average model size of the traditional Lasso. The average coefficient bias of the α-Modification is larger than the α_ν-Modification and the Relaxed Lasso. In comparison with the Relaxed Lasso, the α_ν-Modification has smaller average coefficient bias and a smaller average number of false positives, suggesting that the α_ν-Modification may be preferable in this especially challenging variable selection problem.

Table 3.4: Average results from 500 replications of logistic regression data with $p^* = 5, p = 20$, and $n = 100$ with correlated predictors. Optimal models chosen with 5-fold CV using the minimum APE. Standard errors given in parentheses.

	α	CB	Dev	Size	FN	Time
Lasso		1.76	2.9	7.81	1.31	0.31
		(0.39)	(0.17)	(3.53)	(1.32)	(0.29)
α-Mod	3.61	2.13	2.99	3.29	2.99	2.3
	(4.86)	(0.5)	(0.19)	(3.09)	(1.58)	(0.96)
α_ν-Mod	1.68	1.83	2.92	6.76	1.65	52.31
	(2.38)	(0.4)	(0.17)	(3.94)	(1.51)	(8.05)
Relaxed		1.84	2.92	5.58	1.95	17.08
		(0.41)	(0.18)	(3.53)	(1.46)	(15.56)
Oracle ML		1.27	2.9			
		(0.61)	(0.13)			
Lasso FPFN				2.35	2.83	
				(1.2)	(1.21)	
Lasso FNFP				11.6	0.00	
				(4.34)	(0.00)	

When $n = \{500, 1000\}$ in Table 3.5, the Lasso FPFN and Lasso FNFP results demonstrate that accurate variable selection is more achievable. Once again the α_ν-Modification and the α-Modification improve both the average coefficient bias and the variable selection results of the unmodified Lasso. They are more competitive with the Relaxed Lasso, but tend to give more accurate variable selection results and slightly larger—though similar—average coefficient bias. The two new methods are, on average, more similar in terms of variable selection than coefficient bias, with the average coefficient

Table 3.5: Average results from 500 replications of logistic regression data with $p^* = 5, p = 20$ and correlated predictors. Optimal models chosen with 5-fold CV using a 1SE Rule. Standard errors given in parentheses.

		α	CB	Dev	Size	FN	Time
	Lasso		1.30	3.00	5.98	0.2	0.43
			(0.23)	(0.1)	(1.54)	(0.42)	(0.09)
	α-Mod	2.8	1.24	3.00	4.76	0.65	5.91
		(1.98)	(0.36)	(0.1)	(1.46)	(0.92)	(1.04)
	α_ν-Mod	1 (0.05)	1.00	2.97	5.04	0.50	141.04
			(0.37)	(0.1)	(1.43)	(0.75)	(20.03)
$n = 500$	Relaxed		0.96	2.97	4.51	0.76	15.89
			(0.44)	(0.1)	(1.2)	(0.81)	(3.22)
	Oracle ML		0.5	2.95			
			(0.18)	(0.1)			
	Lasso FPFN				4.57	0.43	
					(0.67)	(0.67)	
	Lasso FNFP				5.67	0.00	
					(1.3)	(0.00)	
	Lasso		1.08	2.99	5.89	0.04	0.66
			(0.17)	(0.09)	(1.21)	(0.19)	(0.1)
	α-Mod	2.19	0.93	2.98	5.09	0.19	15.9
		(1.1)	(0.28)	(0.09)	(0.9)	(0.43)	(2.81)
	α_ν-Mod	1 (0.05)	0.69	2.97	5.13	0.17	371.38
			(0.29)	(0.09)	(0.88)	(0.39)	(49.97)
$n = 1000$	Relaxed		0.62	2.96	4.79	0.35	21.01
			(0.4)	(0.09)	(0.79)	(0.52)	(3.21)
	Oracle ML		0.35	2.95			
			(0.11)	(0.09)			
	Lasso FPFN				4.84	0.16	
					(0.37)	(0.37)	
	Lasso FNFP				5.24	0.00	
					(0.65)	(0.00)	

bias of the α_ν-Modification tending to be smaller than the α-Modification. Once again, the α_ν-Modification results in fewer average false negatives than the Relaxed Lasso and similar average coefficient bias.

3.4.4 Poisson Regression With Independent Predictors

The model for the following results is:

$$E[y_i|\mathbf{x}_i] = \lambda_i = e^{\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}} \tag{3.16}$$

where $y_i \sim \text{Poisson}(\lambda_i)$. β_0 was randomly chosen from $\{-1, -0.5\}$ and $\boldsymbol{\beta}$ were selected such that

$$\sum_{i=1}^{p^*} \beta_i^2 = 1 \quad \text{and} \quad |\beta_0| = |\beta_1| = ... = |\beta_{p^*}| \tag{3.17}$$

where the signs were randomly assigned. This approach is similar to that of Kibria et al. (2015), except for the varying signs of the coefficients. Elements of $\mathbf{X}$ were independently generated from a Normal distribution with mean 0 and standard deviation 0.5. The results in Table 3.6 are from 500 replications of data with $p = 20$, $p^* = 5$ and $n = 100$. Table 3.7 shows the results of 500 replications of data with the same p and p^* but for $n = \{500, 1000\}$ Additional results are given in Appendix B.2.

Table 3.6: Average results from 500 replications of Poisson regression data with $p^* = 5, p = 20$, and $n = 100$. Optimal models chosen with 5-fold CV using the minimum APE. Standard errors given in parentheses.

	α	CB	Dev	Size	FN	Time
Lasso		0.68	0.84	9.63	0.67	0.16
		(0.19)	(0.14)	(3.99)	(1.08)	(0.02)
α-Mod	2.09	0.83	0.89	6.23	1.66	1.19
	(1.6)	(0.22)	(0.18)	(4.84)	(1.78)	(0.04)
α_ν-Mod	1.57	0.72	0.87	7.96	1.11	38.76
	(1.51)	(0.2)	(0.16)	(4.68)	(1.52)	(0.86)
Relaxed		0.72	0.86	6.37	1.24	5.98
		(0.21)	(0.15)	(3.71)	(1.35)	(1.07)
Oracle ML		0.47	2.98			
		(0.18)	(0.2)			
Lasso FPFN				3.13	1.94	
				(1.23)	(1.26)	
Lasso FNFP				9.62	0.00	
				(4.38)	(0.00)	

The Lasso FPFN and Lasso FNFP results demonstrate that, when $n = 100$, the variable selection problem is difficult. The model containing no false negatives has an average of nearly 5 false positives. As such, the smallest average false negatives is achieved by the traditional Lasso, which overselects more than the Relaxed Lasso, the α-, and

α_ν-Modified Lasso. The compromise made by these three methods is fewer false positives. It is noteworthy, however, how similar the α- and α_ν-Modifications are to the Relaxed Lasso.

Table 3.7: Average results from 500 replications of Poisson regression data with $p^* = 5, p = 20$. Optimal models chosen with 5-fold CV using a 1SE Rule. Standard errors given in parentheses.

		α	CB	Dev	Size	FN	Time
	Lasso		0.47 (0.11)	0.98 (0.08)	5.69 (1.09)	0.02 (0.16)	0.28 (0.02)
	α-Mod	2.39 (0.8)	0.41 (0.14)	0.95 (0.08)	4.96 (0.81)	0.2 (0.53)	4.32 (0.23)
	α_ν-Mod	2 (0.41)	0.32 (0.15)	0.94 (0.08)	5.01 (0.82)	0.17 (0.48)	119.41 (4.09)
$n = 500$	Relaxed		0.3 (0.18)	0.94 (0.08)	4.66 (0.74)	0.4 (0.65)	8.22 (0.62)
	Oracle ML		0.2 (0.07)	3.11 (0.11)			
	Lasso FPFN				4.95 (0.3)	0.05 (0.3)	
	Lasso FNFP				5.05 (0.25)	0.00 (0.00)	
	Lasso		0.38 (0.08)	0.97 (0.07)	5.37 (0.74)	0.00 (0.00)	0.41 (0.02)
	α-Mod	2.44 (0.5)	0.29 (0.1)	0.95 (0.07)	4.99 (0.18)	0.02 (0.15)	12.33 (0.96)
	α_ν-Mod	2.24 (0.41)	0.19 (0.08)	0.94 (0.07)	5 (0.22)	0.02 (0.14)	332.93 (26.18)
$n = 1000$	Relaxed		0.16 (0.11)	0.94 (0.07)	4.9 (0.32)	0.1 (0.32)	11.01 (0.62)
	Oracle ML		0.13 (0.05)	3.12 (0.09)			
	Lasso FPFN				5.00 (0.00)	0.00 (0.00)	
	Lasso FNFP				5.00 (0.00)	0.00 (0.00)	

Similarly to the results of the logistic regression simulations, when $n = 500$, the α_ν-Modification results in fewer false negatives on average than the Relaxed Lasso or the α-Modified Lasso. Its average coefficient bias is smaller than that of the α-Modification, but larger than the Relaxed Lasso. This pattern is maintained when $n = 1000$, though the differences in variable selection are less pronounced between the α- and α_ν-Modifications. Both result in fewer false negatives on average than the Relaxed Lasso.

3.4.5 Poisson Regression With Correlated Predictors

Similarly to the above, for these simulations $\mathbf{X}$ is drawn from a multivariate Normal distribution with mean $\mathbf{0}$ and covariance Σ_X, where $\Sigma_{X,ii} = 0.5$ and $\Sigma_{X,ij} = \rho \times 0.5$ whenever $i \neq j$. The correlation between columns of $\mathbf{X}$, ρ, was set to 0.5. Table 3.8 shows the results for $n = 100$. Lasso FNFP demonstrates that an average model size as large as 13 is necessary to correctly select the 5 truly active coefficients in the model. As a result, all methods have false negatives, though the α_ν-Modification has fewer on average than the α-Modification or the Relaxed Lasso.

Table 3.8: Average results from 500 replications of poisson regression data with $p^* = 5, p = 20$, and $n = 100$ with correlated predictors. Optimal models chosen with 5-fold CV using the minimum APE. Standard errors given in parentheses.

	α	CB	Dev	Size	FN	Time
Lasso		0.87	0.88	6.78	1.94	0.19
		(0.2)	(0.16)	(4.32)	(1.63)	(0.03)
α-Mod	2.05	0.96	0.92	4.13	2.92	1.23
	(1.95)	(0.22)	(0.17)	(4.3)	(1.79)	(0.07)
α_ν-Mod	1.48	0.9	0.9	5.74	2.32	39.55
	(1.32)	(0.21)	(0.17)	(4.61)	(1.73)	(1.33)
Relaxed		0.91	0.9	4.78	2.53	7.76
		(0.23)	(0.17)	(3.72)	(1.63)	(6.3)
Oracle ML		0.57	2.97			
		(0.22)	(0.65)			
Lasso FPFN				2.16	3.11	
				(1.15)	(1.15)	
Lasso FNFP				13.45	0 (0)	
				(4.54)		

Results for $n = 500$ and $n = 1000$ are given in Table 3.9. The α_ν-Modification, in comparison to the α-Modification and the Relaxed Lasso, is able to return the fewest average false negatives while maintaining similar coefficient bias to the Relaxed Lasso.

Table 3.9: Average results from 500 replications of poisson regression data with $p^* = 5, p = 20$ and correlated predictors. Optimal models chosen with 5-fold CV using a 1SE Rule. Standard errors given in parentheses.

		α	CB	Dev	Size	FN	Time
	Lasso		0.63 (0.16)	0.99 (0.09)	5.38 (1.85)	0.71 (0.97)	0.33 (0.03)
	α-Mod	2.82 (5.76)	0.62 (0.2)	0.98 (0.09)	4.23 (2.03)	1.3 (1.49)	4.29 (0.28)
	α_ν-Mod	1.56 (0.6)	0.57 (0.21)	0.97 (0.09)	4.51 (2.07)	1.16 (1.38)	120.45 (5.67)
$n = 500$	Relaxed		0.53 (0.24)	0.96 (0.09)	4.2 (1.68)	1.24 (1.24)	10.98 (3.73)
	Oracle ML		0.24 (0.09)	3.05 (0.53)			
	Lasso FPFN				4.02 (1.07)	0.99 (1.07)	
	Lasso FNFP				6.34 (1.75)	0 (0)	
	Lasso		0.48 (0.12)	0.97 (0.07)	5.73 (1.24)	0.1 (0.32)	0.48 (0.05)
	α-Mod	1.71 (0.6)	0.42 (0.15)	0.96 (0.07)	5.1 (1.07)	0.27 (0.63)	12.45 (1)
	α_ν-Mod	1.55 (0.41)	0.35 (0.16)	0.95 (0.07)	5.14 (1.07)	0.26 (0.56)	338.47 (24.55)
$n = 1000$	Relaxed		0.32 (0.2)	0.95 (0.07)	4.73 (0.91)	0.47 (0.66)	14.89 (2.35)
	Oracle ML		0.16 (0.06)	3.04 (0.49)			
	Lasso FPFN				4.75 (0.51)	0.25 (0.51)	
	Lasso FNFP				5.32 (0.72)	0 (0)	

3.5 Raisin Application

To evaluate how the α- and α_ν-Modifications perform on real data, we consider the work of Cinar et al. (2020). The data was made available through the UCI Machine Learning

Repository[1]. This study uses a machine vision system to evaluate various characteristics of two types of Turkish grown raisins: Kecimen and Besni. While many methods exist for classifying raisins as Kecimen or Besni, they are often expensive, time-consuming, and prone to human error. As such, Cinar et al. (2020) sought to use information from the machine vision system to develop a model for classifying the raisin grains. The data consists of 900 observations of 450 Kecimen raisins and 450 Besni raisins.

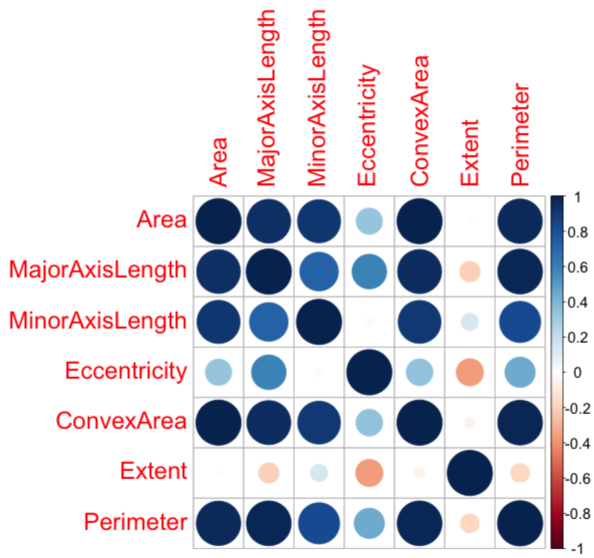

Figure 3.2: Correlation between the seven predictive variables.

In their work, Cinar et al. (2020) use three different methods of classification: logistic regression, Mulitlayer Perceptron, and Support Vector Machine, but do not carry out variable selection among the factors collected in their data. These factors are Area (the number of pixels within the boundaries of the raisin), Perimeter (the distance between the boundaries of the raisin and the pixels around it), Major Axis Length (the length of the longest line that can be drawn on the raisin), Minor Axis Length (the length of the

[1] `https://archive.ics.uci.edu/ml/index.php`

shortest line that can be drawn on the raisin), Eccentricity (of the ellipse of the raisin), Convex Area (the number of pixels in the smallest convex shell of the raisin's region), and Extent (the ratio of the region formed by the raisin to the total number of pixels in the bounding box). It is clear from a description of the predictors that there is potentially high correlation among the factors, because many of these factors concern the size of the raisin. Figure 3.2 demonstrates their relatively high correlation.

Because the factors are highly correlated, it may be prudent to carry out variable selection via penalized estimation. To evaluate the new methods proposed in this paper, we consider an unpenalized ML estimator, the unmodified Lasso, α- and α_ν-Modified Lasso, and finally the Relaxed Lasso. We acknowledge that the Lasso is perhaps an imperfect choice for highly correlated predictors, however our simulation results indicate that it may still be useful. Additional study of the α- and α_ν-Modifications on the Elastic Net penalty, as well as a Relaxed Elastic Net, may prove fruitful.

We carried out the penalized estimation on 80% of the data, holding 20% aside for evaluation. The results of that evaluation may be found in Table 3.10, including the Model Size, Deviance, Time in seconds the calculations took to carry out, and Accuracy. Accuracy, as in Cinar et al. (2020), is:

$$\frac{TP + TN}{TP + FP + TN + FN} \times 100 \tag{3.18}$$

where TP represents the number of observations in the holdout set that were correctly classified as Besni, TN are those that were correctly classified as Kecimen, FP are those that were classified as Besni inaccurately and FN are those that were incorrectly classified as Kecimen. Accuracy would be 100% if every point in the holdout set was accurately classified.

Note that all the penalized methods had higher accuracy and lower deviance than the

ML estimator, with the exception of the unmodified Lasso, which chose all 7 predictors; it is therefore clear that model reduction is desirable for this data. Furthermore, as classification is the ultimate goal, in this application it is prudent to prioritize model accuracy over variable selection. As such, the 1SE Rule methods, which return the sparsest models but lower accuracy that the minimum APE methods, are less desirable. The two methods with the highest accuracy were the α- and α_ν-Modified estimates with a minimum APE rule, where the α_ν-Modifcation resulted in slightly smaller deviance. The Relaxed Lasso with a minimum APE rule had very similar accuracy to the α- and α_ν-Modifications, but had higher deviance and a more complex model. Overall, the α- and α_ν-Modified Lasso approaches returned sparse models with high accuracy and low deviance, although significant reduction in computation time from α_ν-Modified to α-Modified Lasso make the α-Modified approach more desirable if computation time is a considerable concern.

Table 3.10: Summary of the results from the holdout set of Raisin data.

	Accuracy	Model Size	Deviance	Time (Seconds)
OLS	0.856	7	513.792	0.432
Lasso Min	0.872	7	514.840	0.774
Lasso 1SE	0.872	3	504.137	0.774
Mod Min	0.889	3	507.691	10.523
Mod 1SE	0.878	1	500.041	10.523
α_ν Min	0.889	3	506.385	292.536
α_ν 1SE	0.861	2	509.895	292.536
Relaxed Min	0.883	4	512.630	19.510
Relaxed 1SE	0.861	2	507.895	19.510

Table 3.11: Variables chosen via each method for the Raisin data.

OLS	Area, MajorAxisLength, MinorAxisLength, Eccentricity, ConvexArea, Extent, Perimeter
Lasso Min	Area, MajorAxisLength, MinorAxisLength, Eccentricity, ConvexArea, Extent, Perimeter
Lasso 1SE	Eccentricity, Extent, Perimeter
Mod Min	MinorAxisLength, Extent, Perimeter
Mod 1SE	MajorAxisLength
α_ν Min	MinorAxisLength, Extent, Perimeter
α_ν 1SE	MajorAxisLength, Perimeter
Relaxed Min	MinorAxisLength, Eccentricity, Extent, Perimeter
Relaxed 1SE	MajorAxisLength, Perimeter

3.6 Discussion

The α-Modification, proposed in Chapter 2 for the linear model, is an attractive option for reducing the bias of penalized estimates and is adapted in this chapter to address the challenges of GLMs. Under GLMs, the calculation of $\hat{\alpha}_\lambda$ is more involved than under linear models, because of the necessity of accounting for an intercept term and because it is generally impossible to find closed form solutions. As a result, an algorithm was proposed to find $\hat{\beta}^{\alpha}_{0,\lambda}$ and $\hat{\alpha}_\lambda$ under any GLM. The result in Chapter 2, showing that $\hat{\alpha}_\lambda \geq 1$ for any λ, was generalized under certain conditions on the penalty and log-likelihood. It is important to emphasize the generalizablilty of the α-Modified approach: the algorithm provided in this paper can be used for any penalty.

As discussed in Chapter 2, even in the linear model the α-Modification is not guaranteed to reduce the bias of penalized estimates, though simulation results indicate that it often does when paired with Cross Validation. In this chapter, a penalized α-Modification, the α_ν-Modification, has been proposed to ensure bias reduction of estimates. For the Lasso penalty on the linear model, it has been shown that there exists a value of ν for any λ that results in accurate sign recovery, such that the overall bias of the coefficient vector is reduced by multiplying the estimates by $\hat{\alpha}_{\lambda,\nu}$.

The trade-off in reducing coefficient bias through the α_ν-Modification is sometimes, as shown in the simulation results, that overselection can occur. That is, α_ν-Modified CV can reliably reduce coefficient bias, but in doing so it may pick up additional effects that are not necessary to predict the response. Notably, the α-Modification is much faster than either the α_ν-Modification or the Relaxed Lasso and generally achieves similar results.

The fixed sample size results provided in this paper focused on the Lasso penalty because of its popularity, but it may be the case that many of the results can be generalized to other penalties as well. This is one area of future research. Another area of additional study is a different approach to the estimation of α. The calculation of $\hat{\alpha}_\lambda$ and $\hat{\alpha}_{\lambda,\nu}$ described here uses in-sample predictions and observations. As overspecification is a particular concern, the question of whether out-of-sample data can be used to find estimates of α is another subject of further research.

3.A Proofs for Theoretical Results

This section includes the proofs of all theoretical results presented in Chapter 3.

Proof of Theorem 6. We prove this theorem by contradiction. Let $\hat{\beta}_{0,\lambda}$ and $\hat{\boldsymbol{\beta}}_\lambda$ be the unique minimizers of $-\ell(\beta_0, \boldsymbol{\beta}) + P_\lambda(\boldsymbol{\beta})$. Now,

$$\begin{bmatrix} \hat{\beta}_{0,\alpha} \\ \hat{\alpha}_\lambda \end{bmatrix} = \underset{\beta_0,\alpha}{\arg\min} -\ell(\beta_0, \alpha|\hat{\boldsymbol{\beta}}_\lambda, \mathbf{y}, \mathbf{X}) + P_\lambda(\hat{\boldsymbol{\beta}}_\lambda) = \underset{\beta_0,\alpha}{\arg\min} -\ell(\beta_0, \alpha|\hat{\boldsymbol{\beta}}_\lambda, \mathbf{y}, \mathbf{X}) \tag{3.19}$$

where $-\ell(\beta_0, \alpha|\hat{\boldsymbol{\beta}}_\lambda, \mathbf{y}, \mathbf{X})$ iis the negative log-likelihood under Model (1.2) with predictors represented as $g(E[y_i|\mathbf{x}_i]) = \beta_0 + \alpha\mathbf{x}_i^T\boldsymbol{\xi}^*$. First suppose $|\hat{\alpha}_\lambda| < 1$. Note that $\hat{\boldsymbol{\beta}}_\lambda$ minimize the negative log-likelihood under the constraint of the penalty. Expression (3.19) minimizes the negative log-likelihood with respect to α and β_0 without regard to the penalty. Therefore:

$$-\ell(\hat{\beta}_{0,\lambda}^{\alpha}, \hat{\alpha}_\lambda\hat{\boldsymbol{\beta}}_\lambda|\mathbf{y}, \mathbf{X}) \leq -\ell(\hat{\beta}_{0,\lambda}, \hat{\boldsymbol{\beta}}_\lambda|\mathbf{y}, \mathbf{X}). \tag{3.20}$$

We have assumed that $P_\lambda(\cdot)$ is a norm function. A function $p : Z \to \mathbb{R}$ is a norm if it satisfies the following three conditions: (1) $p(z_1 + z_2) \leq p(z_1) + p(z_2)$ for all $z_1, z_2 \in Z$; (2) $p(sz) = |s|p(z)$ for any scalar s and all $z \in Z$; (3) for all $z \in Z$, if $p(z) = 0$, then $z = 0$. It is this second property, known as absolute homogeneity, that informs the contradiction. When $|\hat{\alpha}_\lambda| \in (0, 1)$, $P_\lambda(\hat{\boldsymbol{\beta}}_\lambda) > P_\lambda(\hat{\alpha}_\lambda\hat{\boldsymbol{\beta}}_\lambda)$. Then:

$$-\ell(\hat{\beta}_{0,\lambda}^{\alpha}, \hat{\alpha}_\lambda\hat{\boldsymbol{\beta}}_\lambda|\mathbf{y}, \mathbf{X}) + P_\lambda(\hat{\alpha}_\lambda\hat{\boldsymbol{\beta}}_\lambda) < -\ell(\hat{\beta}_{0,\lambda}, \hat{\boldsymbol{\beta}}_\lambda|\mathbf{y}, \mathbf{X}) + P_\lambda(\hat{\boldsymbol{\beta}}_\lambda). \tag{3.21}$$

Therefore $\hat{\beta}_{0,\lambda}$ and $\hat{\boldsymbol{\beta}}_\lambda$ are not the unique minimizers of $-\ell(\beta_0, \boldsymbol{\beta}) + P_\lambda(\boldsymbol{\beta})$. This is a contradiction, so $|\hat{\alpha}_\lambda| \geq 1$.

We have assumed that the negative log-likelihood is convex and established that

$$-\ell(\hat{\beta}^{\alpha}_{0,\lambda}, \alpha\hat{\boldsymbol{\beta}}_{\lambda}|\mathbf{y}, \mathbf{X}) > -\ell(\hat{\beta}^{\alpha}_{0,\lambda}, \alpha\hat{\boldsymbol{\beta}}_{\lambda}|\mathbf{y}, \mathbf{X}, \alpha = 1) \tag{3.22}$$

when $\alpha \in (-1, 1)$. Therefore, when $\alpha \in (-1, 1)$, the function is increasing in α. Because it is strictly convex, the negative log-likelihood must also be increasing for $\alpha < -1$. Therefore the minimizer $\hat{\alpha}_{\lambda} \geq 1$. $\square$

Proof of Theorem 7. Assume that $\hat{\alpha}_{\lambda,\nu} > 1$; let $d_j = (|\hat{\beta}_{OLS,j}| - \lambda)_+$ and $s_j = \text{sign}(\hat{\beta}_{OLS,j})$ so $\hat{\beta}_{\lambda,j} = s_j d_j$. Then we have the following:

$$\begin{aligned}
\hat{\alpha}_{\lambda,\nu}\hat{\beta}_{\lambda,j^*} &= \Big(\frac{\hat{\boldsymbol{\beta}}_{OLS}^T\hat{\boldsymbol{\beta}}_{\lambda} - \nu}{\hat{\boldsymbol{\beta}}_{\lambda}^T\hat{\boldsymbol{\beta}}_{\lambda}}\Big)\hat{\beta}_{\lambda,j^*} && (3.23)\\
&= \Big(\frac{\sum_{j=1}^p \hat{\beta}_{OLS,j}\hat{\beta}_{\lambda,j} - \nu}{\sum_{j=1}^p \hat{\beta}_{\lambda,j}^2}\Big)\hat{\beta}_{\lambda,j^*} && (3.24)\\
&= \frac{s_{j^*}d_{j^*}\sum_{j\neq j^*}|\hat{\beta}_{OLS,j}|d_j + s_{j^*}d_{j^*}|\hat{\beta}_{OLS,j^*}|d_{j^*} - \nu d_{j^*}s_{j^*}}{\sum_{j=1}^p d_j^2} && (3.25)\\
&= \Big(\frac{d_{j^*}^2}{\sum_{j=1}^p d_j^2}\Big)\hat{\beta}_{OLS,j^*} + \Big(\frac{\sum_{j\neq j^*}(d_j + \lambda)d_j}{\sum_{j=1}^p d_j^2}\Big)\hat{\beta}_{\lambda,j^*} - \frac{\nu d_{j^*}s_{j^*}}{\sum_{j=1}^p d_j^2} && (3.26)\\
&= \Big(\frac{d_{j^*}^2}{\sum_{j=1}^p d_j^2}\Big)\hat{\beta}_{OLS,j^*} + \Big(\frac{\sum_{j\neq j^*} d_j^2}{\sum_{j=1}^p d_j^2}\Big)\hat{\beta}_{\lambda,j^*} + \Big(\frac{\lambda\sum_{j\neq j^*} d_j}{\sum_{j=1}^p d_j^2}\Big)\hat{\beta}_{\lambda,j^*} - \frac{\nu d_{j^*}s_{j^*}}{\sum_{j=1}^p d_j^2} && (3.27)\\
&= w_1\hat{\beta}_{OLS,j^*} + (1 - w_1)\hat{\beta}_{\lambda,j^*} + \Big(\frac{\lambda\sum_{j\neq j^*} d_j}{\sum_{j=1}^p d_j^2}\Big)\hat{\beta}_{\lambda,j^*} - \frac{\nu d_{j^*}s_{j^*}}{\sum_{j=1}^p d_j^2} && (3.28)
\end{aligned}$$

where $w_1 = \left(\frac{d_{j^*}^2}{\sum_{j=1}^p d_j^2}\right)$. Therefore, it is clear that $\alpha_\nu \hat{\beta}_{\lambda,j^*}$ is a convex combination of $\hat{\beta}_{OLS,j^*}$ and $\hat{\beta}_{\lambda,j^*}$ whenever

$$\left(\frac{\lambda \sum_{j \neq j^*} d_j}{\sum_{j=1}^p d_j^2}\right)\hat{\beta}_{\lambda,j^*} - \frac{\nu d_{j^*} s_{j^*}}{\sum_{j=1}^p d_j^2} = 0 \tag{3.29}$$

$$\left(\frac{\lambda \sum_{j \neq j^*} d_j}{\sum_{j=1}^p d_j^2}\right)\hat{\beta}_{\lambda,j^*} = \frac{\nu d_{j^*} s_{j^*}}{\sum_{j=1}^p d_j^2} \tag{3.30}$$

$$\lambda \hat{\beta}_{\lambda,j^*} \sum_{j \neq j^*} d_j = \nu d_{j^*} s_{j^*} \tag{3.31}$$

$$\frac{\lambda \hat{\beta}_{\lambda,j^*} \sum_{j \neq j^*} d_j}{d_{j^*} s_{j^*}} = \nu \tag{3.32}$$

$$\lambda \sum_{j \neq j^*} d_j = \nu. \tag{3.33}$$

Therefore, there exists some ν for any fixed value of $\lambda > 0$ such that the penalized modified estimate is a convex combination of its OLS estimate and its Lasso estimate. $\square$

Proof of Corollary 7.1. Assume that $\nu = \lambda \sum_{j \neq j^*} d_j$ for some nonzero Lasso estimate. Consider the absolute difference between the modified estimate and the OLS estimate:

$$|\hat{\beta}_{OLS,j^*} - \hat{\alpha}_{\lambda,\nu}\hat{\beta}_{\lambda,j^*}| = |\hat{\beta}_{OLS,j^*} - w_1\hat{\beta}_{OLS,j^*} - (1-w_1)\hat{\beta}_{\lambda,j^*}| \tag{3.34}$$

$$= |(1-w_1)\hat{\beta}_{OLS,j^*} - (1-w_1)\hat{\beta}_{\lambda,j^*}| \tag{3.35}$$

$$= (1-w_1)|\hat{\beta}_{OLS,j^*} - \hat{\beta}_{\lambda,j^*}|. \tag{3.36}$$

Therefore:

$$|\hat{\beta}_{OLS,j^*} - \hat{\alpha}_{\lambda,\nu}\hat{\beta}_{\lambda,j^*}| \leq |\hat{\beta}_{OLS,j^*} - \hat{\beta}_{\lambda,j^*}| \tag{3.37}$$

because $0 < w_1 \leq 1$. $\square$

Proof of Theorem 8. Fix $\lambda > 0$ and allow $\mathbf{X}$ to be an orthonormal matrix.

$$
\begin{aligned}
||\hat{\boldsymbol{\beta}}^1_{OLS} - \hat{\boldsymbol{\beta}}^1_{\lambda}||_2^2 &> ||\hat{\boldsymbol{\beta}}^1_{OLS} - \hat{\alpha}_{\lambda,\nu}\hat{\boldsymbol{\beta}}^1_{\lambda}||_2^2 && (3.38)\\
\hat{\alpha}_{\lambda,\nu} &> \frac{\hat{\boldsymbol{\beta}}^{1T}_{OLS}\hat{\boldsymbol{\beta}}_{\lambda} \pm \sqrt{(\hat{\boldsymbol{\beta}}^{1T}_{OLS}\hat{\boldsymbol{\beta}}_{\lambda})^2 - \hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda}(2\hat{\boldsymbol{\beta}}^{1T}_{OLS}\hat{\boldsymbol{\beta}}_{\lambda} - \hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda})}}{\hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda}} && (3.39)\\
\frac{\hat{\boldsymbol{\beta}}^{1T}_{OLS}\hat{\boldsymbol{\beta}}_{\lambda} - \nu}{\hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda}} &> \frac{\hat{\boldsymbol{\beta}}^{1T}_{OLS}\hat{\boldsymbol{\beta}}_{\lambda} \pm \sqrt{(\hat{\boldsymbol{\beta}}^{1T}_{OLS}\hat{\boldsymbol{\beta}}_{\lambda})^2 - \hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda}(2\hat{\boldsymbol{\beta}}^{T}_{OLS}\hat{\boldsymbol{\beta}}_{\lambda} - \hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda})}}{\hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda}} && (3.40)\\
\nu &< \sqrt{(\hat{\boldsymbol{\beta}}^{1T}_{OLS}\hat{\boldsymbol{\beta}}_{\lambda})^2 - \hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda}(2\hat{\boldsymbol{\beta}}^{1T}_{OLS}\hat{\boldsymbol{\beta}}_{\lambda} - \hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda})}. && (3.41)
\end{aligned}
$$

This is only possible when the value under the square root is greater than or equal to zero. So:

$$
\begin{aligned}
(\hat{\boldsymbol{\beta}}^{T}_{OLS}\hat{\boldsymbol{\beta}}_{\lambda})^2 &\geq \hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda}(2\hat{\boldsymbol{\beta}}^{T}_{OLS}\hat{\boldsymbol{\beta}}_{\lambda} - \hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda}) && (3.42)\\
\hat{\boldsymbol{\beta}}^{T}_{OLS}\hat{\boldsymbol{\beta}}_{\lambda}(\hat{\boldsymbol{\beta}}^{T}_{OLS}\hat{\boldsymbol{\beta}}_{\lambda} - 2\hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda}) &\geq -(\hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda})^2 && (3.43)\\
\hat{\alpha}_{\lambda,0}(\hat{\boldsymbol{\beta}}^{T}_{OLS}\hat{\boldsymbol{\beta}}_{\lambda} - 2\hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda}) &\geq -\hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda} && (3.44)\\
\hat{\alpha}_{\lambda,0} &\geq \frac{-\hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda}}{\hat{\boldsymbol{\beta}}^{T}_{OLS}\hat{\boldsymbol{\beta}}_{\lambda} - 2\hat{\boldsymbol{\beta}}^T_{\lambda}\hat{\boldsymbol{\beta}}_{\lambda}} && (3.45)\\
\hat{\alpha}_{\lambda,0} &\geq \frac{(\hat{\boldsymbol{\beta}}^1_{OLS})^T\hat{\boldsymbol{\beta}}^1_{OLS} - 2\lambda\mathbf{s}_{\lambda}^T\hat{\boldsymbol{\beta}}_{OLS} + \lambda^2\mathbf{s}_{\lambda}^T\mathbf{s}_{\lambda}}{(\hat{\boldsymbol{\beta}}^1_{OLS})^T\hat{\boldsymbol{\beta}}^1_{OLS} + 3\lambda\mathbf{s}_{\lambda}^T\hat{\boldsymbol{\beta}}_{OLS} + 2\lambda^2\mathbf{s}_{\lambda}^T\mathbf{s}_{\lambda}}. && (3.46)
\end{aligned}
$$

This last expression arises from the fact that $\hat{\boldsymbol{\beta}}^1_{\lambda} = \hat{\boldsymbol{\beta}}^1_{OLS} - \lambda\mathbf{s}_{\lambda}$ where $\mathbf{s}_{\lambda}$ is the sign vector of the non-zero elements of $\hat{\boldsymbol{\beta}}_{\lambda}$. Inequality (3.46) must always be true because $3\lambda\mathbf{s}_{\lambda}^T\hat{\boldsymbol{\beta}}_{OLS} + 2\lambda^2\mathbf{s}_{\lambda}^T\mathbf{s}_{\lambda} \geq -2\lambda\mathbf{s}_{\lambda}^T\hat{\boldsymbol{\beta}}_{OLS} + \lambda^2\mathbf{s}_{\lambda}^T\mathbf{s}_{\lambda}$. Therefore the left side of the expression is less than or equal to 1 and we have already shown that $\hat{\alpha}_{\lambda,0}$ is always greater than or equal to 1.

The conclusion is that for any fixed $\lambda > 0$, there exists a $\nu > 0$ such that the penalized modified estimates are closer to the OLS estimates than the non-modified Lasso estimates in the case where $\mathbf{X}$ is orthogonal. $\square$

Proof of Theorem 9. Suppose that the Lasso has recovered the sign vector, $\mathbf{s}$ for the true coefficient vector, $\boldsymbol{\beta}^*$. Then $\hat{\boldsymbol{\beta}}^1_\lambda = \hat{\boldsymbol{\beta}}^*_{OLS} - n\lambda(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1$ where $\mathbf{X}_1$ is the covariate matrix including all covariates in the support vector of the true model and $\mathbf{s}_1$ is the sign vector for the active coefficients. $\hat{\boldsymbol{\beta}}^*_{OLS} = (\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{X}_1^T\mathbf{y}$ and $\hat{\boldsymbol{\beta}}^1_\lambda$ is the vector of non-zero entries for $\hat{\boldsymbol{\beta}}_\lambda$. We want to find a ν such that $||\hat{\boldsymbol{\beta}}^*_{OLS} - \hat{\boldsymbol{\beta}}^1_\lambda||_2^2 > ||\hat{\boldsymbol{\beta}}^*_{OLS} - \hat{\alpha}_{\lambda,\nu}\hat{\boldsymbol{\beta}}^1_\lambda||_2^2$, so assume that $\hat{\alpha}_{\lambda,\nu} \neq 1$. Then:

$$\hat{\alpha}_{\lambda,\nu} = \frac{\mathbf{y}^T\mathbf{X}_1\hat{\boldsymbol{\beta}}^*_{OLS} - n\lambda(\hat{\boldsymbol{\beta}}^*_{OLS})^T\mathbf{s}_1 - \nu}{(\hat{\boldsymbol{\beta}}^*_{OLS})^T\mathbf{X}_1^T\mathbf{X}_1\hat{\boldsymbol{\beta}}^*_{OLS} - 2n\lambda\mathbf{s}_1^T\hat{\boldsymbol{\beta}}^*_{OLS} + n^2\lambda^2\mathbf{s}_1^T(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1}. \tag{3.47}$$

Now, we consider when the inequality of interest arises:

$$||\hat{\boldsymbol{\beta}}^*_{OLS} - \hat{\boldsymbol{\beta}}^1_\lambda||_2^2 > ||\hat{\boldsymbol{\beta}}^*_{OLS} - \hat{\alpha}_{\lambda,\nu}\hat{\boldsymbol{\beta}}^1_\lambda||_2^2 \tag{3.48}$$

$$-2(\hat{\boldsymbol{\beta}}^*_{OLS})^T\hat{\boldsymbol{\beta}}^1_\lambda + (\hat{\boldsymbol{\beta}}^1_\lambda)^T\hat{\boldsymbol{\beta}}^1_\lambda > -2\hat{\alpha}_{\lambda,\nu}(\hat{\boldsymbol{\beta}}^*_{OLS})^T\hat{\boldsymbol{\beta}}^1_\lambda + \hat{\alpha}^2_{\lambda,\nu}(\hat{\boldsymbol{\beta}}^1_\lambda)^T\hat{\boldsymbol{\beta}}^1_\lambda \tag{3.49}$$

$$2(\hat{\alpha}_{\lambda,\nu} - 1)(\hat{\boldsymbol{\beta}}^*_{OLS})^T\hat{\boldsymbol{\beta}}^1_\lambda > (\hat{\alpha}^2_{\lambda,\nu} - 1)(\hat{\boldsymbol{\beta}}^1_\lambda)^T\hat{\boldsymbol{\beta}}^1_\lambda \tag{3.50}$$

$$2(\hat{\boldsymbol{\beta}}^*_{OLS})^T\hat{\boldsymbol{\beta}}^1_\lambda > (\hat{\alpha}_{\lambda,\nu} + 1)(\hat{\boldsymbol{\beta}}^1_\lambda)^T\hat{\boldsymbol{\beta}}^1_\lambda \tag{3.51}$$

$$2\hat{\alpha}_{\lambda,\nu}n\lambda(\hat{\boldsymbol{\beta}}^*_{OLS})^T(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1 > (\hat{\alpha}_{\lambda,\nu} - 1)(\hat{\boldsymbol{\beta}}^*_{OLS})^T\hat{\boldsymbol{\beta}}^*_{OLS} \tag{3.52}$$

$$+ (\hat{\alpha}_{\lambda,\nu} + 1)n^2\lambda^2\mathbf{s}_1^T(\mathbf{X}_1^T\mathbf{X}_1)^{-2}\mathbf{s}_1 \tag{3.53}$$

Then we want:

$$\hat{\alpha}_{\lambda,\nu} < \frac{(\hat{\boldsymbol{\beta}}^*_{OLS})^T\hat{\boldsymbol{\beta}}^*_{OLS} - n^2\lambda^2\mathbf{s}_1^T(\mathbf{X}_1^T\mathbf{X}_1)^{-2}\mathbf{s}_1}{(\hat{\boldsymbol{\beta}}^*_{OLS})^T\hat{\boldsymbol{\beta}}^*_{OLS} - 2n\lambda(\hat{\boldsymbol{\beta}}^*_{OLS})^T(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1 + n^2\lambda^2\mathbf{s}_1^T(\mathbf{X}_1^T\mathbf{X}_1)^{-2}\mathbf{s}_1}. \tag{3.54}$$

Let:

$$A = (\hat{\boldsymbol{\beta}}^*_{OLS})^T \mathbf{X}_1^T \mathbf{X}_1 \hat{\boldsymbol{\beta}}^*_{OLS} - 2n\lambda \mathbf{s}_1^T \hat{\boldsymbol{\beta}}^*_{OLS} + n^2\lambda^2 \mathbf{s}_1^T (\mathbf{X}_1^T \mathbf{X}_1)^{-1} \mathbf{s}_1 \tag{3.55}$$

$$B = (\hat{\boldsymbol{\beta}}^*_{OLS})^T \hat{\boldsymbol{\beta}}^*_{OLS} - 2n\lambda (\hat{\boldsymbol{\beta}}^*_{OLS})^T (\mathbf{X}_1^T \mathbf{X}_1)^{-1} \mathbf{s}_1 + n^2\lambda^2 \mathbf{s}_1^T (\mathbf{X}_1^T \mathbf{X}_1)^{-2} \mathbf{s}_1. \tag{3.56}$$

Hence, $\hat{\alpha}_{\lambda,\nu} = \hat{\alpha}_{\lambda,0} - \frac{\nu}{A}$. Now $||\hat{\boldsymbol{\beta}}^*_{OLS} - \hat{\boldsymbol{\beta}}^1_\lambda||_2^2 > ||\hat{\boldsymbol{\beta}}^*_{OLS} - \hat{\alpha}_{\lambda,\nu}\hat{\boldsymbol{\beta}}^1_\lambda||_2^2$ whenever:

$$\nu > \frac{-A\Big((\hat{\boldsymbol{\beta}}^*_{OLS})^T \hat{\boldsymbol{\beta}}^*_{OLS} - n^2\lambda^2 \mathbf{s}_1^T (\mathbf{X}_1^T \mathbf{X}_1)^{-2} \mathbf{s}_1\Big)}{B} + A\hat{\alpha}_{\lambda,0} \tag{3.57}$$

and

$$\nu < A\hat{\alpha}_{\lambda,0} - A \ . \tag{3.58}$$

In order to establish whether this is possible, we must determine whether the lower bound for ν is smaller than the upper bound:

$$\frac{-A\Big((\hat{\boldsymbol{\beta}}^*_{OLS})^T \hat{\boldsymbol{\beta}}^*_{OLS} - n^2\lambda^2 \mathbf{s}_1^T (\mathbf{X}_1^T \mathbf{X}_1)^{-2} \mathbf{s}_1\Big)}{B} + A\hat{\alpha}_{\lambda,0} < A\hat{\alpha}_{\lambda,0} - A \tag{3.59}$$

$$\frac{-\Big((\hat{\boldsymbol{\beta}}^*_{OLS})^T \hat{\boldsymbol{\beta}}^*_{OLS} - n^2\lambda^2 \mathbf{s}_1^T (\mathbf{X}_1^T \mathbf{X}_1)^{-2} \mathbf{s}_1\Big)}{B} < -1 \tag{3.60}$$

$$(\hat{\boldsymbol{\beta}}^*_{OLS})^T \hat{\boldsymbol{\beta}}^*_{OLS} - n^2\lambda^2 \mathbf{s}_1^T (\mathbf{X}_1^T \mathbf{X}_1)^{-2} \mathbf{s}_1 > B \ . \tag{3.61}$$

Using the definition of B, we can determine that this inequality holds whenever:

$$
\begin{aligned}
-n^2\lambda^2\mathbf{s}_1^T(\mathbf{X}_1^T\mathbf{X}_1)^{-2}\mathbf{s}_1 &> -2n\lambda(\hat{\boldsymbol{\beta}}^*_{OLS})^T(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1 + n^2\lambda^2\mathbf{s}_1^T(\mathbf{X}_1^T\mathbf{X}_1)^{-2}\mathbf{s}_1 && (3.62)\\
n\lambda\mathbf{s}_1^T(\mathbf{X}_1^T\mathbf{X}_1)^{-2}\mathbf{s}_1 &< (\hat{\boldsymbol{\beta}}^*_{OLS})^T(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1 && (3.63)\\
n\lambda\mathbf{s}_1^T(\mathbf{X}_1^T\mathbf{X}_1)^{-2}\mathbf{s}_1 &< \mathbf{y}^T\mathbf{X}_1(\mathbf{X}_1^T\mathbf{X}_1)^{-2}\mathbf{s}_1 && (3.64)\\
0 &< \mathbf{y}^T\mathbf{X}_1(\mathbf{X}_1^T\mathbf{X}_1)^{-2}\mathbf{s}_1 - n\lambda\mathbf{s}_1^T(\mathbf{X}_1^T\mathbf{X}_1)^{-2}\mathbf{s}_1 && (3.65)\\
0 &< \hat{\boldsymbol{\beta}}^1_\lambda(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{s}_1\ . && (3.66)
\end{aligned}
$$

Because $(\mathbf{X}_1^T\mathbf{X}_1)^{-1}$ is symmetric and positive definite and we have assumed that $\hat{\boldsymbol{\beta}}_\lambda$ recovered the sign vector $\mathbf{s}_1$, this is always the case. Therefore we have the result. □

CHAPTER

4

SEQUENTIAL OPTIMIZATION FOR TUNING PARAMETER SELECTION

4.1 Motivation

As demonstrated in Chapters 2 and 3, penalized estimation involves one or more tuning parameters that control the impact of penalization. The final estimates reported are based on a single set of these tuning parameters. Often, we want the final set of estimates reported to be unbiased for the true parameters or to recover the true support of the underlying model. Ideally, we would like to efficiently explore the tuning parameter space so that we may consider all possible estimated models in our decision. Considering the entire tuning parameter space is feasible for some procedures, such as the Lasso for linear models, where the solution paths can be derived efficiently (Efron et al. 2004). There are many penalized estimators, some of which were described in Chapter 1, for which fully exploring the tuning parameter space is too computationally intensive to be feasible. As such, the estimates themselves can suffer from improper tuning parameter selection. This chapter presents a sequential method of efficient exploration of complex tuning parameter spaces for such estimators.

K-fold CV (Stone 1974; Allen 1974; Geisser 1975) has also been a major focus of Chapters 2 and 3 because it is one of the most widely used tuning parameter selection methods. K-fold CV breaks up the data set into K folds of size n_k and allows each of the individual folds to be used as a holdout sample. The tuning parameters that minimize the APE across the K folds are often chosen as the final parameters. A 1SE Rule has been shown in practice to be useful for choosing sparser models, though implementing a 1SE Rule for multidimensional tuning parameters can be challenging.

The computational expense of K-fold CV can leave practitioners with two undesirable options: explore a dense grid of tuning parameters, which can be prohibitively time consuming, or inspect a sparse set of λ values and risk missing the minimum APE, or not adequately defining the 1SE region around the minimum. This dilemma was faced by Stallrich et al. (2020) in their implementation of the SAFE procedure for the EMG data application discussed in previous chapters. The computational expense of adaptive penalized estimation of a functional linear model lead to K-fold CV across a relatively coarse grid of tuning parameters. As a result, it is not clear whether sufficient exploration of the tuning parameter space has yet been achieved.

In this chapter, a new approach to K-fold CV is proposed, wherein tuning parameters are sequentially sampled first to minimize the APE and then to explore the 1SE area around the minimum. The tuning parameter selection problem is treated like a computer experiment (Santner et al. 2003), starting with a sparse, "space-filling" set of tuning parameters, and then interpolating the APE function across a dense grid. Sequential optimization is performed to identify (1) the λ that results in the minimum APE, and (2) the λ values that lie within one standard error of the minimum. Although this approach has some precedence in the Machine Learning community (Bergstra et al. 2011; Koehrsen 2018; Brownlee 2020; Cho et al. 2020), these methods were developed to find the minimum APE, and do not address the potential need for a 1SE Rule. The use of a 1SE Rule complicates the approximation of the APE curve because it introduces an incentive to perform more exploration in the area near the minimum. The goals of exploitation and exploration are therefore more interwoven than for the more straightforward minimum APE approach.

Consider the toy example from Chapter 2. Recall that the data is generated from a linear model with $n = p = 100$, errors drawn from a standard Normal distribution, and 5 active coefficients ranging in magnitude from 1.81 to 2.51. The APE curve from 5-fold CV for this data, across a grid of 500 λ values between e^{-20} and e^{3}, is shown in Figure 4.1. A few important qualities may be observed in Figure 4.1 that seem to generally apply to APE curves of this nature. First, the curve is reasonably well behaved. It is smooth and continuous and has a single global minimum. The points within one standard error of the minimum are all close to the minimum, so once the minimum APE λ has been found it would be reasonable to restrict the search to points near the minimum. There are also areas in the curve, for very small λ and very large λ, where the curve is either flat or nearly flat. Finally, the APE curve is deterministic for a fixed set of folds. In Figure 4.2 the curve is approximated using a Gaussian Process on only 12 of the λ values pictured in Figure 4.1. When the Expected Improvement is calculated for all 500 λ values observed in 4.1, the next sampling point chosen is very close to the minimum APE observed from

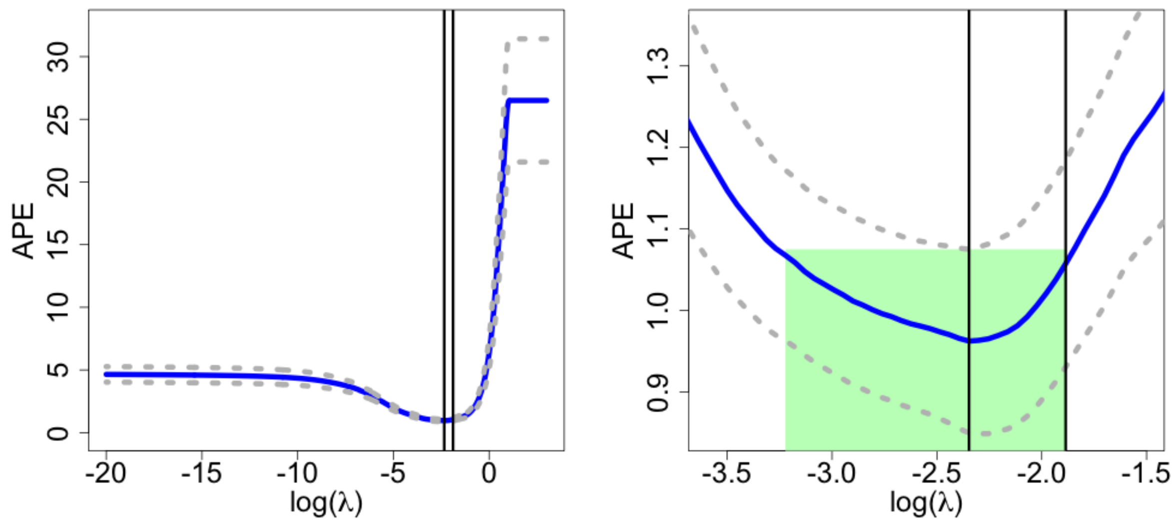

Figure 4.1: APE curve (blue line) for the generated data set in Chapter 2. The left plot shows the full curve, the right plot shows the curve zoomed in on the minimum. The APE plus and minus one standard error are represented by the grey dashed lines. The left-most vertical line is the λ giving the minimum APE and the right-most vertical line is the λ chosen by a 1SE Rule. The green shaded area on the rightmost plot signifies all λs resulting in an APE less than the minimum plus one standard error.

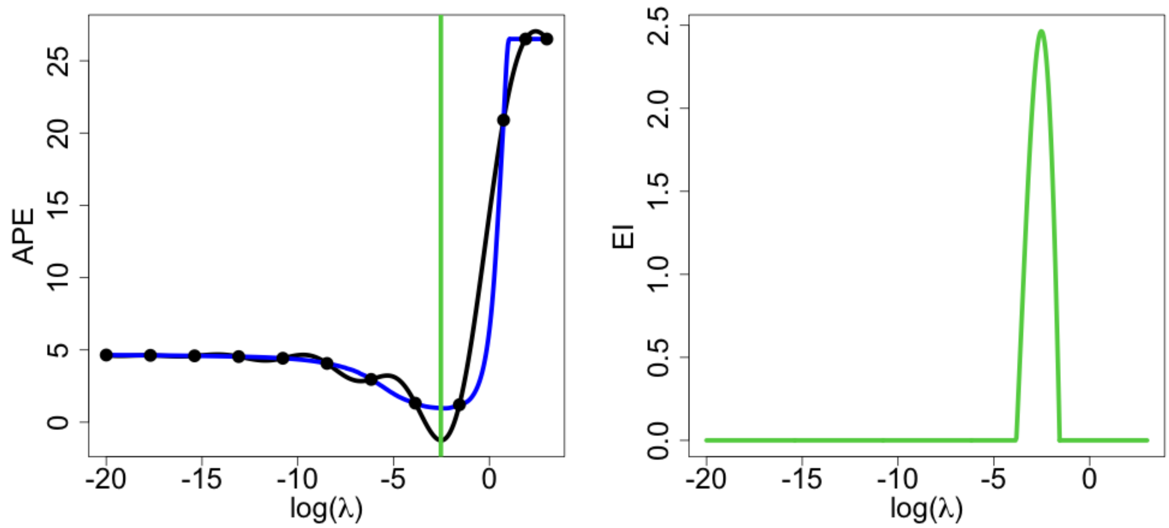

Figure 4.2: APE curve (blue line) for the generated data set in Chapter 2 and the Gaussian Process approximation (black line) from 12 λ values (points). The right plot shows the Expected Improvement for all 500 λ values observed in Figure 4.1. The green vertical line indicates the next sampling point, as chosen by the Expected Improvement.

the dense grid of λ values, suggesting that the approximation has the capability to find the minimum APE with far less information than 500 sampling points.

This simple example illustrates the first step in the sequential optimization procedure described in this chapter. The process would be repeated until some metric of convergence was reached, at which point the APE curve will have been satisfactorily approximated, without carrying out 5-fold Cross Validation on all 500 λ values. The next stage of the process is to find the boundaries of the area of the APE curve that is less than 1SE from the minimum APE. The sparsest model that meets that criterion can be selected as the optimal. In the case of a one-dimensional tuning parameter, a larger λ is always associated with a sparser model and the boundary will be simple to define as the largest λ within one standard error of the minimum. To find the true boundary then, we simply have to sample points around the estimated boundary until we find two sequential λs in the dense grid where the smaller λ is less than the minimum plus one standard error and the larger is greater than it.

This example is explored here as a simple proof of concept. In practice, computation of the APE across a dense set of λ values for the Lasso estimates of coefficients in the linear model is extremely efficient. The LARS algorithm can be used to almost instantaneously find the complete solution path so K-fold CV is fast. The practical use of the proposed sequential optimization procedure is intended for more computationally expensive problems. Computational expense for penalized estimators can come from two sources: first, reaching penalized estimates themselves can be time consuming and second, fully exploring a multi-dimensional tuning parameter space is more expensive than doing so for a single tuning parameter. More complex penalties than the traditional Lasso often suffer from both sources of computational expense.

Additionally, in a multi-dimensional tuning parameter space, the 1SE Rule is less intuitive because the “less sparse direction" can be difficult to determine. Consider the example in Figure 4.3. This plot shows the APE of the same data from Figure 4.1, this time exploring the two-dimensional tuning parameter space induced by the Relaxed Lasso. The APE is calculated for 500 λ_1 values and 500 λ_2 values. The right hand side of the plot only displays those APE values that are less than the minimum plus one standard error. These sets of λ_1 and λ_2 would all be candidates for the optimal model given a 1SE Rule. If we employ the same strategy for implementing a 1SE Rule as used in Chapter 2, the optimal $\hat{\boldsymbol{\beta}}_{\lambda_1,\lambda_2}$ would be the vector from this set that minimizes $\|\hat{\boldsymbol{\beta}}_{\lambda_1,\lambda_2}\|_1$.

Once the sequential approach has interpolated the left side of Figure 4.3, the challenge is in determining how to adequately explore the 1SE region around the minimum to ensure that the best possible model is selected without having to calculate the APE for all 19988 points pictured on the right. This chapter proposes the use of a space-filling design of the sampling points estimated to be within one standard error of the minimum

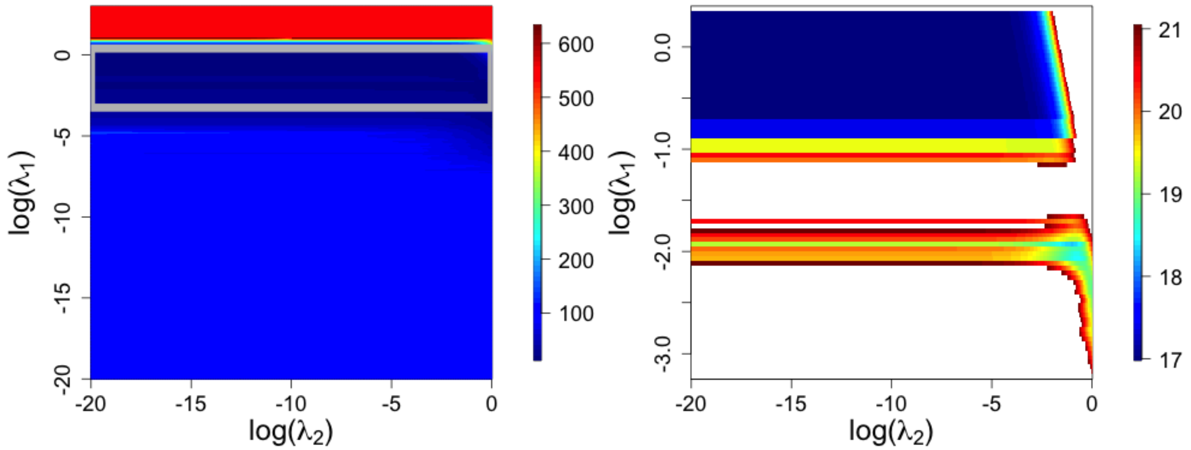

Figure 4.3: APE for the Relaxed Lasso estimates of the data from Figure 4.1. On the left are the APE values for all λ_1 and λ_2. The leftmost plot shows all APE values. The grey box indicates the window of tuning parameters shown on the right. The right plot, with colors re-scaled, shows only the values that are less than the minimum APE plus one standard error.

until the number of desired candidates have been observed within that area. Ideally, focus would be given to points along the border of the 1SE area.

Section 4.2 discusses the various methods of sequential design that can be used to approximate and refine the estimated APE function. Section 4.3 describes the methodology and provides algorithms for its implementation. The results of two simulation studies are discussed in Section 4.4 and the effectiveness of the new procedure is reviewed. In Section 4.4 the new approach is applied to the EMG data and its results and efficiency are compared to the results of the original SAFE application in Stallrich et al. (2020). Finally, directions for future study on this topic are presented in Section 4.6.

4.2 Background

There are three components to the approach described in Section 4.3 that will be discussed in this brief literature review. First, a discussion of Gaussian process regression is required to form an approximation of the APE curve. Second, the principles of Bayesian optimization will be used to carry out selection of new λ, or sampling points, to both refine the approximation of the APE and to minimize it. Finally, a smaller but still important component of the sequential APE approach is the use of space-filling designs to explore the estimated 1SE area around the minimum APE. A few of these designs are detailed here.

4.2.1 Gaussian Process Regression

The method of approximating the APE function proposed here is through Gaussian process regression. A Gaussian process (GP) is a—possibly infinite—collection of random variables wherein every finite subset of those random variables, indexed by $\mathcal{X}$, follows multivariate Normal distribution with mean function $\mu : \mathcal{X} \to \mathbb{R}$ and a positive definite covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Commonly in sequential optimization, the mean function is assumed to be a constant, μ_0, and $k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp(\frac{-\|\mathbf{x}-\mathbf{x}'\|_2^2}{\theta})$ where θ is the lengthscale hyperparameter (Shahriari et al. 2016). In the application of GP regression given in this chapter, the set $\mathcal{X}$ represents the tuning parameter subspace, Λ, and the APE curve is a realization of the GP.

Consider a finite collection of t points $\mathbf{X}_t$ where each row of $\mathbf{X}_t$, $\mathbf{x} \in \mathcal{X}$. The vector $\mathbf{y}_t = [y_1, ..., y_t]^T$ represents t noiseless realizations of the system under the model $y_i = f(\mathbf{x}_i)$. These sampling points provide useful information about the GP that can be used to define the conditional distribution of the system, given $\{\mathbf{y}_t, \mathbf{X}_t\}$. The conditional mean of $f(\mathbf{x})$ is therefore:

$$\mu_t(\mathbf{x}) = \mu_0 + k_t(\mathbf{x})\mathbf{K}_t^{-1}(\mathbf{y}_t - \mu_0 \mathbf{1}_t) \tag{4.1}$$

where $\mathbf{K}_t$ is the covariance matrix based on evaluating $k(\mathbf{x}, \mathbf{x}')$ at all pairs of points in $\mathbf{X}_t$ and $\mathbf{k}_t(\mathbf{x})$ is the $1 \times t$ matrix comprised of $k(\mathbf{x}, \mathbf{x}_i)$ for all $\mathbf{x}_i$ in $\mathbf{X}_t$. The conditional variance is:

$$\sigma_t^2(\mathbf{x}) = \sigma^2 - \mathbf{k}_t(\mathbf{x})^T \mathbf{K}_t^{-1} \mathbf{k}_t(\mathbf{x}) \ . \tag{4.2}$$

The parameters μ_0 and θ will need to be estimated in order to define the conditional distribution. In this chapter, the estimation will be carried out via maximum likelihood. Making predictions about unobserved points in $\mathcal{X}$ is known as kriging and can be done simply by estimating the mean, $\mu(\mathbf{x})$ across all points of interest in $\mathcal{X}$ (Gramacy 2020; Shahriari et al. 2016).

4.2.2 Bayesian Optimization and Acquisition Functions

Bayesian optimization is a sequential approach to optimizing f when it is a noisy and/or expensive black-box function. The process begins with a statistical model of f that represents the prior. While the model is not necessarily a GP, it is often one and the GP model will be assumed for this chapter. The statistical model is then used to evaluate an acquisition function, $a(\mathbf{x})$, that assesses how useful a given sampling point, $\mathbf{x} \in \mathcal{X}$, will be to optimize f. The point that returns the highest value of the acquisition function is suggested as the next sampling point (Shahriari et al. 2016).

Acquisition functions are designed to balance two, sometimes competing, goals: exploration and exploitation. The aim of exploration is to fully inspect the space so as to improve the approximation; exploitation means focusing on areas that have a high probability of containing the global maximum or minimum, depending on the optimization goal. When a new sampling point is added, $\mu_t(\mathbf{x})$ and $\sigma_t^2(\mathbf{x})$ must be updated because the conditional mean and variance are likely to change as sampling points are added (Berk et al. 2020).

Two standard acquisition functions are described here: Expected Improvement and the Gaussian Process Upper Confidence Bound, though many more exist and can be implemented in the algorithm presented in Section 4.3. Optimization problems often involve maximizing some function so many acquisition functions are written for that purpose. The goal of K-fold CV, however, is to minimize the APE function—or to find a reasonable window around the minimum APE—so there are a few small alterations that need to be made to the acquisition functions as they are generally presented.

Expected Improvement (EI) (Mockus et al. 1978) is an acquisition function quantifying the amount of improvement a new value, v, can be expected to have on the current approximated optimum, τ. For minimization, it is defined:

$$a_{EI}(v) = E[(\tau - v)\mathbb{I}(v < \tau)] \ . \tag{4.3}$$

Under the Gaussian Process framework:

$$a_{EI}(\mathbf{x}) = (\tau - \mu_t(\mathbf{x}))\Phi\Big(\frac{\tau - \mu_t(\mathbf{x})}{\sigma_t(\mathbf{x})}\Big) + \sigma_t(\mathbf{x})\phi\Big(\frac{\tau - \mu_t(\mathbf{x})}{\sigma_t(\mathbf{x})}\Big) \ , \tag{4.4}$$

where Φ is the standard Normal cumulative density function and ϕ is the standard Normal probability density function. EI has been shown to perform well in experiments (Osborne 2010) and Bull (2011) proved, under some assumptions, the convergence rate of EI to the optimum.

Another acquisition function of interest is the Gaussian Process Upper Confidence Bound (UCB), an optimistic—rather than improvement-based—policy. The acquisition function for UCB (Srinivas et al. 2012) is:

$$a_{UCB}(\mathbf{x}) = -\mu_t(\mathbf{x}) + \sqrt{\beta_t}\sigma_t(\mathbf{x}) \ . \tag{4.5}$$

where β_t is a hyperparameter chosen by the practitioner. Increasing β_t forces the function to favor high-variance points, whereas decreasing it will cater to low mean points and prioritize finding the minimum. Theoretical results give an expression for β_t that ensures that the algorithm has sub-linear regret. In practice, however, the UCB acquisition function has the tendency to encourage exploration over exploitation, which can lead to the selection of many unnecessary additional points (Berk et al. 2020).

The solution to this over-exploration problem is given in Berk et al. (2020). They propose a Randomized Gaussian Process UCB approach. The acquisition function remains the same in their algorithm, but β_t is now drawn from the Gamma distribution with scale θ and shape

$$\kappa_t = \frac{\log\big((2\pi)^{-1/2}(t^2+1)\big)}{\log\big(1+\zeta/2\big)} \ . \tag{4.6}$$

In this expression, ζ is a hyperparameter chosen by the practitioner. Berk et al. (2020) suggest that $\zeta = 8$ has been shown in simulations to favor exploration, $\zeta = 0.5$ favors exploitation, and $\zeta = 1$ is a good balance of the two; they offer no theoretical justifications for these claims, however.

These sequential optimization procedures are all focused on minimizing the function. For the K-fold CV procedures of interest here, minimization is an important part of the process but the main goal is classifying tuning parameter points as within 1SE of the minimum and over 1SE from the minimum. This classification goal is shared in the work of Inatsu et al. (2020). Their approach, called the Level Sets classification method, sequentially adds points for the purpose of classifying values as above or below some threshold (Inatsu et al. 2020). The authors present algorithms for a predefined threshold and an implicit threshold that is defined as some proportion of the optimal value. Integrating this method into the approach proposed in Section 4.3 would present a few challenges. First of all, a predefined threshold in the APE application is impossible, but the implicit threshold algorithm given in Inatsu et al. (2020) is also inapplicable. The threshold needed in the 1SE application comes from the minimum plus another quantity determined by the function itself, the standard error of the minimum. That is, it is not merely necessary to estimate the minimum of the APE, but to observe it as well in order to determine the necessary threshold. This wrinkle adds an additional layer of uncertainty that is not addressed by the Level Sets algorithms. Because of these challenges, this chapter assumes the use of more traditional acquisition functions, but acknowledges the use of Level Sets to define the desired boundary as a subject of future study.

4.2.3 Space-filling Designs

The implementation of a sequential sampling method to find the 1SE bound of an APE function will require a space-filling design algorithm. This is an interesting design problem in that it involves a finite sample of potential new sampling points from which a small set of space-filling points must be selected. The APE curve will then be observed on that set.

There are a few approaches to carry this out. First, consider Spatial Coverage Designs (Royle and Nychka 1998). This technique performs optimization with respect to the geometric coverage of a set of data points. Let C be the candidate set of points and D_n be a subset of n design points from C. The distance between any given location $\mathbf{x}$ and the design D_n is defined:

$$d_n(\mathbf{x}, D_n) = \Big(\sum_{\mathbf{y} \in D_n} \|\mathbf{x} - \mathbf{y}\|^p \Big)^{\frac{1}{p}} \tag{4.7}$$

where $p < 0$. A design, D_n^*, is optimal if, over all possible designs in the candidate set, it minimizes:

$$C_{p,q}(C, D_n) = \Big(\sum_{\mathbf{x} \in C} d_p(\mathbf{x}, D_n)^q \Big)^{\frac{1}{q}} \ . \tag{4.8}$$

Increasing the constant q increases the importance of the distance of the design to poorly covered locations. The point-swapping algorithm presented in Royle and Nychka (1998)

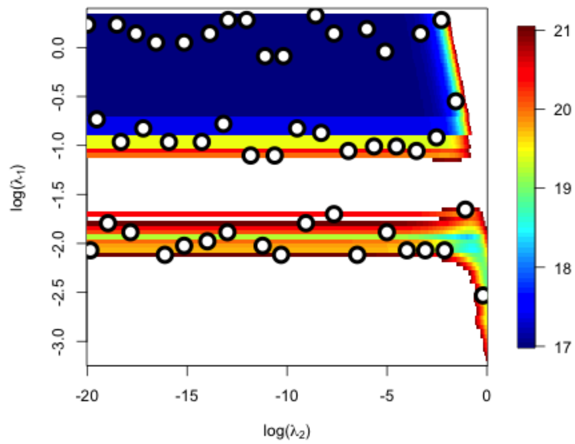

Figure 4.4: White points represent a Maximin space-filling design of the 19988 combinations of λ_1 and λ_2 that are within 1SE of the minimum APE shown in Figure 4.3.

has been implemented in S-PLUS and Stata (Bia and Kerm 2014).

Another approach is to use the Maximin criterion, which seeks to maximize the minimum pairwise distance between design points:

$$\Delta(D_n) = \min_{\mathbf{x}_i, \mathbf{x}_j \in D_n} \left\{ \|\mathbf{x}_i - \mathbf{x}_j\| : j \neq i \right\} \tag{4.9}$$

where $\|\cdot\|$ is generally the Euclidean distance but can be any distance metric (Tan 2013). The `R` package `maximin` can be used to efficiently implement this approach. As an illustration, consider the 1SE area found for the Relaxed Lasso example in Figure 4.3. A Maximin space-filling design sampling 50 points within 1SE of the minimum is shown in Figure 4.4. Notice that points along the borders of the 1SE area seem to be favored in this design approach.

4.3 Gaussian Process Sequential APE

Consider some APE function from K-fold Cross Validation with fixed folds. This function is deterministic, in that sampling one λ multiple times should lead to the same result. Furthermore, empirical evidence indicates that APE curves tend to be fairly well-behaved, meaning it is reasonable to approximate them using GP regression.

To begin, the APE must be evaluated for a coarse grid of tuning parameters, Λ_0 and dense grid from the tuning parameter space, Λ, must be defined. The sampling points in Λ_0 should be space-filling and encompass the full range of the tuning parameters in Λ. Denote the initial APE observations $APE(\lambda_0)$ for all $\lambda_0 \in \Lambda_0$. The mean and standard deviation of the GP are estimated for all λ in Λ. The predicted value of the APE for a point $\lambda' \in \Lambda$ is denoted $\widehat{APE}(\lambda')$. Let $a : \Lambda \to \mathbb{R}$ be whichever acquisition function is preferred. In the simulations and application presented in Sections 4.4 and 4.5 respectively, the acquisition function implemented is Expected Improvement (EI), but note that any acquisition function could be used in its place. Then, calculate $a(\lambda)$ for all $\lambda \in \Lambda$ and maximize it to find the next sampling point, λ_t. Find $APE(\lambda_t)$ and re-estimate $\widehat{APE}(\Lambda)$ with the information from this newest sampling point.

The sequential sampling process must be repeated until a metric of convergence is satisfied or some maximum number of iterations has been reached. Because minimization of the APE function is the ultimate goal for this application, we recommend iterating until the estimated minimum converges. This first stage of the approach is detailed in Algorithm 2 and its purpose is to obtain a reasonable approximation of the APE curve and, crucially, identify the minimum.

The goal of the next stage is to locate the 1SE region around the minimum APE. If the minimum APE is desired rather than a 1SE Rule, this step is unnecessary and the addition of new sampling points can be halted after convergence is reached in the

Algorithm 2 Sequential APE

Initialize Λ_0, a sparse sample set of cardinality T_0 and calculate $APE(\lambda)$ for all $\lambda \in \Lambda_0$. Find $\widehat{APE}_0(\Lambda)$. Choose $\epsilon > 0$, $T_{\max} > T_0$, and acquisition function $a : \Lambda \to \mathbb{R}$, with an interest in minimization. Let Λ be the finite sample set of possible λ we are interested in approximating. Set $t = 0$.

while $t + T_0 \leq T_{\max}$ or $\delta > \epsilon$ **do**

$t \leftarrow t + 1$

$\lambda_t \leftarrow \arg\max_{\lambda \in D} a(\lambda)$

Calculate $APE(\lambda_t)$

$\Lambda_t \leftarrow \Lambda_{t-1} \cup \lambda_t$

Compute $\widehat{APE}_t(\lambda)$, for all $\lambda \in \Lambda_t$

$\delta \leftarrow |\min_{\lambda \in \Lambda} \widehat{APE}_t(\lambda) - \min_{\lambda \in D} \widehat{APE}_{t-1}(\lambda)|$

end while

$\Lambda_T \leftarrow \Lambda_t$

$\lambda_{min} \leftarrow \arg\min_{\lambda \in \Lambda_T} \widehat{APE}_t(\lambda)$

$T \leftarrow \text{Card}(\Lambda_T)$ and $h \leftarrow APE(\lambda_{min}) + SE(\lambda_{min})$

previous stage. If, however, a 1SE Rule is needed, C, the desired number of λ within one standard error of the minimum APE, should be pre-set. It is from among the models arising from the C candidates that the optimal will be chosen by minimizing a model complexity criterion. If C was equal to 1, for example, the minimum could be the only model found with an APE less than one standard error plus the minimum. The larger C is, the more candidate models there will be to choose from to select the optimal, but C should not be so large as to cancel out the computational advantage of the sequential sampling approach.

Use the minimum APE λ found in the first stage to find h, the minimum APE plus one standard error. Then find all $\lambda \in \Lambda$ such that $\widehat{APE}(\lambda) < h$. From among these predictions, some will have already been observed as less than the cutoff value. These sampling points can be placed immediately in the set of candidates for the optimal model. As long as the cardinality of the candidate set is less than C, choose a space-filling set of the left-over points whose predictions were less than h of size $\rho \times C$ where $\rho \in (0, 1)$. The purpose of ρ is to ensure that the 1SE region around the minimum is explored while protecting against potential changes to the approximation. If h is estimated to be much smaller than necessary and $\rho = 1$, all candidate models will be too close to the minimum and potentially not sparse enough. The implementation of a space-filling design will explore the 1SE region, placing special attention on the boundary.

Consider the example given in Figure 4.5. The black points represent 500 discrete two-dimensional points. Shown in purple are 20 points sampled using the Maximin space-filling algorithm. Note that many of the sampled points are along the borders of the general shape of the area. This is the kind of behavior we hope to see in the search for λ within one standard error of the minimum because whatever measurement of model complexity we use to decide on a final set of tuning parameters, the optimal λ is likely to be along the edges of this space. The space-filling design is not constricted to only border points, however, which is also important because we want to explore the space around the approximated minimum in order to be sure that the approximated minimum is exactly or very close to the true minimum.

The APE function is then evaluated at the additional sampling points suggested by the space filling algorithm. With these additional points, $\widehat{APE}(\lambda)$ is re-estimated for all $\lambda \in \Lambda$. This process continues until C candidate points within 1SE of the minimum have been identified. This refinement stage is detailed in Algorithm 3. The model complexity of the candidates is evaluated and the optimal set of estimates chosen are those that minimize that complexity. Ideally, the sequential process can be carried out more quickly than observing the true APE curve across Λ.

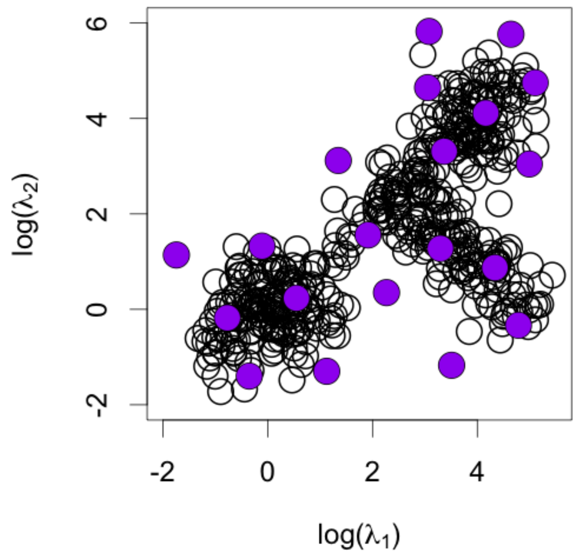

Figure 4.5: Maximin space-filling sampling from a discrete set of points. Purple points show the sample of size 20.

Algorithm 3 Refinement for 1SE Rule

Initialize $i = 1$. Use Algorithm 2 to find Λ_T and h. Let C be the total number of candidate models (models within one standard error of the minimum) desired. Let $\Lambda^* = \{\lambda \in \Lambda_T : APE(\lambda) \leq h\}$ with c equal to the cardinality of Λ^*, and let $\Lambda_{pc} = \{\lambda \in \Lambda : \widehat{APE}_T(\lambda) \leq h\}$, with d equal to the cardinality of Λ_{pc}. For some $\rho \in (0, 1)$, let $n_{pc} = \rho \times C$.

$\Lambda_{T,0} \leftarrow \Lambda_T$

while $c < C$ and $d > 0$ **do**

  Choose Λ^i_{pc}, a representative sample from Λ_{pc} of size n_{pc}

  Evaluate $APE(\lambda)$ and $SE(\lambda)$ for all $\lambda \in \Lambda^i_{pc}$

  $\Lambda_{T,i} = \Lambda_{T,i-1} \cup \Lambda^i_{pc}$

  if $\min\limits_{\lambda \in \Lambda^i_{pc}} APE(\boldsymbol{\lambda}) < \min\limits_{\lambda \in \Lambda_{T,i-1}} APE(\lambda)$ **then**

    $\lambda_{min} \leftarrow \arg\min\limits_{\lambda \in \Lambda^i_{pc}} APE(\lambda)$

    $h \leftarrow APE(\lambda_{min}) + SE(\lambda_{min})$

  end if

  $\Lambda^* = \{\lambda \in \Lambda_{T,i} : APE(\lambda) \leq h\}$

  $c \leftarrow \mathrm{Card}(\Lambda^*)$

  Calculate $\widehat{APE}(\lambda)$ for all $\lambda \in \Lambda$, having sampled $\Lambda_{T,i}$

  $\Lambda_{pc} = \{\lambda \in \Lambda : \widehat{APE}(\lambda) \leq h\}$, with d equal to the cardinality of Λ_{pc}

  $i \leftarrow i + 1$

end while

4.4 Numerical Results

The following section describes two simulation studies performed to evaluate the effectiveness of the approach, as well as its efficiency. The settings chosen are both computationally expensive: the Relaxed Lasso for Poisson regression and a functional setting for the sparsity-smoothness penalty on the functional linear model. The illustration in Section 4.1 demonstrated the use of the sequential approach for the linear model with a Lasso penalty. This was used as a proof of concept to motivate the development of an algorithm that could be more generally used. The sequential approach is usually unnecessary for the linear model—especially for the Lasso penalty—because computation speeds for an extremely dense set of tuning parameters is inexpensive. This section instead evaluates the method's usefulness for two scenarios where computation time can be severely hindered by the evaluation of many tuning parameters.

4.4.1 Relaxed Lasso Poisson Regression

In this section, focus is given to how the approximated results mirror results from observing the full APE function. As noted in Chapter 3, the 1SE rule as implemented here is not always reliable for choosing the best submodel or finding the lowest bias coefficients in GLMs. For the purposes of this discussion, these concerns are set aside to determine whether the sequential APE method can recover the 1SE estimates, leaving further consideration of how to implement the 1SE Rule in the most effective way for future research.

To evaluate this approach, consider the results of the simulation study given in Table 4.1. To generate these results, data with sample size $n = 100$ were drawn from a Poisson regression model with independent predictors, as in Section 3.4. Coefficients selected such that $\sum_{i=1}^{p^*} = \beta_j^2 = 1$ with randomly assigned signs and the intercept was randomly chosen from $\{-1, 0\}$ (Kibria et al. 2015). Once again, the elements of $\mathbf{X}$ were independently generated from a Normal distribution with mean 0 and standard deviation 0.5. 500 data sets were randomly generated.

To evaluate the performance of the sequential method, the APE curve was observed across a grid of 200 λ_1 and 200 λ_2 values and the optimal model under a 1SE Rule, $\hat{\boldsymbol{\beta}}_{GS}$ was determined; h_{GS} is the minimum APE plus one standard error. The coefficient estimate vector from the more efficient approximated APE approach is represented by $\hat{\boldsymbol{\beta}}_A$ and h_A gives the minimum APE plus one standard error from the approximated function. APE Diff gives the sum of the element-wise squared difference between the predicted and observed APE curves divided by the number of elements. HD is the Hamming distance between $\hat{\beta}_A$ and $\hat{\beta}_{GS}$, excluding the intercept. Finally, Sampling Points gives the number of combinations of λ_1 and λ_2 evaluated for the sequential APE approach. Note that the number of sampling points for the Grid Search is 200×200.

We begin by reviewing the particulars from a single data set generated in this fashion. Consider Figure 4.6. APE (Initial) gives the APE curve for the initial values of λ_1 and λ_2, covering the full range of data. APE (True) gives the APE function realized across all 200×200 sets of tuning parameters. This is the function that the new method is attempting to approximate. APE Approximation (Stage 1) gives the full curve approximated after convergence is reached in the first stage. Sampling points are marked using white points and the large black dot gives the approximated minimum. APE Approximation (Stage 2) gives the GP approximation after the refinement stage, where $C = 50$ candidates within one standard error of the minimum have been found. These candidate points are marked with a small black ex. The minimum plus one standard error estimated from the GP approximation in APE Approximation (Stage 2) is 19.43, compared to that of the APE (True), which is 18.90; the number of points sampled to achieve the approximation was 74, compared to the full observation of the function at 200×200. Although there is some discrepancy in the values, the approximation approach has sampled roughly 2% of the points evaluated in the Grid Search. The final models chosen by each of the approaches have the same number of active coefficients and the $\|\hat{\boldsymbol{\beta}}_{GS} - \hat{\boldsymbol{\beta}}_A\|_2$ is 0.457. That is, they do not return identical models but extremely similar models.

This demonstration only describes a single data set, so consider the simulation study of 500 replications of data under the same settings, found in Tables 4.1 and 4.2. For these results, $C = \{10, 20, 50, 100\}$, $\rho = \{0.3, 0.5\}$, and $n = \{100, 500\}$. It is clear that the sequential method returns coefficient vectors that are extremely close to the estimates returned from the far more expensive procedure. Naturally, as C increases, the computational expense of the procedure also increases and the approximation tends to perform marginally better. Even the smallest C, however, gives results that are extremely close to the Grid Search results. There do not appear to be considerable differences between $\rho = 0.3$ and $\rho = 0.5$ except that $\rho = 0.5$ leads to more tuning parameter points being sampled. These results offer a glimpse of how effective the method can be when used for multiple tuning parameters.

Table 4.1: Average results from 500 replications of the Sequential APE process for the Relaxed Lasso penalty on a Poisson regression model with $n = 100$. Standard errors given in parentheses.

	C	$\|\hat{\boldsymbol{\beta}}_{GS} - \hat{\boldsymbol{\beta}}_A\|_2$	$\|h_{GS} - h_A\|$	APE Diff	HD	Sampling Points
$\rho = 0.3$	10	0.366 (0.009)	0.77 (0.041)	0.03 (0.001)	2.455 (0.093)	45.591 (0.872)
	20	0.356 (0.009)	0.767 (0.041)	0.03 (0.001)	2.41 (0.093)	48.07 (0.896)
	50	0.268 (0.009)	0.706 (0.039)	0.032 (0.001)	1.533 (0.069)	76.964 (0.835)
	100	0.219 (0.008)	0.694 (0.039)	0.035 (0.001)	1.331 (0.062)	135.908 (1.169)
$\rho = 0.5$	10	0.364 (0.009)	0.767 (0.041)	0.03 (0.001)	2.444 (0.093)	45.856 (0.884)
	20	0.353 (0.009)	0.753 (0.041)	0.031 (0.001)	2.371 (0.093)	48.689 (0.907)
	50	0.278 (0.009)	0.701 (0.039)	0.033 (0.001)	1.56 (0.071)	81.26 (0.895)
	100	0.222 (0.008)	0.694 (0.039)	0.035 (0.001)	1.394 (0.061)	145.232 (1.294)

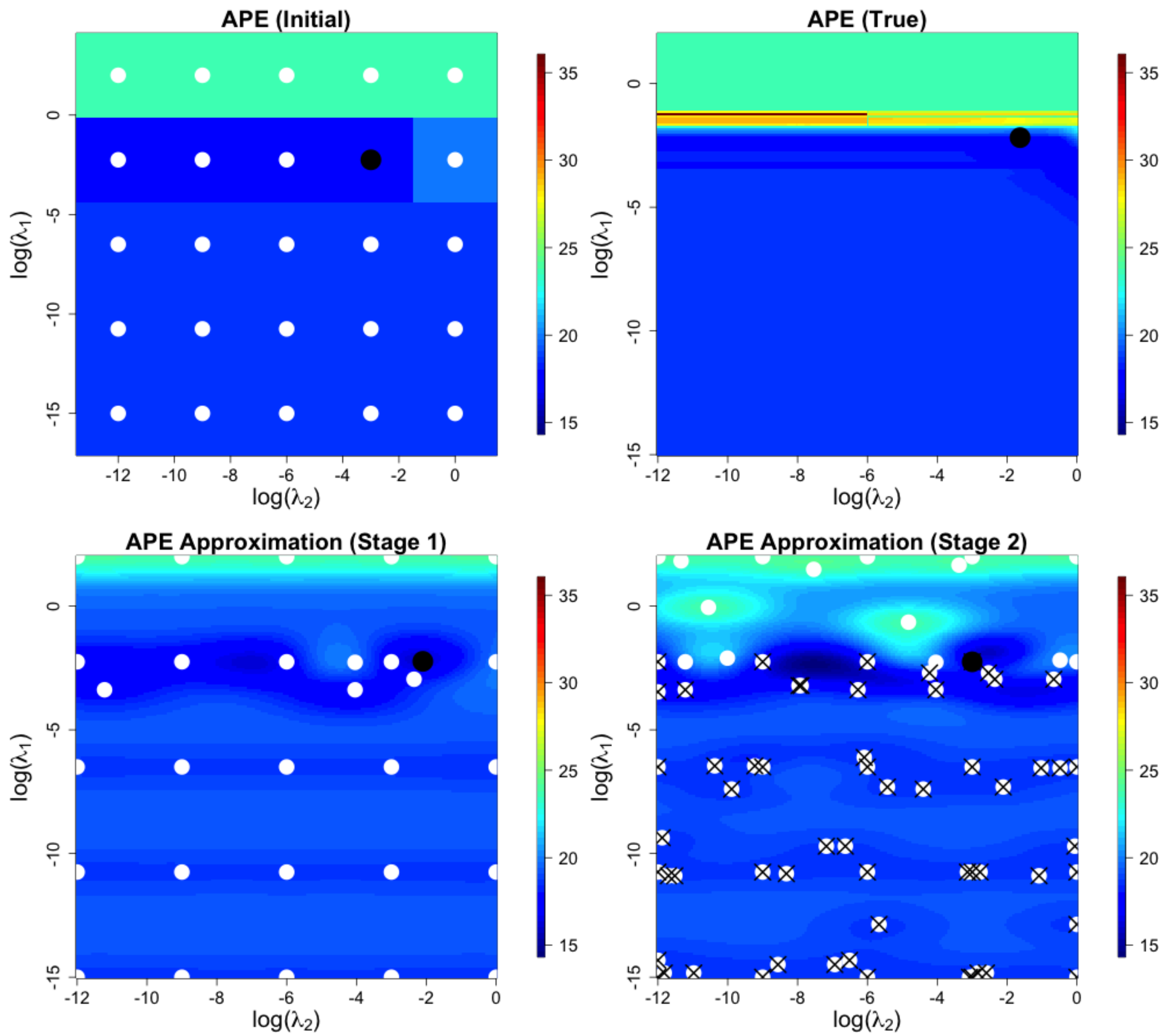

Figure 4.6: True APE function values (top row) and approximated APE function values (bottom row). Sampling points marked with white dots, the minimum denoted with a large black dot, and for the 1SE Search stage of the Approximation method, black exes mark all candidate models.

Table 4.2: Average results from 500 replications of the Approximate APE process for the Relaxed Lasso penalty on a Poisson regression model with $n = 500$. Standard errors given in parentheses.

	C	$\|\hat{\boldsymbol{\beta}}_{GS} - \hat{\boldsymbol{\beta}}_A\|_2$	$\lvert h_{GS} - h_A \rvert$	APE Diff	HD	Sampling Points
$\rho = 0.3$	10	0.199 (0.003)	0.567 (0.032)	0.172 (0.004)	1.574 (0.073)	70.851 (1.503)
	20	0.199 (0.003)	0.567 (0.032)	0.172 (0.004)	1.57 (0.073)	71.002 (1.499)
	50	0.163 (0.003)	0.555 (0.031)	0.176 (0.005)	0.982 (0.06)	84.61 (1.026)
	100	0.131 (0.003)	0.549 (0.031)	0.189 (0.005)	0.725 (0.048)	134.194 (0.67)
$\rho = 0.5$	10	0.199 (0.003)	0.567 (0.032)	0.172 (0.004)	1.572 (0.073)	70.879 (1.502)
	20	0.199 (0.003)	0.568 (0.032)	0.172 (0.004)	1.571 (0.074)	71.055 (1.5)
	50	0.17 (0.003)	0.552 (0.031)	0.175 (0.005)	1.101 (0.064)	85.659 (1.018)
	100	0.136 (0.003)	0.549 (0.031)	0.19 (0.005)	0.774 (0.048)	145.992 (1.081)

4.4.2 Functional Linear Regression

Another setting whereby this approach can be useful in reducing the number of calculations is for functional linear regression. The functional linear model used to generate this data was:

$$y_i = \sum_{j=1}^{15} \int_1^{300} \beta_j(t) \mathbf{X}_{ij}(t) dt + \epsilon_i \tag{4.10}$$

where all ϵ_i were taken from independent Normal distributions with mean 0 and variance determined by a fixed signal-to-noise ratio (SNR) of 5. The SNR, was defined $SNR = s^2_{\hat{y}}/\sigma^2$ where $s^2_{\hat{y}}$ is the sample variance of the response without error, as in McLean et al. (2014). Covariate functions $\beta_1(t)$ and $\beta_2(t)$ are the sinusoidal functions pictured in Figure 4.7, all other $\beta_j(t)$ were set to zero, and $n = 100$. The covariate functions were generated using

$$\mathbf{X}_{ij}(t) = \{\sigma(t)\}^{-1} \sum_{r=1}^{5} (a_{ijr}\sin(\pi t(5 - a_{ijr})/150) - m_{ijr} \tag{4.11}$$

where $t \in (1, 300)$, $a_{ijr} \sim U(0, 5)$, and $m_{ijr} \sim U(0, 2\pi)$. Note that $U(a, b)$ denotes the Uniform distribution on the interval $[a, b]$. Here, $\sigma(t)$ is defined such that the sample variance of all components of the generated $\mathbf{X}_{ij}$ matrix is equal to 0.01. These simulation conditions were taken from Gertheiss et al. (2013) and McLean et al. (2014). The penalty used is the smoothness-sparsity penalty (1.17) and estimates are found using the Group Lasso approach outlined in Gertheiss et al. (2013).

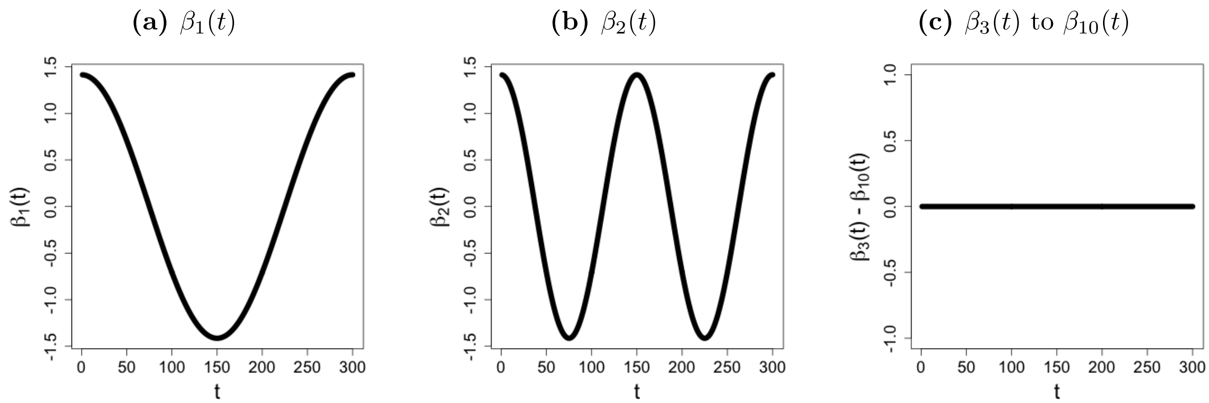

Figure 4.7: True values of the covariate functions for the simulated functional data.

The sequential approach (Seq) is performed over 100 λ_1 values between 1 and e^5 and 100 λ_2 values between 10 and 10^{10}. The initial observation of the APE curve was recorded for 5 values of λ_1 and λ_2 each. The tolerance was set to 0.01 and C was set to $\{10, 20, 50, 100\}$. It is extremely time consuming to calculate the APE function across all 100×100 points in Λ, so the sequential method is compared to the more traditional CV approach for a grid using 10 values of λ_1 and λ_2 each (Grid Search). CV was carried out over 5 folds.

The two methods are evaluated by comparing their results to the true underlying model and special notice is taken when the methods coincide. Consider the number of falsely included covariates (FP), the number of falsely excluded covariates (FN), the 1SE cutoff values found (h), the number of sampling points observed (Sampled) and directly compare the covariate function estimates using the squared error $\int(\hat{\beta}_j(t) - \beta_j(t))^2 dt$ for individual covariates, shown in Figures 4.8 and 4.9.

Table 4.3: Average results from 100 replications of the sequential APE process for the smoothness-sparsity penalty on functional data for varying values of C compared to the Grid Search (GS) approach for a sparser set of tuning parameters. Standard errors given in parentheses.

		GS	C = 10	C = 20	C = 50	C = 100
$n = 100$	FP	5.05	4.95	4.86	4.65	4.53
		(0.22)	(0.22)	(0.22)	(0.21)	(0.22)
	FN	0.02	0.01	0.01	0.01	0.00
		(0.01)	(0.01)	(0.01)	(0.01)	(0.00)
	h	0.77	0.75	0.75	0.76	0.76
		(0.03)	(0.02)	(0.02)	(0.02)	(0.02)
	Sampled	100	39.64	47.6	83.12	140.44
		(0.00)	(0.61)	(0.65)	(1.00)	(1.82)
$n = 500$	FP	3.6	2.65	2.57	2.47	2.36
		(0.19)	(0.19)	(0.18)	(0.17)	(0.16)
	FN	0.00	0.00	0.00	0.00	0.00
		(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
	h	0.4	0.41	0.42	0.42	0.42
		(0.01)	(0.01)	(0.01)	(0.01)	(0.01)
	Sampled	100	31.79	42.05	78.53	139.62
		(0.00)	(0.36)	(0.57)	(0.80)	(1.66)

The sequential method, for all values of C appears to always perform better, on average, than the Grid Search method. The best improvement is gained when $C = 100$, although—unsurprisingly—this method results in many more calculations than the other sequential results. When C is as small as 10, the average false positives, false negatives, and all the squared errors have slightly smaller averages and very similar standard errors as the Grid Search. All this is achieved, however, using an average of about 40% of the sampling points. Therefore, the sequential method can achieve extremely similar—if not slightly better—results than the more traditional CV approach, but with far fewer calculations.

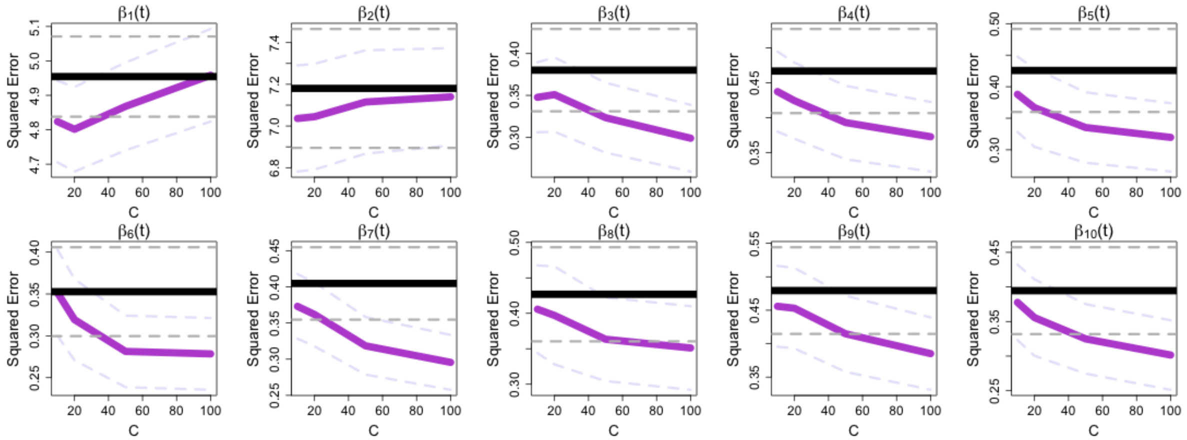

Figure 4.8: The average squared error, where squared error is defined $\int(\hat{\beta}_j(t) - \beta_j(t))^2 dt$, of each covariate function for $n = 100$. The purple line shows the results from the approximated method for increasing values of C. The black line shows the average squared error of the full observation of the sparse set of tuning parameters.

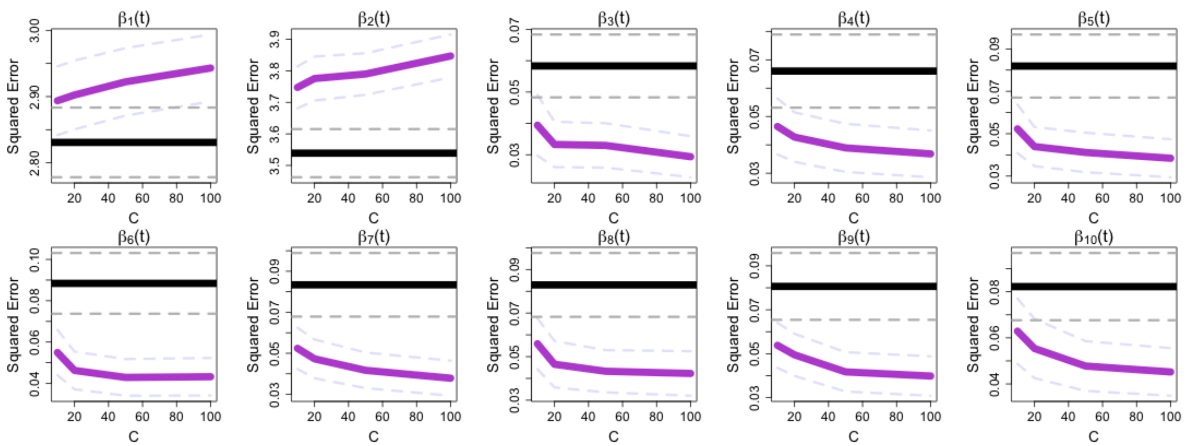

Figure 4.9: The average squared error, where squared error is defined $\int(\hat{\beta}_j(t) - \beta_j(t))^2 dt$, of each covariate function for $n = 500$. The purple line shows the results from the approximated method for increasing values of C. The black line shows the average squared error of the full observation of the sparse set of tuning parameters.

4.5 Application to EMG Data

As a demonstration of the sequential K-fold CV approach, consider once again the EMG data and the SAFE procedure proposed in Stallrich et al. (2020). Recall the model in question is given in (1.21) with penalty (1.22). This process requires as many as five stages to ensure accurate variable selection across a relatively sparse grid of λ_1, λ_t, and λ_z. The grid of tuning parameters in Stallrich et al. (2020) explored 80 values of $\log(\lambda_1)$ from -20 to 0 and $\log(\lambda_t), \log(\lambda_z) \in \{-10, -5, 0, 5, 10\}$, a total of 2000 discrete sampling points.

This relatively coarse—particularly over the smoothing parameters λ_t and λ_z—set of points over which to observe the APE function was chosen to limit the overall computation time of the estimation procedure. A better way to cut down the computation time without limiting the scope of the analysis by imposing coarse grids of tuning parameters may be via the sequential approach to finding optimal tuning parameters. There are two ways in which computation time might be improved using the sequential K-fold CV method. First, more complete exploration of the tuning parameter space could result in ideal models being chosen after fewer stages of SAFE. The other way is that the reduced number of calculations at each individual stage may make the process faster, so even if the same number of stages are required, each stage will be less computationally expensive.

The initial observation of the APE curve was done over a coarse grid of tuning parameters: 5 values of λ_1, 3 each values of λ_t and λ_z. The curve was explored over a much denser grid of 100 λ_1, λ_t, and λ_z values each, with the same ranges as the original estimation, for the sequential approach. C, the desired number of models within one standard error of the mean, was set to 20 and the tolerance was set to 0.001.

The success of variable selection of the models is evaluated using Size—the number of EMG signals active in the final model—and FP—the number of falsely included EMG signals in the final model. Recall that for the hand movement studied here only X_5, X_7, and X_{12} are necessary to explain movement, but because X_5 and X_7 relate to the same muscle contraction, the ideal model contains only X_5 and X_{12} or X_7 and X_{12}. A model with all three signals would contain no false positives. Also recorded are the Min APE and Min AR2, respectively the APE and AR2 from the model used to find the 1SE cutoff. If the approximation method is successful, then the Min APE from the Grid Search APE will be identical to or larger than the Min APE from the sequential APE method. Finally, Sampled is the number of sampling points evaluated to find the final model. Sampled for the Grid Search methods is always equal to 2000, the total number of points in the coarse grid. Results from the first stage—without any adaptive weighting—are given in Table 4.4, and results from the fifth and final stage of SAFE are presented in Table 4.5. Results from the intermediary stages can be found in Appendix C.

After the initial stage, the sequential APE approach does not return sparser models than the coarse grid for any movement type—returning a more complicated model in the case of FC1, FC2, FC3, FR1, and FR3. This can be attributed to the more thorough exploration of the tuning parameter space. Indeed, the Min APE is always smaller for the sequential APE method. That is, the Grid Search method appears to be missing the true minimum. After several adaptive stages the variable selection results are identical, despite the fact that the sequential method is still able to recover the smaller minimum APE for every data set. There is also a huge discrepancy in the number of sampling points evaluated. The sequential APE method is able to identify the same model with less than 10% of the APE calculations.

A similar pattern emerges when comparing Grid Search with Sequential AR2 results. In the first stage, the Grid Search method is able to return sparser models, but the sequential method finds the smaller minimum AR2. In the final stage, the variable selection results are identical, with one interesting exception. After the final stage of adaptive weighting, the sequential AR2 method has overselected by two covariates for FC3. This is because

Table 4.4: Model size, number of false positives (FP), and number of tuning parameter sampling points (Sampled) evaluated for the EMG data without adaptive weights. Performance is evaluated for observing the APE and AR2 curves over a grid of tuning parameters (Grid Search APE, Grid Search AR2), the approximation of the APE (Approximated APE) curve and the approximation of the AR2 curve (Approximated AR2) across a denser grid of tuning parameters.

		FC1	FC2	FC3	FR1	FR2	FR3	Mean
Size	Grid Search APE	4	3	4	5	4	4	4
	Sequential APE	6	4	8	7	4	5	5.667
	Grid Search AR2	2	3	6	2	4	3	3.333
	Sequential AR2	5	5	8	2	6	5	5.167
FP	Grid Search APE	2	1	2	2	1	1	1.5
	Sequential APE	3	1	5	4	1	2	2.667
	Grid Search AR2	0	1	3	0	1	1	1
	Sequential AR2	2	2	5	0	3	2	2.333
Min APE	Grid Search APE	0.076	0.088	0.096	0.2	0.156	0.172	0.131
	Sequential APE	0.067	0.078	0.078	0.183	0.141	0.168	0.119
	Grid Search AR2	0.073	0.086	0.105	0.219	0.163	0.174	0.137
	Sequential AR2	0.067	0.089	0.088	0.184	0.141	0.175	0.124
Min AR2	Grid Search APE	0.064	0.078	0.088	0.182	0.163	0.21	0.131
	Sequential APE	0.058	0.078	0.081	0.177	0.144	0.178	0.119
	Grid Search AR2	0.068	0.08	0.086	0.19	0.152	0.201	0.13
	Sequential AR2	0.058	0.08	0.075	0.177	0.138	0.182	0.118
Sampled	Grid Search APE	2000	2000	2000	2000	2000	2000	2000
	Sequential APE	84	92	136	92	93	78	95.833
	Grid Search AR2	2000	2000	2000	2000	2000	2000	2000
	Sequential AR2	83	83	93	73	113	93	89.667

the sequential approach was able to identify a smaller minimum AR2, and thus its target was different. Although Sequential AR2 led to overselection, it is noteworthy that the Grid Search method missed the true minimum AR2.

Table 4.5: Model size, number of false positives (FP), and number of tuning parameter sampling points (Sampled) evaluated for the EMG data with four stages of adaptive weighting. Performance is evaluated for observing the APE and AR2 curves over a grid of tuning parameters (Grid Search APE, Grid Search AR2), the approximation of the APE (Approximated APE) curve and the approximation of the AR2 curve (Approximated AR2) across a denser grid of tuning parameters.

		FC1	FC2	FC3	FR1	FR2	FR3	Mean
Size	Grid Search APE	2	2	2	2	3	2	2.167
	Sequential APE	2	2	2	2	3	2	2.167
	Grid Search AR2	2	2	2	2	4	2	2.333
	Sequential AR2	2	2	4	2	4	2	2.667
FP	Grid Search APE	0	0	0	0	0	0	0
	Sequential APE	0	0	0	0	0	0	0
	Grid Search AR2	0	0	0	0	1	0	0.167
	Sequential AR2	0	0	2	0	1	0	0.5
Min APE	Grid Search APE	0.073	0.084	0.093	0.198	0.142	0.177	0.128
	Sequential APE	0.061	0.074	0.087	0.192	0.137	0.174	0.121
	Grid Search AR2	0.074	0.096	0.096	0.203	0.168	0.19	0.138
	Sequential AR2	0.063	0.091	0.077	0.186	0.137	0.173	0.121
Min AR2	Grid Search APE	0.064	0.082	0.088	0.186	0.135	0.232	0.131
	Sequential APE	0.054	0.073	0.083	0.193	0.141	0.183	0.121
	Grid Search AR2	0.064	0.074	0.086	0.184	0.124	0.2	0.122
	Sequential AR2	0.055	0.078	0.068	0.179	0.131	0.177	0.115
Sampled	Grid Search APE	2000	2000	2000	2000	2000	2000	2000
	Sequential APE	116	107	74	77	108	70	92
	Grid Search AR2	2000	2000	2000	2000	2000	2000	2000
	Sequential AR2	77	107	104	83	114	88	95.5

4.6 Discussion

In this chapter, a sequential optimization approach to K-fold CV was proposed where tuning parameter values are sequentially added to optimize an acquisition function. The next stage of the process is exploring the space around the minimum APE in order to define the sparsest model within one standard error of the minimum APE. This approach will be useful for any practitioner who needs to evaluate multiple tuning parameters when the penalized estimation procedure is time-consuming because it will significantly reduce the number of calculations necessary to find the optimal set of tuning parameters. Results from the simulation study performed here are promising. For an application like that of the EMG data, where multi-dimensional tuning parameter selection is critical but penalized estimation is costly, this allows for more efficient and thorough exploration of the tuning parameter space without drastically adding to the computation time. In this application, where multiple adaptive stages are required, the computation time is a meaningful consideration.

In the simulation and application results, computational complexity is evaluated using the number of sampling points required—that is, the number of K-fold CV calculations needed. We acknowledge that this may not always be the best gauge of overall computation time. The benefit of Algorithms 2 and 3 is that they are flexible enough to be applied to any penalized estimation procedure where K-fold CV is required. In applying them to a particular penalty or model, however, it is possible to specify the algorithms to the problem in order to optimize their efficiency. For example, in the Group Lasso approach to functional linear models, calculations across a relatively dense grid of λ_1 values can be fast. The limiting factor in the overall computation time is the number of smoothness parameters, λ_t and λ_z, evaluated. In the interest of making the algorithm as fast as possible for this problem, it would therefore be advantageous to be parsimonious about sampling λ_t and λ_z. Sampling over λ_1, however, could be done much more freely. Another consideration to make is the fact that some penalized estimates benefit from the evaluation of multiple close tuning parameters sequentially because they have a "warm-start." This fact means that, potentially, it is faster to sequentially add batches of tuning parameters, rather than individual points.

The purpose of this work was to showcase the idea of using sequential optimization in K-fold CV for the purpose of minimizing the APE and finding the 1SE window around the APE. The algorithms given here can be broadly applied to any K-fold CV practice for any penalized estimator, but can be altered to exploit idiosyncrasies of individual problems.

Future avenues of exploration for this process include evaluating the use of acquisition functions other than Expected Improvement, such as UCB. Furthermore, the stationarity assumption on the APE function may not be reasonable because of the areas noted in Section 4.1 where the curve flattens out. Additional accuracy for the approximation may be achieved using Treed Partitioning for non-stationary modeling (Gramacy and Lee 2008). As discussed in Chapter 3, the use of a 1SE Rule for multidimensional tuning parameters in GLMs is complicated by the measurement of model complexity used. As such, closer study of how to implement the 1SE Rule in these cases may be necessary to improve the quality of estimates. Finally, GPs can be used to approximate noisy systems as well as deterministic systems. In this chapter, the folds have been fixed to eliminate noise from the system. An extension of the process would be to allow the folds to change, treating the different K-fold CV results that arise as noisy observations from the system.

CHAPTER

5

CONCLUSIONS AND FUTURE WORK

Tuning parameter selection for penalized estimators—particularly when tuning parameters are multidimensional—can indeed be a difficult task. For the Lasso and Lasso-type penalties, the biasing effect of the penalty can shrink coefficient estimates too much, leading to poor prediction even when the correct submodel has been recovered or overselection when the active coefficients are able to attain their necessary magnitudes. Even when tuning parameter selection does not lead to variable selection, such as for the Ridge penalty, it can still be a struggle to balance model fit with model complexity. For GAMs and functional linear models, the task of tuning parameter selection is made even more difficult because both smoothness and sparsity are imposed. K-Fold CV, one of the most popular methods of tuning parameter selection, can lead to overspecified models and can be computationally intensive to achieve when fully exploring the tuning parameter space.

This dissertation has addressed these concerns with new proposals for K-fold CV. For the linear model, AR2 CV can be used when variable selection is taking place to evaluate model fit without the inconvenience of the scaling issue. A very similar approach is α-Modified CV, which deals with the scaling problem in a different way but has also been shown to lead to sparser models than traditional CV for the Lasso. The new CV methods led to a de-biasing approach known as the α-Modification. This is a direct answer to the shrinkage problem of Lasso and Lasso-type penalties, but can be used to enhance estimates under any penalty. The correction—even when used in conjunction with K-fold CV—is generally faster than finding Relaxed estimates because closed form solutions for $\hat{\alpha}_\lambda$ exist for the linear model. Although the α-Modified Lasso does not always out-perform the Relaxed Lasso, it can be thought of as a quick and simple approximation to the Relaxed Lasso that can be used more universally.

In Chapter 2, it is shown that the α-Modification will never shrink penalized estimates for the linear model further towards zero, under reasonable penalty constraints. Furthermore, the α-Modification maintains the consistency properties of any estimator it modifies and whenever an estimator approaches a scalar of the truth, the α-Modified version of the estimator will be consistent. These results do not rely on any specific penalty and can be applied to many different types of penalties. The finite sample theoretical results presented here focus on the Lasso penalty. They show that, when $\mathbf{X}$ is orthogonal, the absolute difference between the α-Modified Lasso and the OLS estimate is bounded. When $\mathbf{X}$ is not orthogonal, the absolute value approaches a constant that is less than the absolute difference between the OLS estimate and the unmodified Lasso estimate as the signal increases to infinity.

Chapter 3 extended the α-Modification in two ways. First, by applying it to GLMs, and next by adding a penalty to α to allow for more control over the impact that the scaling factor will have on estimates. The fact that estimates of $\hat{\alpha}_{\lambda,\nu}$ are always within $[1, \hat{\alpha}_\lambda]$ means that K-fold CV will be able to evaluate both fully α-Modified estimates, and estimates with no modification whatsoever, as well as many values in between. Therefore, those λ for which the α-Modification can exacerbate bias problems are less of a concern. An algorithm is presented to find solutions for both methods. Similarly to the linear model, it has been proven that $\hat{\alpha}_\lambda$ is greater than or equal to 1 for many penalties, showing that the α-Modification, even when applied to GLM estimates, will never further shrink estimates when the penalty meets certain conditions. The finite sample results from this chapter focus on the linear model once again. It is shown that there always exists a ν such that the α_ν-Modified Lasso is closer to the oracle OLS estimate whenever the sign vector has been recovered for any $\mathbf{X}$ matrix.

Finally, in Chapter 4, a new way to perform K-fold CV is suggested, whereby sequential design procedures are employed to more completely explore the APE curve with a Gaussian process approximation. This can drastically reduce the number of calculations needed to thoroughly explore the tuning parameter space to find, not only the minimum APE λ, but also the one standard error area around the minimum so that a 1SE Rule can be used.

Many areas of future study were discussed in more detail in Chapters 2, 3, and 4. Two particularly important areas, however, will be highlighted here in conclusion. Firstly, the α- and α_ν-Modifications are found using in-sample data. Because the topics of generalizability and overspecification are so fundamental to this discussion, the question of whether out-of-sample data could be used to even greater effect should be addressed. Next, the question of how to implement a 1SE Rule for multidimensional tuning parameters and GLMs is one that deserves additional scrutiny. The simulations from Chapter 2 showed the greatest advantage of the α-Modification was when it was used in conjunction with a 1SE Rule. Perhaps even more promising results for the α- and α_ν Modifications to logistic and Poisson Lasso regression are possible under a different 1SE Rule.

REFERENCES

Akaike, H. (1974) A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19**, 716–723.

Albert, A. and Anderson, J. A. (1984) On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, **71**, 1–10. URLhttp://www.jstor.org/stable/2336390.

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

Bergstra, J., Bardenet, R., Bengio, Y. and Kégl, B. (2011) Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems* (eds. J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira and K. Weinberger), vol. 24. Curran Associates, Inc. URLhttps://proceedings.neurips.cc/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

Berk, J., Gupta, S., Rana, S. and Venkatesh, S. (2020) Randomised gaussian process upper confidence bound for bayesian optimisation. *CoRR*, **abs/2006.04296**. URLhttps://arxiv.org/abs/2006.04296.

Bia, M. and Kerm, P. V. (2014) Space-filling location selection. *The Stata Journal*, **14**, 605–622. URLhttps://doi.org/10.1177/1536867X1401400308.

Bickel, P. J., Ritov, Y. and Tsybakov, A. B. (2009) Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, **37**, 1705 – 1732. URLhttps://doi.org/10.1214/08-AOS620.

Breiman, L. (1995) Better subset regression using the nonnegative garrote. *Technometrics*, **37**, 373–384.

Brownlee, J. (2020) How to implement bayesian optimization from scratch in python. https://machinelearningmastery.com/what-is-bayesian-optimization/. Accessed: 2022-04-11.

Buhlmann, P. and van de Geer, S. (2011) *Statistics for high-dimensional data*. Springer Series in Statistics. Springer, Heidelberg. Methods, theory and applications.

Bull, A. D. (2011) Convergence rates of efficient global optimization algorithms. URLhttps://arxiv.org/abs/1101.3501.

Chatterjee, A. and Lahiri, S. N. (2011) Strong consistency of lasso estimators. *Sankhyā: The Indian Journal of Statistics, Series A (2008-)*, **73**, 55–78.

Chen, J. and Chen, Z. (2008) Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, **95**, 759–771. URLhttp://www.jstor.org/stable/20441500.

— (2012) Extended bic for small-n-large-p sparse glm. *Statistica Sinica*, **22**, 555–574. URLhttp://www.jstor.org/stable/24310025.

Chetverikov, D., Liao, Z. and Chernozhukov, V. (2021) On cross-validated Lasso in high dimensions. *The Annals of Statistics*, **49**, 1300 – 1317.

Cho, H., Kim, Y., Lee, E., Choi, D., Lee, Y. and Rhee, W. (2020) Basic enhancement strategies when using bayesian optimization for hyperparameter tuning of deep neural networks. *IEEE Access*, **8**, 52588–52608.

Cinar, I., Koklu, M. and Tasdemir, S. (2020) Classification of raisin grains using machine vision and artificial intelligence methods. *Gazi Journal of Engineering Sciences*, **6**, 200–209.

Craven, P. and Wahba, G. (1978) Smoothing noisy data with spline functions. *Numerische Mathematik*, **31**, 377–403.

Dicker, L., Huang, B. and Lin, X. (2013) Variable selection and estimation with the seamless-l 0 penalty. *Statistica Sinica*, **23**, 929–962. URLhttp://www.jstor.org/stable/24310368.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004) Least angle regression. *The Annals of Statistics*, **32**, 407–499.

Fan, J. and Li, R. (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, **96**, 1348–1360.

Fan, Y., James, G. M. and Radchenko, P. (2015) Functional additive regression. *The Annals of Statistics*, **43**, 2296 – 2325. URLhttps://doi.org/10.1214/15-AOS1346.

Feng, Y. and Yu, Y. (2013) Consistent cross-validation for tuning parameter selection in high-dimensional variable selection. *arXiv preprint arXiv:1308.5390*, **1**.

Frank, I. E. and Friedman, J. H. (1993) A statistical view of some chemometrics regression tools. *Technometrics*, **35**, 109–135. URLhttp://www.jstor.org/stable/1269656.

Friedman, J. H., Hastie, T. and Tibshirani, R. (2010) Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, **33**, 1–22. URLhttps://www.jstatsoft.org/index.php/jss/article/view/v033i01.

Fuchs, J.-J. (2004) Recovery of exact sparse representations in the presence of noise. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, ii–533.

van de Geer, S., Bühlmann, P., Ritov, Y. and Dezeure, R. (2014) On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, **42**.

Geisser, S. (1975) The predictive sample reuse method with applications. *Journal of the American Statistical Association*, **70**, 320–328.

Gertheiss, J., Maity, A. and Staicu, A.-M. (2013) Variable selection in generalized functional linear models. *Stat*, **2**, 86–101.

Golub, G. H., Heath, M. and Wahba, G. (1979) Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, **21**, 215–223.

Gramacy, R. B. (2020) *Surrogates: Gaussian Process Modeling, Design and Optimization for the Applied Sciences*. Boca Raton, Florida: Chapman Hall/CRC.

Gramacy, R. B. and Lee, H. K. H. (2008) Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, **103**, 1119–1130.

Hastie, T., Friedman, J. and Tisbshirani, R. (2017) *The Elements of statistical learning: data mining, inference, and prediction*. Springer.

Hastie, T. and Tibshirani, R. (1986) Generalized Additive Models. *Statistical Science*, **1**, 297 – 310. URL`https://doi.org/10.1214/ss/1177013604`.

Hastie, T., Tibshirani, R. and Wainwright, M. (2015) *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC.

Hoerl, A. E. and Kennard, R. W. (1970) Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, **12**, 55–67. URL`https://www.tandfonline.com/doi/abs/10.1080/00401706.1970.10488634`.

Homrighausen, D. and McDonald, D. J. (2018) A study on tuning parameter selection for the high-dimensional lasso. *Journal of Statistical Computation and Simulation*, **88**, 2865–2892.

Huang, J., Horowitz, J. L. and Ma, S. (2008) Asymptotic properties of bridge estimators in sparse high-dimensional regression models. *The Annals of Statistics*, **36**, 587 – 613. URL`https://doi.org/10.1214/009053607000000875`.

Hui, F. K. C., Warton, D. I. and Foster, S. D. (2015) Tuning parameter selection for the adaptive lasso using eric. *Journal of the American Statistical Association*, **110**, 262–269.

Hurvich, C. M. and Tsai, C.-L. (1989) Regression and time series model selection in small samples. *Biometrika*, **76**, 297–307.

Inatsu, Y., Karasuyama, M., Inoue, K. and Takeuchi, I. (2020) Active Learning for Level Set Estimation Under Input Uncertainty and Its Extensions. *Neural Computation*, **32**, 2486–2531. URL`https://doi.org/10.1162/neco_a_01332`.

James, G. M. (2002) Generalized linear models with functional predictors. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, **64**, 411–432. URL`http://www.jstor.org/stable/3088780`.

Javanmard, A. and Montanari, A. (2013) Confidence intervals and hypothesis testing for high-dimensional statistical models. In *Advances in Neural Information Processing Systems* (eds. C. Burges, L. Bottou, M. Welling, Z. Ghahramani and K. Weinberger), vol. 26. Curran Associates, Inc.

Jia, J. and Yu, B. (2010) On model selection consistency of the elastic net when p $\gg$ n. *Statistica Sinica*, **20**, 595–611.

Karush, W. (1939) *Minima of Functions of Several Variables with Inequalities as Side Conditions*. Master's thesis, Department of Mathematics, University of Chicago, Chicago, IL, USA.

Kibria, B. M. G., Månsson, K. and Shukur, G. (2015) A simulation study of some biasing parameters for the ridge type estimation of poisson regression. *Communications in Statistics - Simulation and Computation*, **44**, 943–957. URL`https://doi.org/10.1080/03610918.2013.796981`.

Knight, K. and Fu, W. (2000) Asymptotics for lasso-type estimators. *The Annals of Statistics*, **28**, 1356–1378. URL`http://www.jstor.org/stable/2674097`.

Koehrsen, W. (2018) A conceptual explanation of bayesian hyperparameter optimization for machine learning. `https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f`. Accessed: 2022-04-11.

Krstajic, D., Buturovic, L., Leahy, D. and Thomas, S. (2014) Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of cheminformatics*, **6**, 10.

Kuhn, H. W. and Tucker, A. W. (1951) Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950*, 481–492. Berkeley and Los Angeles: University of California Press.

Lahiri, S. N. (2021) Necessary and sufficient conditions for variable selection consistency of the LASSO in high dimensions. *The Annals of Statistics*, **49**, 820 – 844. URL`https://doi.org/10.1214/20-AOS1979`.

Lee, J. D., Sun, D. L., Sun, Y. and Taylor, J. E. (2016) Exact post-selection inference, with application to the lasso. *The Annals of Statistics*, **44**, 907 – 927.

Li, W. and Lederer, J. (2019) Tuning parameter calibration for ℓ_1-regularized logistic regression.

Li, Z., Wang, S. and Lin, X. (2012) Variable selection and estimation in generalized linear models with the seamless l_0 penalty. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, **40**, 745–769. URL`http://www.jstor.org/stable/41724559`.

Lockhart, R., Taylor, J., Tibshirani, R. J. and Tibshirani, R. (2014) A significance test for the lasso. *The Annals of Statistics*, **42**, 413–468.

McLean, M. W., Hooker, G., Staicu, A.-M., Scheipl, F. and Ruppert, D. (2014) Functional generalized additive models. *Journal of Computational and Graphical Statistics*, **23**, 249–269. URL`https://doi.org/10.1080/10618600.2012.729985`. PMID: 24729671.

Meier, L., van de Geer, S. and Bühlmann, P. (2009) High-dimensional additive modeling. *The Annals of Statistics*, **37**, 3779 – 3821. URL`https://doi.org/10.1214/09-AOS692`.

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

Meinshausen, N. and Bühlmann, P. (2006) High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, **34**, 1436 – 1462.

Mockus, J., Tiesis, V. and Zilinskas, A. (1978) The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, **2**, 2.

Müller, H.-G. and Stadtmüller, U. (2005) Generalized functional linear models. *The Annals of Statistics*, **33**, 774 – 805. URL`https://doi.org/10.1214/009053604000001156`.

Nardi, Y. and Rinaldo, A. (2008) On the asymptotic properties of the group lasso estimator for linear models. *Electronic Journal of Statistics*, **2**, 605 – 633.

Nishii, R. (1984) Asymptotic Properties of Criteria for Selection of Variables in Multiple Regression. *The Annals of Statistics*, **12**, 758 – 765.

Osborne, M. A. (2010) *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. Ph.D. thesis, Oxford University, UK.

Park, M. Y. and Hastie, T. (2007) L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, **69**, 659–677. URL`http://www.jstor.org/stable/4623289`.

Ramsay, J. O. and Silverman, B. W. (2005) *Functional Data Analysis*. Springer. URLhttp://www.worldcat.org/isbn/9780387400808.

Roberts, S. and Nowak, G. (2014) Stabilizing the lasso against cross-validation variability. *Computational Statistics & Data Analysis*, **70**, 198–211. URLhttps://www.sciencedirect.com/science/article/pii/S016794731300323X.

Royle, J. and Nychka, D. (1998) An algorithm for the construction of spatial coverage designs with implementation in splus. *Computers & Geosciences*, **24**, 479–488. URLhttps://www.sciencedirect.com/science/article/pii/S009830049800020X.

Santner, T., Williams, B. and Notz, W. (2003) *The Design and Analysis Computer Experiments*.

Sauter, R. and Held, L. (2016) Quasi-complete separation in random effects of binary response mixed models. *Journal of Statistical Computation and Simulation*, **86**, 2781–2796.

Schwarz, G. (1978) Estimating the dimension of a model. *The Annals of Statistics*, **6**, 461–464.

Segerstedt, B. (1992) On ordinary ridge regression in generalized linear models. *Communications in Statistics - Theory and Methods*, **21**, 2227–2246. URLhttps://doi.org/10.1080/03610929208830909.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P. and de Freitas, N. (2016) Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, **104**, 148–175.

Shi, X., Liang, B. and Zhang, Q. (2020) Post-selection inference of generalized linear models based on the lasso and the elastic net. *Communications in Statistics - Theory and Methods*, **0**, 1–18.

Srinivas, N., Krause, A., Kakade, S. M. and Seeger, M. W. (2012) Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, **58**, 3250–3265. URLhttp://dx.doi.org/10.1109/TIT.2011.2182033.

Stallrich, J., Islam, M. N., Staicu, A.-M., Crouch, D., Pan, L. and Huang, H. (2020) Optimal EMG placement for a robotic prosthesis controller with sequential, adaptive functional estimation (SAFE). *The Annals of Applied Statistics*, **14**, 1164 – 1181.

Stone, M. (1974) Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, **36**, 111–147.

Tan, M. H. Y. (2013) Minimax designs for finite design regions. *Technometrics*, **55**, 346–358. URLhttps://doi.org/10.1080/00401706.2013.804439.

Tay, J. K., Narasimhan, B. and Hastie, T. (2021) Elastic net regularization paths for all generalized linear models. *arXiv preprint arXiv:2103.03475*.

Taylor, J. and Tibshirani, R. J. (2015) Statistical learning and selective inference. *Proceedings of the National Academy of Sciences*, **112**, 7629–7634.

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

Tibshirani, R. J. (2013) The lasso problem and uniqueness. *Electronic Journal of Statistics*, **7**, 1456 – 1490. URL`https://doi.org/10.1214/13-EJS815`.

Tibshirani, R. J., Taylor, J. E., Lockhart, R. A. and Tibshirani, R. (2014) Exact post-selection inference for sequential regression procedures. *Journal of the American Statistical Association*, **111**, 600 – 620.

Tropp, J. (2006) Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, **52**, 1030–1051.

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**, 2183–2202.

Wu, Y. (2021) Can't ridge regression perform variable selection? *Technometrics*, **63**, 263–271. URL`https://doi.org/10.1080/00401706.2020.1791254`.

Xia, L., Nan, B. and Li, Y. (2020) A revisit to de-biased lasso for generalized linear models. *arXiv preprint arXiv:2006.12778*.

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

Zhang, C.-H. (2010) Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, **38**, 894 – 942.

Zhang, C.-H. and Zhang, S. S. (2014) Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **76**, 217–242.

Zhao, P. and Yu, B. (2006) On model selection consistency of lasso. *Journal of Machine Learning Research*, **7**, 2541–2563.

Zou, H. (2006) The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, **101**, 1418–1429.

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

Zou, H., Hastie, T. and Tibshirani, R. (2007) On the "degrees of freedom" of the lasso. *The Annals of Statistics*, **35**, 2173 – 2192.

APPENDICES

APPENDIX

A

SUPPLEMENTARY MATERIAL FOR CHAPTER 2

The supplementary material in the following appendix includes two sections. Section A.1 provides tables and figures that more fully explore the simulation studies discussed in the Numerical Results. This section also evaluates results from a simulation study of the non-convex SCAD and MC+ penalties. Section A.2 describes the approximations used to achieve the estimates from the EMG data application and includes results from multiple stages of adaptive estimation.

A.1 Further Numerical Results

A.1.1 Simulation Study for Independent Predictors

In this section we give additional results from the simulation studies conducted in the main document. Tables **??** through A.20 give the average model size, average False Discovery Rates, average false negatives, and average prediction bias for the four methods considered in this part of the study: APE CV, AR2 CV, Mod CV, and the Relaxed Lasso for an additional p^*, $p^* = 15$. Mod CV was left out of the main document because of its similarity to AR2 CV. We can see from these results that all False negatives are close to zero. For smaller p^*, AR2 and Mod CV generally produce lower False Discovery Rates than APE CV. Average Prediction Bias, however generally appears lower for the two new approaches than APE CV, but higher than the Relaxed Lasso on average.

Table A.1: Average Hamming Distance for 100 replications of CV 1SE. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100	3.1	4.3	6	7.8	9.1	2.2	4	7.5	9.3	10.7
	(0.3)	(0.4)	(0.6)	(1.1)	(1.2)	(0.2)	(0.4)	(1.2)	(1)	(1.2)
500	0.6	1	1.3	1.9	2.9	0.9	0.9	1.7	1.7	1.5
	(0.1)	(0.2)	(0.2)	(0.5)	(0.5)	(0.2)	(0.1)	(0.3)	(0.3)	(0.3)
1000	0.2	0.4	0.4	0.6	0.9	0.1	0.4	0.5	0.7	0.9
	(0)	(0.1)	(0.1)	(0.1)	(0.2)	(0)	(0.1)	(0.1)	(0.1)	(0.2)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100	10.1	14.1	16.7	19.6	21.4	9.3	15.7	22.5	25.5	30.9
	(0.3)	(0.6)	(0.8)	(1.4)	(1.6)	(0.4)	(0.8)	(1.1)	(1.4)	(2)
500	4.1	7.5	9.7	11.4	15.7	3.9	6.1	10.4	13.9	16.6
	(0.3)	(0.4)	(0.6)	(0.6)	(1)	(0.2)	(0.4)	(0.6)	(0.8)	(1)
1000	2.8	4.1	6.5	8.4	9.4	2.6	3.8	6.4	7.8	10.4
	(0.2)	(0.3)	(0.5)	(0.6)	(0.6)	(0.2)	(0.3)	(0.5)	(0.6)	(0.7)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100	31.7	43.5	50.1	51.1	53.2	5	32	47	53.8	54.5
	(1.3)	(0.6)	(0.5)	(0.4)	(1.1)	(0.4)	(0.6)	(0.7)	(1.1)	(1.1)
500	1.6	18.2	33.3	48.6	57.4	0 (0)	16.8	35.3	58.4	81.2
	(0.2)	(0.5)	(0.8)	(1.3)	(1.8)		(0.5)	(0.9)	(1.6)	(2)
1000	0.4	14.5	28.5	43.5	57.6	0 (0)	14.2	28	44.8	64.2
	(0.1)	(0.4)	(0.9)	(1.4)	(1.6)		(0.5)	(0.8)	(1.3)	(1.8)

Table A.2: Average Hamming Distance for 100 replications of AR2 CV 1SE. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100	2.1 (0.3)	2.8 (0.3)	3.3 (0.3)	4.9 (0.7)	5 (0.5)	0.6 (0.1)	0.6 (0.1)	1.9 (0.8)	1.8 (0.4)	2.6 (0.4)
500	0.1 (0)	0.2 (0)	0.1 (0)	0.2 (0)	0.4 (0.1)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
1000	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100	10.1 (0.3)	14.9 (0.7)	18.1 (1.1)	20.8 (1.5)	22.9 (1.6)	6.4 (0.4)	10.2 (0.6)	15.9 (1)	17.7 (1)	21.8 (1.4)
500	1.8 (0.2)	2.8 (0.3)	3.2 (0.2)	4.2 (0.3)	5.3 (0.4)	0.3 (0.1)	0.3 (0.1)	0.6 (0.1)	1.4 (0.2)	1.5 (0.2)
1000	0.4 (0.1)	0.6 (0.1)	0.9 (0.2)	1.2 (0.1)	1.3 (0.1)	0 (0)	0 (0)	0 (0)	0.1 (0)	0.1 (0)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100	23.6 (1.3)	40.4 (0.5)	51.4 (0.8)	55.1 (1.2)	54.8 (1.2)	4.2 (0.4)	31.9 (0.5)	47.9 (0.8)	54.6 (1.3)	53.5 (0.9)
500	1.8 (0.2)	17.1 (0.5)	28.7 (0.8)	42.3 (1.3)	51.9 (2.2)	0.2 (0)	9.1 (0.4)	17.8 (0.7)	30 (1)	46.6 (1.3)
1000	0.7 (0.1)	9.1 (0.4)	15.4 (0.6)	23.1 (0.9)	32.5 (1.1)	0 (0)	3.9 (0.3)	6.3 (0.4)	11.1 (0.6)	16.9 (0.8)

We also evaluated the results for more limited data settings—fixing n and p, but varying p^* and SNR across more methods of tuning parameter selection, including AIC, BIC, ERIC with $\nu = 0.5$, and GCV. For AIC and BIC, fixed $\hat{\sigma}^2$ was calculated as the MSE of the model chosen by CV 1SE whereas ERIC was calculated assuming unknown variance for the likelihood. An additional alternative we consider is a simple Cutoff Method (R2 Cut). Without splitting the data, R^2 is calculated for all λ and the largest λ that achieves some threshold is chosen as the optimal.

Figures A.1 and A.2 give the average false positives and negatives for the simulated data provided in the main document in Figure 2.5. For larger sample sizes, the number of false negatives is close to zero for most p^*, but for $n = 100$, as noted in the main document, average false negatives increase with p^*.

For the following results, a simple cutoff of 0.95 was used in the R2 Cut method. Table A.21 gives the average false negatives from the setting considered in the main document: $n = p = 100$. Figure A.4 gives the average false positives and prediction bias for data with $n = 100$ and $p = 800$. Table A.5 gives average false positives and prediction bias

Table A.3: Average Hamming Distance for 100 replications of α-modified CV 1SE. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100	2.3 (0.3)	2.7 (0.3)	3.7 (0.4)	5.9 (1)	6.8 (1.3)	0.5 (0.1)	0.9 (0.1)	2.7 (0.9)	2.7 (0.6)	3.1 (0.5)
500	0 (0)	0.3 (0.1)	0.1 (0)	0.3 (0.2)	0.5 (0.3)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
1000	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100	9.9 (0.3)	14 (0.6)	16.8 (0.9)	19.1 (1.4)	21.8 (1.8)	7.2 (0.4)	11.6 (0.6)	18 (1.1)	21.1 (1.4)	27.9 (2)
500	1.7 (0.2)	2.8 (0.3)	3.3 (0.2)	4.2 (0.3)	5 (0.3)	0.6 (0.1)	0.5 (0.1)	0.9 (0.1)	2.1 (0.3)	2 (0.3)
1000	0.4 (0.1)	0.6 (0.1)	0.9 (0.1)	1.3 (0.1)	1.4 (0.1)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.2 (0.1)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100	35.7 (1.5)	45.2 (0.6)	51.5 (0.5)	51.5 (0.7)	54.3 (1.4)	4.9 (0.5)	32.7 (0.6)	48.7 (0.8)	55.5 (1.2)	55.6 (1.4)
500	1.7 (0.2)	16.4 (0.4)	26.5 (0.6)	36.8 (1)	43.9 (1.5)	0.1 (0)	9.9 (0.4)	20.3 (0.7)	34.1 (1.1)	51.5 (1.4)
1000	0.7 (0.1)	8.6 (0.4)	14.5 (0.5)	21.6 (0.7)	27.9 (1)	0 (0)	4.6 (0.3)	7.8 (0.4)	13.6 (0.6)	20 (1)

for $n = 1000$ and $p = 100$. Tables A.22, A.23, and A.24 give the false discovery rates, prediction bias, and false negatives for this last setting.

Table A.4: Average Hamming Distance for 100 replications of Relaxed Lasso. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100	3.7	5.4	7.4	18.4	28.1	1.1	1.8	4.5	6.6	7.9
	(0.4)	(0.8)	(1.1)	(3.4)	(4.2)	(0.3)	(0.5)	(1.7)	(2.3)	(2.6)
500	0.8	1.2	0.5	0.7	1.5	1.2	0.8	0.8	0.3	0.1
	(0.3)	(0.3)	(0.2)	(0.4)	(0.6)	(0.3)	(0.2)	(0.4)	(0.1)	(0)
1000	0.6	0.5	0.7	0.4	1.9	0.6	0.8	0.4	0.4	0.4
	(0.2)	(0.2)	(0.3)	(0.1)	(1.5)	(0.2)	(0.3)	(0.2)	(0.2)	(0.2)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100	12.3	18.8	32.2	45.3	55.9	9.1	14.2	34.2	81.1	82.3
	(0.6)	(1.1)	(3)	(4.3)	(4.9)	(0.7)	(1)	(3.4)	(4.8)	(4.7)
500	2.9	3.8	4.4	5.1	5	0.6	0.7	0.4	0.8	0.8
	(0.5)	(0.6)	(0.5)	(0.8)	(0.4)	(0.1)	(0.2)	(0.1)	(0.2)	(0.1)
1000	1	0.8	1.8	1.2	0.9	0.3	0.3	0.1	0.2	0.1
	(0.3)	(0.1)	(0.8)	(0.2)	(0.1)	(0.1)	(0.1)	(0)	(0.1)	(0)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100	19.5	39.9	57.6	70.8	67.9	2.1	34.1	75.9	87.6	93.6
	(1.3)	(0.5)	(1.7)	(3.1)	(3.2)	(0.3)	(0.5)	(2.5)	(3.3)	(3.8)
500	0.9	22.9	39.6	66.2	74.5	0.1	7.2	14.8	25.4	36.8
	(0.1)	(0.9)	(1.8)	(3.4)	(4.2)	(0)	(0.5)	(1.2)	(2.3)	(2.7)
1000	0.2	12.3	17.5	27.9	32.7	0 (0)	2.4	3.1	4.9	5.5
	(0)	(0.9)	(1.4)	(2.7)	(2.9)		(0.3)	(0.3)	(0.5)	(0.3)

Table A.5: Average number of selected variables 100 replications of CV 1SE. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100		6.9	7.6	9.3	10.4	11.1	7.1	8.9	12.4	14.2	15.5
		(0.4)	(0.5)	(0.7)	(1.2)	(1.3)	(0.2)	(0.4)	(1.2)	(1)	(1.2)
500		5.6	6	6.2	6.9	7.9	5.9	5.9	6.7	6.7	6.5
		(0.1)	(0.2)	(0.2)	(0.5)	(0.5)	(0.2)	(0.1)	(0.3)	(0.3)	(0.3)
1000		5.2	5.4	5.4	5.6	5.9	5.2	5.4	5.5	5.7	5.9
		(0)	(0.1)	(0.1)	(0.1)	(0.2)	(0)	(0.1)	(0.1)	(0.1)	(0.2)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100		14.5	15.5	12.2	12.6	11.5	23.4	28.7	33.9	33.6	35.5
		(0.7)	(1.1)	(1.3)	(1.9)	(2.1)	(0.4)	(0.9)	(1.3)	(1.8)	(2.4)
500		18.9	21.8	24.1	25.2	29.5	18.9	21.1	25.4	28.9	31.6
		(0.3)	(0.4)	(0.6)	(0.7)	(1)	(0.2)	(0.4)	(0.6)	(0.8)	(1)
1000		17.7	19	21.4	23.2	24.2	17.6	18.8	21.4	22.8	25.4
		(0.2)	(0.3)	(0.5)	(0.6)	(0.6)	(0.2)	(0.3)	(0.5)	(0.6)	(0.7)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100		18.3	14.6	12.3	4.4	5.5	45	45.8	30.2	19.7	12.8
		(1.3)	(1.5)	(1.8)	(1.1)	(1.6)	(0.4)	(1.7)	(2.4)	(2.5)	(2.1)
500		48.4	60.5	71.5	77.3	76.8	50	66.6	84.8	107.1	129
		(0.2)	(0.6)	(1.1)	(2)	(2.6)	(0)	(0.5)	(0.9)	(1.7)	(2.1)
1000		49.6	62.9	76	88.8	100.9	50	64.2	78	94.7	114.2
		(0.1)	(0.5)	(0.9)	(1.5)	(1.7)	(0)	(0.5)	(0.8)	(1.3)	(1.8)

Table A.6: Average number of selected variables 100 replications of AR2 CV 1SE. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^{*} = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^{*} = 5; SNR = 5; \bar{\sigma} = 2.59$				
100	4.8 (0.4)	5 (0.4)	5.5 (0.4)	6.4 (0.9)	6.1 (0.6)	5.3 (0.1)	5.4 (0.1)	6.6 (0.8)	6.4 (0.4)	7.1 (0.5)
500	4.9 (0)	4.9 (0)	4.9 (0)	5 (0)	5.2 (0.1)	5 (0)	5 (0)	5 (0)	5 (0)	5 (0)
1000	5 (0)	5 (0)	5 (0)	5 (0)	5 (0)	5 (0)	5 (0)	5 (0)	5 (0)	5 (0)
n	$p^{*} = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^{*} = 15; SNR = 5; \bar{\sigma} = 4.53$				
100	13.9 (0.8)	16.6 (1.2)	13.7 (1.6)	15.7 (2)	14.9 (2.1)	19.6 (0.4)	21.7 (0.7)	25.6 (1.2)	22.6 (1.4)	23.9 (1.8)
500	15.4 (0.2)	16.3 (0.3)	16.4 (0.3)	16.7 (0.4)	17.5 (0.5)	15.3 (0.1)	15.2 (0.1)	15.4 (0.1)	16.3 (0.2)	16.3 (0.2)
1000	15 (0.1)	15.1 (0.1)	15.3 (0.2)	15.5 (0.2)	15.5 (0.2)	15 (0)	15 (0)	15 (0)	15 (0)	15.1 (0)
n	$p^{*} = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^{*} = 50; SNR = 5; \bar{\sigma} = 8.3$				
100	26.4 (1.3)	25 (1.8)	22 (2.2)	14.4 (2.2)	9.9 (1.7)	45.8 (0.4)	47.5 (1.7)	36.5 (2.5)	23 (2.7)	13.9 (1.8)
500	48.2 (0.2)	57.6 (0.7)	63.4 (1.2)	67.6 (2.1)	67.4 (3.1)	49.8 (0)	58.4 (0.4)	66.4 (0.7)	77.1 (1.1)	92.1 (1.5)
1000	49.3 (0.1)	56 (0.5)	60.1 (0.8)	64.3 (1.1)	71.8 (1.4)	50 (0)	53.7 (0.3)	56.1 (0.4)	60.7 (0.6)	66.4 (0.8)

Table A.7: Average number of selected variables 100 replications of α-modified CV 1SE. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100	5.1 (0.4)	5.1 (0.4)	5.9 (0.5)	7.6 (1.2)	7.8 (1.5)	5.3 (0.1)	5.7 (0.2)	7.5 (0.9)	7.5 (0.6)	7.8 (0.5)
500	5 (0)	5 (0.1)	4.9 (0)	5.1 (0.2)	5.3 (0.3)	5 (0)	5 (0)	5 (0)	5 (0)	5 (0)
1000	5 (0)	5 (0)	5 (0)	5 (0)	5 (0)	5 (0)	5 (0)	5 (0)	5 (0)	5 (0)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100	11.7 (0.7)	11.9 (1.2)	9.2 (1.5)	10.1 (2)	11.1 (2.3)	20.7 (0.4)	23.8 (0.7)	28 (1.4)	27.3 (1.8)	30.8 (2.5)
500	15.5 (0.2)	16.4 (0.3)	16.4 (0.3)	16.3 (0.4)	17 (0.4)	15.5 (0.1)	15.4 (0.1)	15.8 (0.1)	17 (0.3)	16.9 (0.3)
1000	15.1 (0.1)	15.1 (0.1)	15.3 (0.2)	15.6 (0.2)	15.6 (0.2)	15.1 (0)	15.1 (0)	15.1 (0)	15.1 (0)	15.2 (0.1)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100	14.3 (1.5)	11.4 (1.6)	11.4 (2.1)	4.4 (1.4)	7 (2.1)	45.1 (0.5)	44.9 (1.9)	31.4 (2.9)	20.7 (2.9)	12.6 (2.5)
500	48.3 (0.2)	55 (0.7)	59.1 (1.1)	56.4 (2)	51.4 (2.7)	49.9 (0)	59.4 (0.5)	69 (0.8)	81.6 (1.1)	97.5 (1.6)
1000	49.4 (0.1)	55.5 (0.4)	58.9 (0.7)	62.2 (0.9)	66 (1.2)	50 (0)	54.4 (0.3)	57.5 (0.5)	63.3 (0.6)	69.7 (1)

Table A.8: Average number of selected variables 100 replications of Relaxed Lasso. Standard errors given in parentheses.

n \ p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100	7.6	8.7	10.6	21.3	31	6	6.6	9.3	11.5	12.6
	(0.5)	(0.9)	(1.2)	(3.5)	(4.3)	(0.3)	(0.5)	(1.7)	(2.3)	(2.6)
500	5.7	6.2	5.5	5.6	6.4	6.2	5.8	5.8	5.3	5.1
	(0.3)	(0.3)	(0.2)	(0.4)	(0.6)	(0.3)	(0.2)	(0.4)	(0.1)	(0)
1000	5.6	5.5	5.7	5.4	6.9	5.6	5.8	5.4	5.4	5.4
	(0.2)	(0.2)	(0.3)	(0.1)	(1.5)	(0.2)	(0.3)	(0.2)	(0.2)	(0.2)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100	19.2	24	33.9	44.2	52	22.9	26.5	45.6	91.1	90
	(1)	(1.5)	(3.5)	(4.9)	(5.5)	(0.8)	(1.2)	(3.6)	(5.2)	(5.1)
500	17.1	17.4	17.6	17.2	16.5	15.6	15.6	15.3	15.6	15.4
	(0.5)	(0.6)	(0.6)	(0.9)	(0.5)	(0.1)	(0.2)	(0.2)	(0.2)	(0.1)
1000	15.8	15.3	16.2	15.5	14.9	15.3	15.3	15.1	15.2	15.1
	(0.3)	(0.1)	(0.8)	(0.2)	(0.1)	(0.1)	(0.1)	(0)	(0.1)	(0)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100	30.5	32.2	34.9	40.1	27	47.9	58.2	90.1	76.4	69
	(1.3)	(2)	(3.5)	(4.9)	(4.3)	(0.3)	(1.7)	(4.5)	(5.3)	(5.2)
500	49.1	65.9	78.2	97.3	95.2	49.9	55.8	62.5	70.9	79.2
	(0.1)	(1.3)	(2.4)	(4.4)	(5.5)	(0)	(0.6)	(1.3)	(2.4)	(2.9)
1000	49.8	59.4	61.1	67.6	68.9	50	51.8	52	53.3	53.1
	(0)	(1.1)	(1.7)	(3)	(3.4)	(0)	(0.3)	(0.4)	(0.5)	(0.4)

Table A.9: Average false discovery rate from 100 replications of CV 1SE. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100	0.3 (0)	0.3 (0)	0.4 (0)	0.4 (0)	0.4 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.5 (0)	0.5 (0)
500	0.1 (0)	0.1 (0)	0.2 (0)	0.2 (0)	0.2 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0.2 (0)	0.2 (0)
1000	0 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100	0.3 (0)	0.3 (0)	0.3 (0)	0.3 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.6 (0)	0.6 (0)	0.6 (0)
500	0.2 (0)	0.3 (0)	0.4 (0)	0.4 (0)	0.5 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.4 (0)	0.5 (0)
1000	0.1 (0)	0.2 (0)	0.3 (0)	0.3 (0)	0.3 (0)	0.1 (0)	0.2 (0)	0.3 (0)	0.3 (0)	0.4 (0)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100	0 (0)	0.2 (0)	0.2 (0)	0.1 (0)	0.1 (0)	0 (0)	0.3 (0)	0.3 (0)	0.3 (0)	0.3 (0)
500	0 (0)	0.2 (0)	0.4 (0)	0.5 (0)	0.5 (0)	0 (0)	0.2 (0)	0.4 (0)	0.5 (0)	0.6 (0)
1000	0 (0)	0.2 (0)	0.3 (0)	0.5 (0)	0.5 (0)	0 (0)	0.2 (0)	0.4 (0)	0.5 (0)	0.6 (0)

Table A.10: Average false discovery rate for 100 replications of AR2 CV 1SE. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100		0.1 (0)	0.2 (0)	0.2 (0)	0.2 (0)	0.3 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0.2 (0)
500		0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
1000		0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100		0.2 (0)	0.4 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.3 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.5 (0)
500		0.1 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0.2 (0)	0 (0)	0 (0)	0 (0)	0.1 (0)	0.1 (0)
1000		0 (0)	0 (0)	0 (0)	0 (0)	0.1 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100		0 (0)	0.2 (0)	0.4 (0)	0.5 (0)	0.4 (0)	0 (0)	0.3 (0)	0.4 (0)	0.4 (0)	0.4 (0)
500		0 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0 (0)	0.1 (0)	0.2 (0)	0.4 (0)	0.5 (0)
1000		0 (0)	0.1 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0.2 (0)

Table A.11: Average false discovery rate of 100 replications for α-modified CV 1SE. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100	0.1 (0)	0.1 (0)	0.2 (0)	0.2 (0)	0.3 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0.2 (0)	0.2 (0)
500	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
1000	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100	0.2 (0)	0.3 (0)	0.2 (0)	0.2 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.5 (0)	0.5 (0)
500	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0 (0)	0 (0)	0 (0)	0.1 (0)	0.1 (0)
1000	0 (0)	0 (0)	0 (0)	0.1 (0)	0.1 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100	0 (0)	0.1 (0)	0.2 (0)	0.1 (0)	0.1 (0)	0 (0)	0.3 (0)	0.3 (0)	0.3 (0)	0.2 (0)
500	0 (0)	0.2 (0)	0.3 (0)	0.3 (0)	0.4 (0)	0 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.5 (0)
1000	0 (0)	0.1 (0)	0.2 (0)	0.3 (0)	0.3 (0)	0 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0.3 (0)

Table A.12: Average false discovery rate for 100 replications of Relaxed Lasso. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100		0.3 (0)	0.3 (0)	0.4 (0)	0.4 (0)	0.5 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0.2 (0)
500		0.1 (0)	0.1 (0)	0.1 (0)	0 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0 (0)	0 (0)
1000		0.1 (0)	0.1 (0)	0 (0)	0 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0 (0)	0 (0)	0 (0)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100		0.3 (0)	0.5 (0)	0.6 (0)	0.6 (0)	0.7 (0)	0.3 (0)	0.4 (0)	0.6 (0)	0.7 (0)	0.8 (0)
500		0.1 (0)	0.1 (0)	0.2 (0)	0.1 (0)	0.2 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
1000		0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100		0 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.4 (0)	0 (0)	0.3 (0)	0.6 (0)	0.6 (0)	0.6 (0)
500		0 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.5 (0)	0 (0)	0.1 (0)	0.2 (0)	0.3 (0)	0.4 (0)
1000		0 (0)	0.2 (0)	0.2 (0)	0.3 (0)	0.3 (0)	0 (0)	0 (0)	0 (0)	0.1 (0)	0.1 (0)

Table A.13: Average number of false negatives from 100 replications of CV 1SE. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100		0.6 (0.1)	0.8 (0.1)	0.9 (0.1)	1.1 (0.1)	1.5 (0.1)	0 (0)	0 (0)	0 (0)	0 (0)	0.1 (0)
500		0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
1000		0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100		5.3 (0.3)	6.8 (0.4)	9.8 (0.4)	11 (0.4)	12.4 (0.3)	0.4 (0.1)	1 (0.1)	1.8 (0.2)	3.5 (0.3)	5.2 (0.3)
500		0.1 (0)	0.4 (0.1)	0.3 (0.1)	0.6 (0.1)	0.6 (0.1)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
1000		0.1 (0)	0 (0)	0 (0)	0.1 (0)	0.1 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100		31.7 (1.3)	39.5 (1)	43.9 (0.8)	48.4 (0.4)	48.8 (0.3)	5 (0.4)	18.1 (1)	33.4 (1.1)	42 (0.8)	45.9 (0.6)
500		1.6 (0.2)	3.9 (0.2)	5.9 (0.3)	10.6 (0.5)	15.3 (0.5)	0 (0)	0.1 (0)	0.3 (0.1)	0.6 (0.1)	1.1 (0.1)
1000		0.4 (0.1)	0.8 (0.1)	1.3 (0.1)	2.4 (0.2)	3.4 (0.2)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)

Table A.14: Average number of false negatives for 100 replications of AR2 CV 1SE. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^{*}=5; SNR=1.25; \bar{\sigma}=5.19$					$p^{*}=5; SNR=5; \bar{\sigma}=2.59$				
100	1.1 (0.1)	1.4 (0.1)	1.4 (0.1)	1.8 (0.1)	1.9 (0.1)	0.1 (0)	0.1 (0)	0.1 (0)	0.2 (0.1)	0.2 (0.1)
500	0.1 (0)	0.2 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
1000	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n	$p^{*}=15; SNR=1.25; \bar{\sigma}=9.06$					$p^{*}=15; SNR=5; \bar{\sigma}=4.53$				
100	5.6 (0.3)	6.7 (0.4)	9.7 (0.3)	10 (0.4)	11.5 (0.3)	0.9 (0.1)	1.7 (0.2)	2.7 (0.2)	5 (0.3)	6.4 (0.3)
500	0.7 (0.1)	0.8 (0.1)	0.9 (0.1)	1.3 (0.1)	1.4 (0.1)	0 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)
1000	0.2 (0)	0.3 (0.1)	0.3 (0.1)	0.4 (0.1)	0.4 (0.1)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n	$p^{*}=50; SNR=1.25; \bar{\sigma}=16.6$					$p^{*}=50; SNR=5; \bar{\sigma}=8.3$				
100	23.6 (1.3)	32.7 (1.1)	39.7 (0.9)	45.4 (0.6)	47.4 (0.3)	4.2 (0.4)	17.2 (0.9)	30.7 (1)	40.8 (0.8)	44.8 (0.5)
500	1.8 (0.2)	4.7 (0.2)	7.6 (0.4)	12.4 (0.5)	17.3 (0.7)	0.2 (0)	0.4 (0.1)	0.7 (0.1)	1.5 (0.1)	2.3 (0.2)
1000	0.7 (0.1)	1.6 (0.1)	2.6 (0.2)	4.4 (0.2)	5.4 (0.2)	0 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0.2 (0)

Table A.15: Average number of false negatives of 100 replications for α-modified CV 1SE. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100		1.1 (0.1)	1.3 (0.1)	1.4 (0.1)	1.7 (0.1)	2 (0.2)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)
500		0 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
1000		0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100		6.6 (0.4)	8.6 (0.5)	11.3 (0.4)	12 (0.4)	12.8 (0.3)	0.8 (0.1)	1.4 (0.1)	2.5 (0.2)	4.4 (0.3)	6.1 (0.4)
500		0.6 (0.1)	0.7 (0.1)	0.9 (0.1)	1.4 (0.1)	1.5 (0.1)	0 (0)	0.1 (0)	0 (0)	0.1 (0)	0.1 (0)
1000		0.2 (0)	0.3 (0.1)	0.3 (0.1)	0.4 (0.1)	0.4 (0.1)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100		35.7 (1.5)	41.9 (1.1)	45 (0.9)	48.6 (0.4)	48.7 (0.4)	4.9 (0.5)	18.9 (1.1)	33.6 (1.3)	42.4 (0.9)	46.5 (0.6)
500		1.7 (0.2)	5.7 (0.3)	8.7 (0.4)	15.2 (0.6)	21.2 (0.8)	0.1 (0)	0.3 (0.1)	0.6 (0.1)	1.2 (0.1)	2 (0.2)
1000		0.7 (0.1)	1.5 (0.1)	2.8 (0.2)	4.7 (0.2)	5.9 (0.3)	0 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0.2 (0)

Table A.16: Average number of false negatives for 100 replications of Relaxed Lasso. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^{*}=5; SNR=1.25; \bar{\sigma}=5.19$					$p^{*}=5; SNR=5; \bar{\sigma}=2.59$				
100		0.6 (0.1)	0.8 (0.1)	0.9 (0.1)	1 (0.1)	1 (0.1)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)
500		0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
1000		0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n		$p^{*}=15; SNR=1.25; \bar{\sigma}=9.06$					$p^{*}=15; SNR=5; \bar{\sigma}=4.53$				
100		4 (0.3)	4.9 (0.3)	6.7 (0.4)	8 (0.4)	9.4 (0.4)	0.6 (0.1)	1.4 (0.1)	1.8 (0.2)	2.5 (0.3)	3.7 (0.3)
500		0.4 (0.1)	0.7 (0.1)	0.9 (0.1)	1.4 (0.1)	1.7 (0.1)	0 (0)	0 (0)	0.1 (0)	0.1 (0)	0.2 (0)
1000		0.1 (0)	0.2 (0.1)	0.3 (0.1)	0.3 (0.1)	0.5 (0.1)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n		$p^{*}=50; SNR=1.25; \bar{\sigma}=16.6$					$p^{*}=50; SNR=5; \bar{\sigma}=8.3$				
100		19.5 (1.3)	28.8 (1.1)	36.3 (1.1)	40.4 (1)	45.5 (0.6)	2.1 (0.3)	13 (0.8)	17.9 (1.1)	30.6 (1.1)	37.3 (0.8)
500		0.9 (0.1)	3.5 (0.3)	5.7 (0.4)	9.4 (0.6)	14.6 (0.8)	0.1 (0)	0.7 (0.1)	1.2 (0.1)	2.2 (0.2)	3.8 (0.3)
1000		0.2 (0)	1.5 (0.1)	3.2 (0.3)	5.2 (0.3)	6.9 (0.4)	0 (0)	0.3 (0.1)	0.5 (0.1)	0.8 (0.1)	1.2 (0.1)

Table A.17: Average prediction bias from 100 replications of CV 1SE. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100		27.7	31.4	32.5	36.3	38.4	14	15.1	16	18	18.3
		(0.7)	(0.8)	(0.9)	(0.9)	(1)	(0.4)	(0.3)	(0.4)	(0.5)	(0.4)
500		33.7	36.3	37.1	38.4	40.8	16.5	18.5	18.6	19.3	20.9
		(0.8)	(0.7)	(0.9)	(0.9)	(1)	(0.4)	(0.5)	(0.4)	(0.4)	(0.5)
1000		38	39.2	40.5	42.9	44.7	19.7	19.1	20.7	20.9	22.3
		(0.9)	(0.8)	(0.9)	(0.9)	(0.9)	(0.4)	(0.4)	(0.4)	(0.5)	(0.5)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100		62.5	70.8	81	84.1	89.5	30.8	36.1	40.6	48	54.5
		(1.3)	(1.5)	(1.6)	(1.3)	(1.6)	(0.5)	(0.6)	(0.8)	(1.2)	(1.7)
500		69.2	76.6	82	90.2	95.7	35.4	39.1	41.5	45.1	48.1
		(1)	(1.2)	(1.2)	(1.4)	(1.3)	(0.6)	(0.6)	(0.6)	(0.7)	(0.7)
1000		74.2	81	86.3	92.4	103.4	37.9	42.1	43.7	47.4	51.2
		(1.3)	(1.2)	(1.4)	(1.2)	(1.6)	(0.6)	(0.6)	(0.7)	(0.8)	(0.8)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100		135.4	154.6	164.7	179.4	176.8	64.1	87.7	121.5	148.3	159.3
		(2.8)	(3)	(2.8)	(2.1)	(2.2)	(0.8)	(2.3)	(3.5)	(3.9)	(3.4)
500		139.8	173.6	196.7	226.4	252.7	73.7	86.1	99.5	113.1	125.3
		(1.7)	(1.8)	(2)	(2.2)	(3)	(0.8)	(0.9)	(0.9)	(1.1)	(1.1)
1000		151.6	180.7	205.5	232	254.5	77.4	92.2	105.2	115.4	126.6
		(1.9)	(1.9)	(2)	(2)	(2.2)	(0.8)	(1)	(1.1)	(0.9)	(1.2)

Table A.18: Average prediction bias for 100 replications of AR2 CV 1SE. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100		27.2	29.1	30	33.2	35.2	13.9	14.2	14.5	15.8	16
		(0.8)	(0.9)	(0.8)	(1)	(1)	(0.5)	(0.4)	(0.5)	(0.5)	(0.6)
500		30	31.1	31	30.7	31	16.6	17.2	16.6	16.5	17.4
		(0.8)	(0.9)	(0.8)	(0.9)	(0.9)	(0.5)	(0.6)	(0.4)	(0.5)	(0.5)
1000		33.9	34.8	34.4	35	36.1	19.5	19.7	19.8	19.3	19.9
		(0.8)	(0.8)	(0.8)	(0.7)	(1)	(0.4)	(0.5)	(0.5)	(0.5)	(0.5)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100		63.9	70.2	80.5	81.6	85.9	31.5	37.2	41.5	48.6	53.3
		(1.1)	(1.2)	(1)	(1)	(1.2)	(0.6)	(0.7)	(0.8)	(1.1)	(1.4)
500		66.8	71.8	74.5	79.7	81.9	35	37	37.8	38.4	40
		(1.2)	(1.3)	(1.2)	(1.5)	(1.4)	(0.6)	(0.8)	(0.7)	(0.7)	(0.6)
1000		70.3	73.1	73.8	79.7	84.7	35.8	38.5	40.3	42.1	40.8
		(1.2)	(1.2)	(1.5)	(1.2)	(1.4)	(0.6)	(0.7)	(0.7)	(0.8)	(0.8)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100		131.2	148.5	160.6	172.4	173.5	65.8	85.6	107.1	132.5	141.9
		(2.1)	(1.8)	(1.6)	(1.7)	(1.5)	(0.8)	(1.5)	(2.7)	(3.5)	(3.1)
500		143.8	177.7	202.5	229.1	254	76.5	87.2	97.9	110.5	120.3
		(1.7)	(1.8)	(1.9)	(2)	(2.4)	(0.9)	(1)	(1)	(1.3)	(1.3)
1000		153.4	182.4	202.2	226.1	240.2	80.4	91.1	100.5	105.5	110.5
		(2)	(1.9)	(1.9)	(2.2)	(2.1)	(0.9)	(1)	(1.1)	(1.2)	(1.1)

Table A.19: Average number of prediction bias of 100 replications for α-modified CV 1SE. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100	26.2	28.8	30.2	32.9	35.9	12.6	13	13.2	14.6	14.6
	(0.8)	(0.9)	(0.8)	(1)	(1.1)	(0.4)	(0.3)	(0.4)	(0.5)	(0.5)
500	29.6	30.3	30.1	30.5	30.7	15	15.3	15.1	15.1	15.8
	(0.8)	(0.9)	(0.9)	(0.9)	(0.8)	(0.4)	(0.5)	(0.4)	(0.5)	(0.4)
1000	33.2	33.4	34.8	33.8	35.4	17.9	17.5	18	17.2	18.1
	(0.8)	(0.8)	(0.8)	(0.8)	(0.9)	(0.4)	(0.5)	(0.5)	(0.4)	(0.5)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100	66.6	76.1	86.4	87.5	91.1	30.4	35.6	40.2	46.4	52.1
	(1.4)	(1.6)	(1.5)	(1.3)	(1.4)	(0.5)	(0.6)	(0.8)	(1.1)	(1.5)
500	65.8	70.9	73.6	80.7	82.7	32.8	34.5	35.4	36.4	38.2
	(1.1)	(1.3)	(1.2)	(1.5)	(1.5)	(0.6)	(0.7)	(0.6)	(0.7)	(0.6)
1000	69.1	73.1	73.4	79.7	84	34	36.3	37.1	38.5	38.1
	(1.2)	(1.2)	(1.4)	(1.3)	(1.4)	(0.6)	(0.6)	(0.7)	(0.8)	(0.8)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100	151.6	167.5	174.8	182.9	180.3	66.5	89.6	118.5	144.6	158.4
	(2.6)	(2.4)	(2)	(1.7)	(1.6)	(0.9)	(2.4)	(3.6)	(4.2)	(3.9)
500	142.6	182.1	207.1	238.1	266.5	73.4	85.4	95.2	107.5	118.1
	(1.8)	(1.9)	(2.1)	(2.5)	(3.4)	(0.8)	(0.9)	(0.9)	(1.1)	(1.2)
1000	152.4	183	204.7	228.5	246.4	76.4	88.5	97.5	102.1	107.7
	(2)	(1.9)	(2)	(2.2)	(2.3)	(0.8)	(1)	(1.1)	(1.1)	(1.2)

Table A.20: Average prediction bias for 100 replications of Relaxed Lasso. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 5.19$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.59$				
100		20.4	23.8	26.2	30.6	34.9	8.2	8.8	9.9	10.5	11.2
		(0.6)	(0.8)	(0.8)	(1.1)	(1.2)	(0.4)	(0.4)	(0.5)	(0.6)	(0.7)
500		14.3	16.1	13.6	14.4	15.1	8.3	8.3	7.9	7.4	7.6
		(0.6)	(0.8)	(0.6)	(0.7)	(0.9)	(0.4)	(0.4)	(0.4)	(0.4)	(0.3)
1000		13.4	14	13.4	13.6	16.1	8	7.8	7.3	7	7.6
		(0.6)	(0.7)	(0.7)	(0.6)	(0.9)	(0.3)	(0.3)	(0.3)	(0.3)	(0.4)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 9.06$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.53$				
100		54.7	60.9	69.9	76.3	82.3	27.5	32.8	37.4	44.4	46.4
		(1)	(1)	(1.3)	(1.4)	(1.3)	(0.5)	(0.5)	(0.7)	(0.6)	(0.9)
500		45.3	51.6	57.7	60.3	66.6	20.5	20.6	20.4	22	22.6
		(1.1)	(1.3)	(1.3)	(1.5)	(1.4)	(0.5)	(0.6)	(0.5)	(0.7)	(0.7)
1000		39.6	42.1	42.8	47.3	50	19.1	20	19.2	20.6	19
		(1)	(1)	(1.5)	(1.3)	(1.4)	(0.4)	(0.5)	(0.4)	(0.5)	(0.4)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 16.6$					$p^* = 50; SNR = 5; \bar{\sigma} = 8.3$				
100		113.7	130.2	149.5	159.1	165.9	60	76	88.4	102	108.6
		(1.9)	(2.1)	(2.4)	(2.3)	(2.2)	(0.6)	(1.3)	(2)	(3)	(3.6)
500		118.4	153.1	176	203.8	225.1	61.2	71.9	82.9	96.7	111.8
		(1.4)	(1.5)	(1.6)	(1.6)	(2.1)	(0.6)	(0.9)	(0.9)	(1.1)	(1.2)
1000		116.9	149.5	172	198.3	218.3	58.7	67.2	71.2	77.9	80.9
		(1.4)	(1.6)	(1.8)	(2.3)	(2.3)	(0.6)	(0.9)	(1.1)	(1.3)	(1.4)

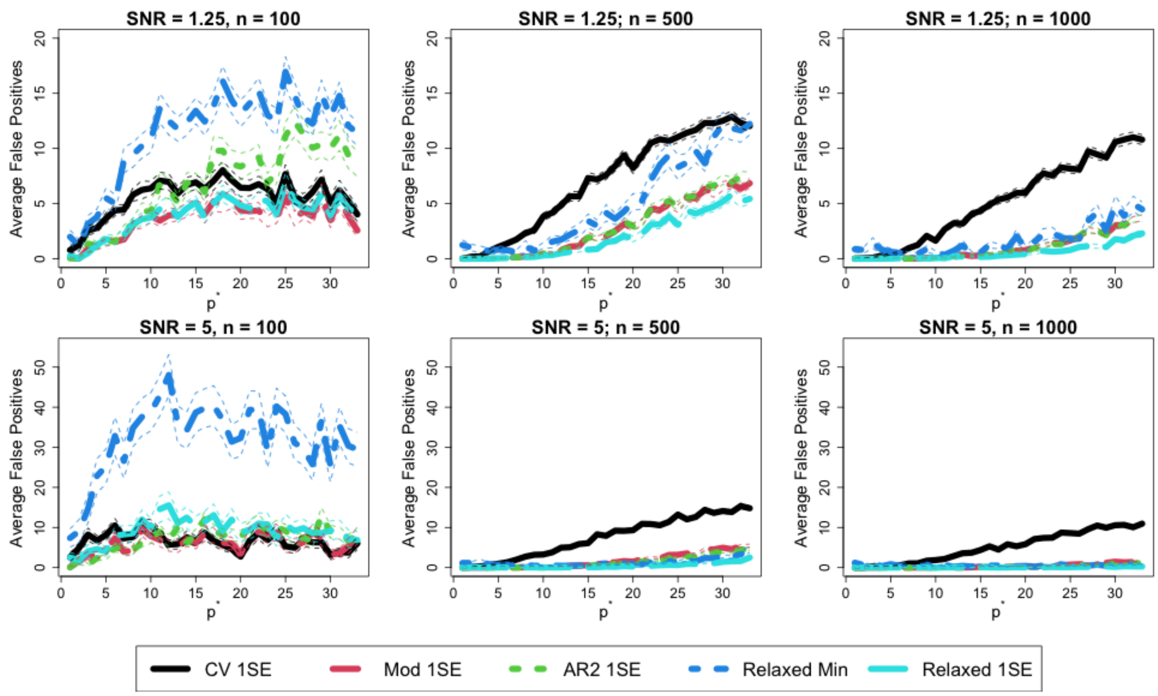

Figure A.1: Average false positives from 100 replications of simulated data with $n = 100, n = 500, n = 1000$ and $p = 100$ with independent predictors for both SNR $= 1.25$ and SNR $= 5$. Thin dotted lines represent the mean $\pm$ one standard error.

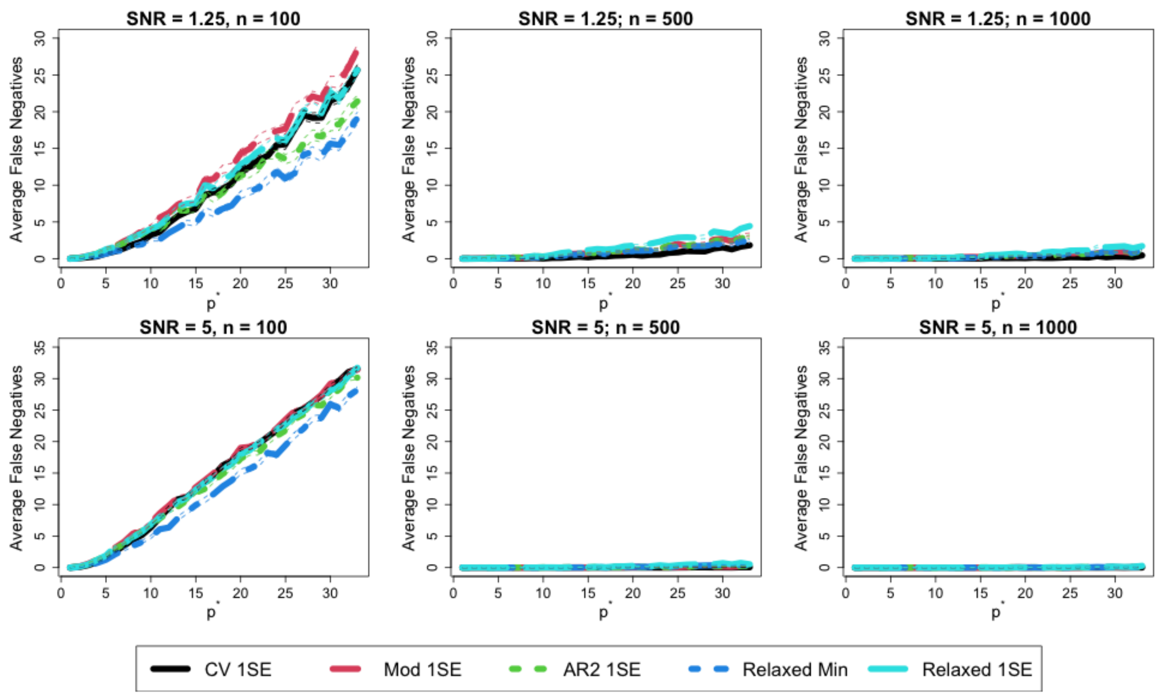

Figure A.2: Average false negatives from 100 replications of simulated data with $n = 100, n = 500, n = 1000$ and $p = 100$ with independent predictors for both SNR $= 1.25$ and SNR $= 5$. Thin dotted lines represent the mean $\pm$ one standard error.

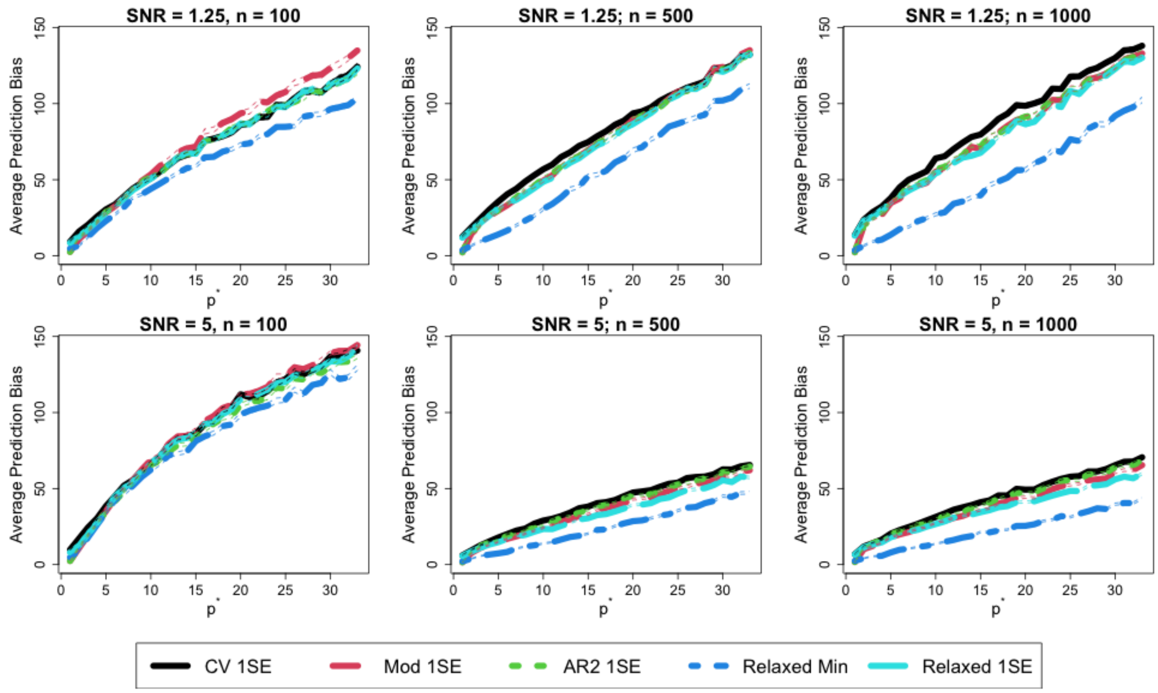

Figure A.3: Average prediction error plots from 100 replications of simulated data with $n = 100, n = 500, n = 1000$ and $p = 100$ with independent predictors for both SNR $= 1.25$ and SNR $= 5$. Thin dotted lines represent the mean $\pm$ one standard error.

Table A.21: Average false negatives for 100 replications of simulated data with $n = 100$. Standard errors given in parentheses.

	$p = 100$				$p = 800$			
	SNR = 1.25		SNR = 5		SNR = 1.25		SNR = 5	
	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$
CV Min	0.22 (0.05)	15.57 (0.67)	0 (0)	4.05 (0.28)	0.69 (0.09)	28.05 (0.49)	0.02 (0.01)	22.24 (0.62)
CV 1SE	0.71 (0.09)	25.67 (0.74)	0.01 (0.01)	7.71 (0.46)	1.45 (0.15)	31.54 (0.3)	0.04 (0.02)	29.41 (0.46)
Mod Min	0.48 (0.07)	20.78 (0.97)	0.01 (0.01)	4.75 (0.39)	1.11 (0.12)	28.87 (0.55)	0.03 (0.02)	24.36 (0.75)
Mod 1SE	1.4 (0.13)	28.42 (0.7)	0.09 (0.03)	8.74 (0.57)	1.96 (0.16)	31.42 (0.36)	0.14 (0.04)	29.52 (0.5)
AR2 Min	0.48 (0.08)	11.91 (0.82)	0.01 (0.01)	4.26 (0.4)	1.02 (0.11)	26.34 (0.48)	0.05 (0.03)	22.83 (0.6)
AR2 1SE	1.33 (0.12)	21.41 (0.82)	0.11 (0.03)	8.78 (0.56)	1.93 (0.14)	30.15 (0.31)	0.17 (0.04)	28.42 (0.43)
R2 Cut	0.08 (0.03)	4.47 (0.17)	0 (0)	3.01 (0.16)	0.37 (0.05)	23.04 (0.24)	0.02 (0.01)	19.21 (0.27)
AIC	0.29 (0.05)	20.28 (0.76)	0 (0)	4.19 (0.35)	0.94 (0.1)	29.86 (0.34)	0.02 (0.01)	26.44 (0.48)
BIC	0.73 (0.1)	28.39 (0.65)	0.01 (0.01)	7.71 (0.56)	1.54 (0.14)	31.84 (0.27)	0.06 (0.02)	30.34 (0.43)
ERIC	1.96 (0.16)	32.96 (0.03)	0.07 (0.03)	31.91 (0.55)	0.33 (0.05)	20.14 (0.27)	0.02 (0.01)	15.35 (0.26)
GCV	0.14 (0.03)	12.65 (0.68)	0 (0)	3.67 (0.27)	0.33 (0.05)	20.14 (0.27)	0.02 (0.01)	15.35 (0.26)
Relaxed Min	0.68 (0.09)	19.14 (0.81)	0.09 (0.03)	5.64 (0.48)	1.24 (0.12)	28.2 (0.48)	0.18 (0.05)	21.11 (0.69)
Relaxed 1SE	1.27 (0.12)	25.67 (0.76)	0.13 (0.04)	9.49 (0.62)	1.91 (0.14)	31.71 (0.3)	0.24 (0.05)	26.48 (0.67)

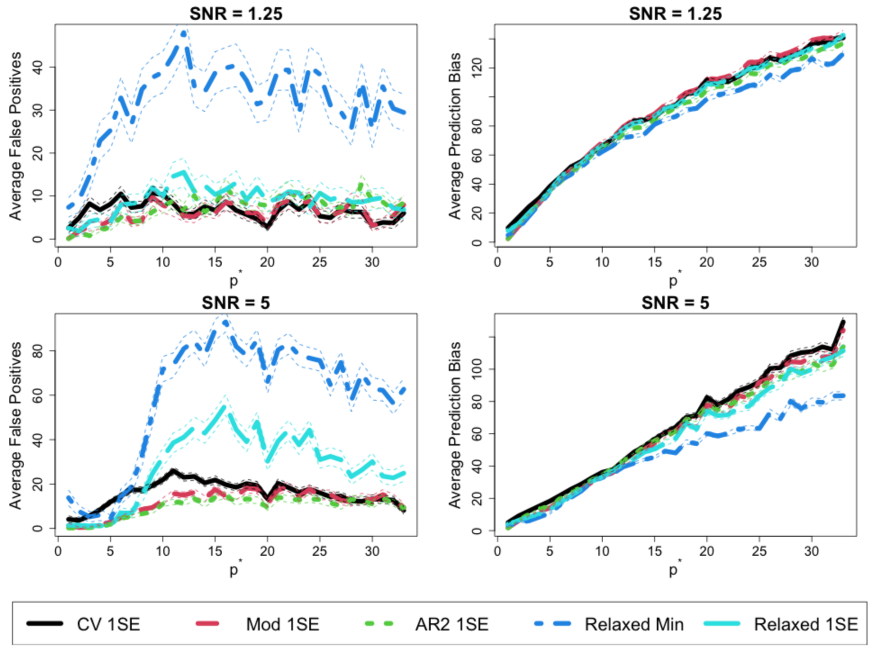

Figure A.4: Average false positives and prediction error plots from 100 replications of simulated data with $n = 100$ and $p = 800$ for both SNR $= 0.5$ and SNR $= 2$. Thin dotted lines represent the mean $\pm$ one standard error.

Table A.22: Average false discovery rate for 100 replications of simulated data with $n = 1000$. Standard errors given in parentheses.

	SNR $= 1.25$		SNR $= 5$	
	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$
CV Min	0.68 (0.01)	0.49 (0)	0.69 (0.01)	0.49 (0.01)
CV 1SE	0.06 (0.01)	0.24 (0.01)	0.06 (0.01)	0.24 (0.01)
Mod Min	0.37 (0.03)	0.36 (0.01)	0.26 (0.03)	0.32 (0.01)
Mod 1SE	0 (0)	0.1 (0.01)	0 (0)	0.05 (0)
AR2 Min	0.38 (0.03)	0.38 (0.01)	0.27 (0.03)	0.32 (0.01)
AR2 1SE	0 (0)	0.1 (0.01)	0 (0)	0.03 (0)
R2 Cut	0 (0)	0.09 (0.01)	0 (0)	0.05 (0.01)
AIC	0.68 (0.01)	0.5 (0.01)	0.69 (0.01)	0.49 (0.01)
BIC	0.2 (0.01)	0.22 (0.01)	0.2 (0.02)	0.22 (0.01)
ERIC	0.07 (0.01)	0.11 (0.01)	0.08 (0.01)	0.12 (0.01)
GCV	0.7 (0.01)	0.5 (0.01)	0.73 (0.01)	0.5 (0.01)
Relaxed Min	0.04 (0.01)	0.1 (0.01)	0.06 (0.02)	0.02 (0)
Relaxed 1SE	0 (0)	0.06 (0.01)	0 (0)	0.01 (0)

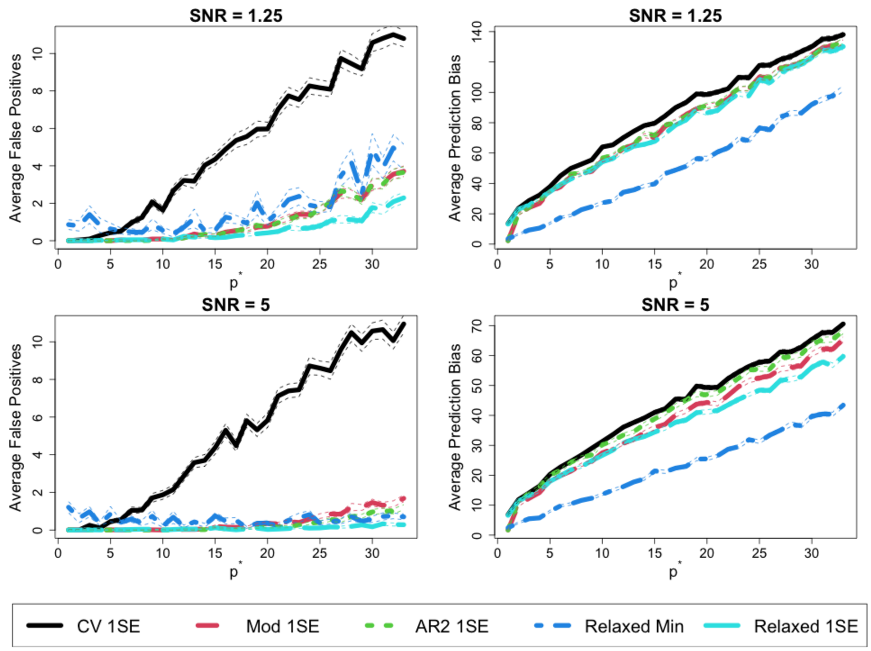

Figure A.5: Average false positives and prediction error plots from 100 replications of simulated data with $n = 1000$ and $p = 100$ for both SNR $= 0.5$ and SNR $= 2$. Thin dotted lines represent the mean $\pm$ one standard error.

Table A.23: Average prediction bias for 100 replications of simulated data with $n = 1000$. Standard errors given in parentheses.

	SNR $= 1.25$		SNR $= 5$	
	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$
CV Min	23.87 (0.67)	111.79 (1.23)	12.67 (0.34)	56.79 (0.64)
CV 1SE	37.85 (0.96)	138.02 (1.5)	20.31 (0.56)	70.56 (0.77)
Mod Min	18.13 (0.63)	110.67 (1.3)	9.14 (0.32)	51.99 (0.66)
Mod 1SE	34.64 (0.96)	132.97 (1.54)	18.08 (0.54)	65.4 (0.89)
AR2 Min	18.35 (0.63)	110.69 (1.29)	9.14 (0.32)	52.05 (0.66)
AR2 1SE	35.3 (0.99)	133.79 (1.56)	19.99 (0.54)	68.21 (0.91)
R2 Cut	77.07 (2.09)	187.98 (2.28)	50.31 (1.82)	112.37 (1.51)
AIC	24.12 (0.64)	112.62 (1.23)	12.98 (0.35)	57.37 (0.65)
BIC	29.12 (0.83)	136.75 (1.81)	15.13 (0.42)	70.12 (0.87)
ERIC	32.23 (0.91)	164.4 (2.56)	16.47 (0.43)	81.64 (1.11)
GCV	24.46 (0.66)	112.66 (1.24)	13.16 (0.36)	57.28 (0.64)
Relaxed Min	13.75 (0.61)	102.47 (1.77)	7.96 (0.34)	43.42 (0.75)
Relaxed 1SE	34.52 (0.96)	130.12 (1.81)	17.97 (0.47)	59.71 (0.84)

Table A.24: Average false negatives for 100 replications of simulated data with $n = 1000$. Standard errors given in parentheses.

	SNR = 1.25		SNR = 5	
	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$
CV Min	0 (0)	0.09 (0.03)	0 (0)	0.01 (0.01)
CV 1SE	0.01 (0.01)	0.45 (0.06)	0 (0)	0.02 (0.01)
Mod Min	0.01 (0.01)	0.29 (0.05)	0 (0)	0.02 (0.01)
Mod 1SE	0.06 (0.02)	1.05 (0.1)	0 (0)	0.07 (0.03)
AR2 Min	0.01 (0.01)	0.24 (0.05)	0 (0)	0.02 (0.01)
AR2 1SE	0.08 (0.03)	1.1 (0.11)	0 (0)	0.1 (0.03)
R2 Cut	0.07 (0.03)	1.2 (0.12)	0 (0)	0.1 (0.04)
AIC	0 (0)	0.09 (0.03)	0 (0)	0.01 (0.01)
BIC	0.01 (0.01)	0.47 (0.07)	0 (0)	0.02 (0.01)
ERIC	0.01 (0.01)	1.01 (0.11)	0 (0)	0.03 (0.02)
GCV	0 (0)	0.07 (0.03)	0 (0)	0.01 (0.01)
Relaxed Min	0.01 (0.01)	1.17 (0.12)	0 (0)	0.18 (0.05)
Relaxed 1SE	0.05 (0.02)	1.71 (0.16)	0 (0)	0.27 (0.05)

A.1.2 Simulation Study for Non-Convex Penalties

This set of simulations examines the effect of the modification on the non-convex penalties, SCAD (Fan and Li 2001) and MCP (Zhang 2010). Both penalties are most commonly expressed in terms of their derivatives which, for SCAD at a single point, β_j, is:

$$P'_\lambda(\beta_j) = \lambda_1\Big[I(\beta_j \leq \lambda_1) + \frac{(\lambda_2\beta_j - \beta_j)_+}{(\lambda_2 - 1)\beta_j} I(\beta_j > \lambda_1)\Big] \,. \tag{A.1}$$

The derivative of the MC+ penalty for a single coefficient, β_j, is:

$$P'_\lambda(\beta_j) = \begin{cases} \text{sign}(\beta_j)\Big(\lambda_1 - \frac{|\beta_j|}{\lambda_2}\Big) & \text{if } |\beta_j| \leq \lambda_1\lambda_2 \\ 0 & \text{otherwise} \end{cases} \,. \tag{A.2}$$

We investigated SCAD with $\lambda_2 = 3.7$ and MC+ with $\lambda_2 = 3$, both with and without the α-Modification using 10-fold CV with a 1SE rule. We generated data from the following model:

$$\mathbf{y}_{n\times 1} = \beta_0^* + \mathbf{X}_{n\times p}\boldsymbol{\beta}^*_{p\times 1} + \boldsymbol{\epsilon}_{n\times 1} \tag{A.3}$$

where $\mathbf{X}$ was drawn from a Standard Normal distribution, β_0^* and $\boldsymbol{\beta}^*$ were drawn from the same distribution as the Lasso simulations above, and SNR $= \{1.25, 5\}$. We include an intercept here because the `R` package `ncvreg` does not include a no-intercept option. The optimization to find $\hat{\alpha}_\lambda$ is slightly more complicated with an intercept term. We now optimize the likelihood function over both α and β_0, the intercept. Figures A.6 and A.7 include the average False Positives, False Negatives, and Prediction Bias across increasing values of p^* when $n = 100$ and $p = 100$ over 500 replications. Note the similarity between the two approaches.

Although average false positives for the non-convex penalties are lower than or equal to that of the α-Modified Lasso, their average false negatives tend to be higher, particularly as p^* increases. Prediction bias is slightly worse for α-Modified SCAD and MC+ than unmodified, but similar to the α-Modified Lasso for small p^* and similar to unmodified for larger p^*. This is a fairly limited simulation study, capturing only one n and p. Furthermore, for non-convex penalties, we no longer have a guarantee that $\hat{\alpha}_\lambda$ will always be greater than 1, so the bias-reduction aim of the modification is in question. Allowing the α-Modification to compete with and enhance non-convex penalties may be the subject of future research.

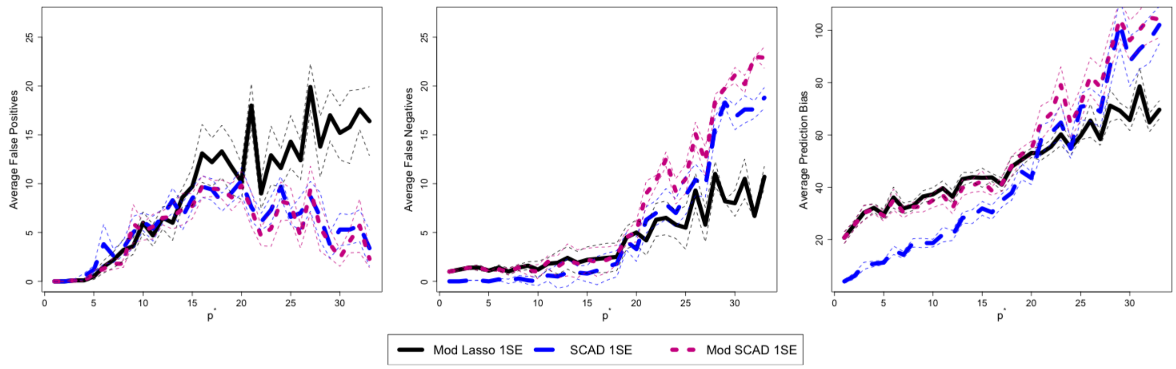

Figure A.6: Average false positives, prediction error, and false negatives from 500 replications of simulated data with $n = 100$ and $p = 100$ for SNR = 5. Thin dotted lines represent the mean $\pm$ one standard error.

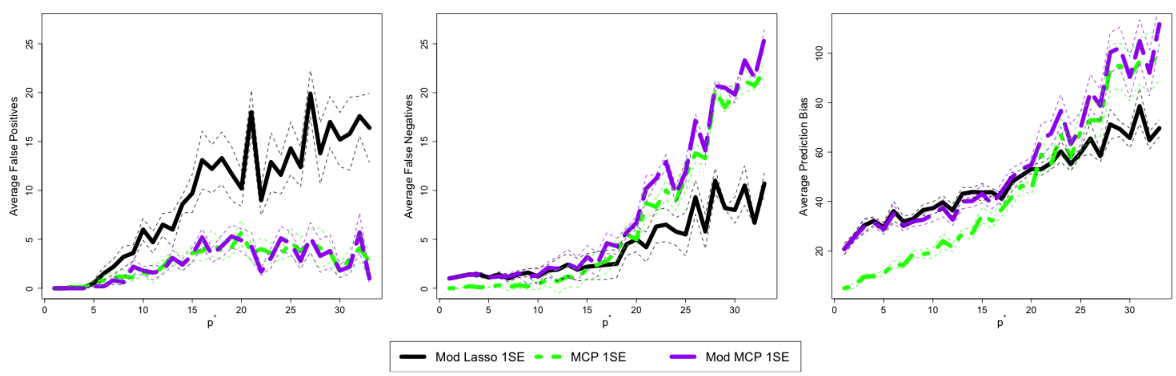

Figure A.7: Average false positives, prediction error, and false negatives from 500 replications of simulated data with $n = 100$ and $p = 100$ for SNR = 5. Thin dotted lines represent the mean $\pm$ one standard error.

A.1.3 Simulation Study for Correlated Predictors

Additional results from the simulation study described in the main document for correlated predictors can be found in Tables A.33 through A.44. These results include average model size, average FDR, average false negatives, and average prediction bias for traditional CV, α-modified and Average R^2 CV, and the Relaxed Lasso using a minimum APE rule.

Table A.25: Average Hamming Distance for 100 replications of CV 1SE with correlated predictors. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100	5.1	6.3	8.8	11.4	11.5	3.8	5.8	9.8	13.4	14.9
	(0.4)	(0.4)	(0.8)	(1.3)	(1.1)	(0.3)	(0.4)	(1.1)	(1.4)	(1.2)
500	1.9	3.2	4.4	5.1	7.6	1.8	2.9	4.7	4.9	7.2
	(0.2)	(0.4)	(0.4)	(0.6)	(0.6)	(0.2)	(0.3)	(0.4)	(0.4)	(0.7)
1000	1.5	1.9	3	4.1	4.3	1.2	1.4	2.9	4.4	4.3
	(0.2)	(0.2)	(0.4)	(0.4)	(0.5)	(0.2)	(0.2)	(0.4)	(0.5)	(0.5)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100	12.4	16.2	18.5	22	22.7	11	15.5	21.6	24.7	30.2
	(0.3)	(0.6)	(0.7)	(1.3)	(1.5)	(0.4)	(0.6)	(1.1)	(1.5)	(1.7)
500	7.1	10.1	12.6	15	18.9	5.1	7.4	12.4	16.2	20.2
	(0.4)	(0.4)	(0.6)	(0.6)	(1)	(0.3)	(0.4)	(0.6)	(0.7)	(0.9)
1000	4.6	6.8	9.4	12.7	14.7	3.5	5.1	9.3	11.5	14.2
	(0.3)	(0.4)	(0.6)	(0.7)	(0.8)	(0.3)	(0.3)	(0.5)	(0.7)	(0.9)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100	39.3	46.7	50.8	53.7	57.5	20.8	39.3	48.6	55.8	59.2
	(0.8)	(0.4)	(0.4)	(0.8)	(1.2)	(1.4)	(0.6)	(0.6)	(1.1)	(1.2)
500	13.5	28.2	40.2	48.4	55.3	3.1	19.5	35.8	56.5	74.3
	(1.3)	(1)	(0.9)	(0.9)	(1.5)	(0.6)	(0.7)	(0.8)	(1.7)	(2.2)
1000	8.4	21.4	33.1	42.3	54.3	1.4	15.3	29.2	43.2	60.3
	(1.1)	(1)	(0.9)	(1.1)	(1.3)	(0.4)	(0.5)	(0.8)	(1.1)	(1.6)

Figures A.8, A.9, and A.10 give the average false positives, false negatives, and prediction bias for the simulated data provided in the main document in Figure 2.6. Similar to independent predictors, larger sample sizes maintain low false negatives, but for $n = 100$ and $n = 500$ with the smaller SNR, average false negatives increase with p^*.

Finally, we evaluate the methods for limited data settings, letting $n = 100$, $p = \{100, 800\}$, SNR $= \{1.25, 5\}$, and varying p^*. Once again, $\mathbf{X}$ contained correlated columns

Table A.26: Average Hamming Distance for 100 replications of AR2 CV 1SE with correlated predictors. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100	4.3	4.7	5.8	7.1	6.8	2.3	3.1	5	5.5	7.1
	(0.3)	(0.4)	(0.4)	(0.8)	(0.4)	(0.2)	(0.3)	(0.5)	(0.6)	(0.6)
500	1.3	1.7	2.4	2.3	4	0.7	1	1.7	1.8	2
	(0.1)	(0.2)	(0.3)	(0.3)	(0.5)	(0.1)	(0.2)	(0.2)	(0.3)	(0.3)
1000	0.8	1	1.3	1.5	1.5	0.3	0.6	1	1.2	1.8
	(0.1)	(0.2)	(0.2)	(0.2)	(0.3)	(0.1)	(0.1)	(0.2)	(0.2)	(0.4)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100	12.7	15.5	17.4	17.6	18	9.6	13.4	18.3	19	22.2
	(0.3)	(0.5)	(0.8)	(0.8)	(0.6)	(0.4)	(0.5)	(0.9)	(1.1)	(1.3)
500	6.6	8.3	9.2	11.1	13.7	3.7	4.5	6.6	8.8	11.4
	(0.4)	(0.5)	(0.5)	(0.5)	(0.7)	(0.3)	(0.4)	(0.5)	(0.6)	(0.9)
1000	3.4	4.9	6.4	8.1	9.9	1.6	2.5	4.4	5.9	7.4
	(0.3)	(0.4)	(0.5)	(0.6)	(0.7)	(0.2)	(0.3)	(0.5)	(0.6)	(0.7)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100	40.4	46.5	50.8	53.1	54.6	23.4	39.8	48.5	53	54.3
	(1.1)	(0.5)	(0.5)	(0.8)	(1)	(1.7)	(0.7)	(0.5)	(0.8)	(0.8)
500	14.4	28.6	38.7	47.1	52.4	4.2	17.3	29.5	43.6	61.1
	(1.5)	(1.2)	(0.9)	(1)	(1.3)	(0.8)	(0.8)	(1)	(1.1)	(1.9)
1000	9.5	20.5	29.6	36.4	47.2	1.7	11.6	19.9	29.5	40.6
	(1.2)	(1.1)	(1.1)	(1.1)	(1.5)	(0.5)	(0.7)	(1)	(1.3)	(1.6)

and more methods are evaluated in Figure A.11 and Tables A.45, A.46, and A.47. Figure A.12 and Tables A.48 through A.50 give results from the case where $n = 1000$. Average false negatives are higher for all methods when predictors are correlated. Particularly for large p^* all methods struggle to capture the true model, though the Relaxed Lasso seemingly performs the best. For the $n = 1000$ case, Mod 1SE, AR2 1SE, and Relaxed 1SE all have the smallest average FDRs so it seems that the new methods are most competitive with the Relaxed Lasso when sample sizes are large.

Table A.27: Average Hamming Distance for 100 replications of α-modified CV 1SE with correlated predictors. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100	4.3	4.7	6.7	8.7	9.7	2.7	3.6	7.3	7.6	10.3
	(0.3)	(0.3)	(0.8)	(1.3)	(1.4)	(0.2)	(0.3)	(1.1)	(0.9)	(1)
500	1.1	1.7	2.2	2.5	3.5	1.1	1.1	2.4	2.3	2.6
	(0.1)	(0.2)	(0.2)	(0.3)	(0.4)	(0.1)	(0.2)	(0.3)	(0.3)	(0.3)
1000	0.7	1.1	1.3	1.8	1.6	0.6	0.8	1.3	1.6	2.3
	(0.1)	(0.2)	(0.2)	(0.3)	(0.2)	(0.1)	(0.1)	(0.2)	(0.3)	(0.4)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100	12.6	15.4	16.4	20.4	20.9	10.1	14.2	19.8	21.7	28.6
	(0.3)	(0.5)	(0.6)	(1.5)	(1.7)	(0.4)	(0.5)	(1.1)	(1.5)	(1.9)
500	6.4	8.1	9.2	11.1	12.3	4	5.2	7.8	10.6	13
	(0.4)	(0.4)	(0.5)	(0.5)	(0.6)	(0.3)	(0.4)	(0.6)	(0.7)	(1)
1000	3.4	4.7	6.5	8.1	10	2.2	3.1	5.8	7.3	8.6
	(0.3)	(0.4)	(0.5)	(0.6)	(0.7)	(0.2)	(0.3)	(0.5)	(0.7)	(0.7)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100	42.9	48	50.5	53.6	56.5	22.1	40.4	49	56.4	58.7
	(1)	(0.3)	(0.4)	(1)	(1.6)	(1.6)	(0.7)	(0.6)	(1.4)	(1.6)
500	14.7	28.9	37.7	44.5	49	3.3	17.7	30.6	47.2	64.3
	(1.5)	(1.2)	(1)	(0.9)	(1)	(0.6)	(0.8)	(0.9)	(1.2)	(1.9)
1000	9.3	20.4	29.1	34.4	41.6	1.4	12.2	21.4	32.1	45.2
	(1.3)	(1.2)	(1.1)	(1.2)	(0.9)	(0.4)	(0.7)	(1)	(1.3)	(1.8)

Table A.28: Average Hamming Distance for 100 replications of Relaxed Lasso with correlated predictors. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100		6.3	9	14.2	19.9	35.9	3.6	6.1	15.1	21.8	31.8
		(0.5)	(0.9)	(2)	(3.3)	(5.2)	(0.4)	(0.8)	(2.6)	(3.7)	(4.9)
500		2.8	3.9	4.8	5.6	8.8	1.7	1.6	2.7	3.3	4
		(0.4)	(0.6)	(0.6)	(0.9)	(1.7)	(0.3)	(0.4)	(0.5)	(0.6)	(1.2)
1000		1.9	2.2	3.7	4	6.3	1.1	1.4	2	2.4	1.5
		(0.4)	(0.4)	(0.8)	(0.7)	(1.8)	(0.3)	(0.4)	(0.5)	(0.7)	(0.4)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100		13.9	21.5	27.6	37.6	59.2	12.9	19.1	30.1	49	66.5
		(0.5)	(0.9)	(2.4)	(3.7)	(6)	(0.6)	(1.1)	(2.3)	(4.4)	(5.6)
500		8.9	11.6	13.5	14.9	21	4	5	7.3	8.3	9.7
		(0.6)	(1)	(1.1)	(1.4)	(2)	(0.5)	(0.7)	(1.1)	(1.4)	(1.2)
1000		4.7	7.7	11.6	12.3	13.7	1.5	3	4.6	5.4	5.7
		(0.5)	(0.9)	(1.5)	(1.5)	(1.6)	(0.3)	(0.7)	(0.8)	(1)	(1.1)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100		33.6	44.6	56.6	67.3	78.3	14.8	38.8	63.6	83.5	97.5
		(1.2)	(0.5)	(1.4)	(2.9)	(4.1)	(1.4)	(0.6)	(2.6)	(4.1)	(5.3)
500		5.7	30.1	48.3	62.5	74.8	0.9	18.1	30.2	51.1	68.8
		(0.9)	(0.9)	(1.4)	(2.2)	(3.1)	(0.3)	(1.2)	(1.9)	(3.1)	(4.2)
1000		2.1	23.2	38.8	46.1	63.9	0.3	12.2	15.6	22.4	31.7
		(0.5)	(1.2)	(1.9)	(2.9)	(3.8)	(0.2)	(1.2)	(1.6)	(2.7)	(3.8)

Table A.29: Average number of selected variables 100 replications of CV 1SE with correlated predictors. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100	6.5	7.4	9.1	11.2	10.7	8	10	13.7	16.8	18.2
	(0.4)	(0.4)	(0.9)	(1.4)	(1.2)	(0.3)	(0.4)	(1.1)	(1.4)	(1.2)
500	6.3	7.4	8.4	8.9	11.3	6.7	7.8	9.5	9.7	11.9
	(0.2)	(0.3)	(0.4)	(0.5)	(0.6)	(0.2)	(0.3)	(0.4)	(0.4)	(0.7)
1000	6.1	6.5	7.4	8.4	8.6	6.2	6.3	7.8	9.4	9.2
	(0.2)	(0.2)	(0.3)	(0.4)	(0.5)	(0.2)	(0.2)	(0.4)	(0.5)	(0.5)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100	10.5	11.4	9.2	11.9	11.2	19.2	22.1	25.3	24	28.3
	(0.6)	(1)	(1)	(1.6)	(1.7)	(0.7)	(0.9)	(1.5)	(1.9)	(2.1)
500	15.6	19.2	20.2	22.1	24.3	18.8	21.1	25.4	29.4	33.2
	(0.4)	(0.6)	(0.7)	(0.8)	(1.1)	(0.3)	(0.4)	(0.6)	(0.8)	(1)
1000	17.2	18.7	20.7	23	24.6	18.3	19.6	23.8	25.7	28.4
	(0.3)	(0.4)	(0.5)	(0.7)	(0.7)	(0.3)	(0.3)	(0.5)	(0.6)	(0.8)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100	10.7	9	8.7	8.6	11.2	29.2	27.6	21.4	20	18.4
	(0.8)	(0.9)	(1)	(1.1)	(1.7)	(1.4)	(1.9)	(1.8)	(2.1)	(1.8)
500	36.5	40.9	45.1	41.6	42.1	46.9	59.1	70.6	88.4	103.4
	(1.3)	(1.8)	(2.4)	(2.7)	(3.1)	(0.6)	(1.3)	(2.1)	(3.2)	(4.1)
1000	41.6	49.3	56.3	60.7	66.5	48.6	61.6	74.7	87.3	102.5
	(1.1)	(1.7)	(2.3)	(2.5)	(3.2)	(0.4)	(0.7)	(1.2)	(1.6)	(2.3)

Table A.30: Average number of selected variables 100 replications of AR2 CV 1SE with correlated predictors. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100		4.6	4.5	5	6.1	5.3	5.7	6.5	7.9	7.8	9.3
		(0.4)	(0.5)	(0.5)	(0.9)	(0.5)	(0.3)	(0.4)	(0.5)	(0.6)	(0.7)
500		4.9	5.2	5.5	5.1	6.9	5.3	5.6	6.3	6.4	6.4
		(0.2)	(0.2)	(0.4)	(0.3)	(0.5)	(0.1)	(0.2)	(0.3)	(0.3)	(0.3)
1000		5	5.3	5.2	5.4	5.3	5.2	5.5	5.8	6	6.7
		(0.1)	(0.2)	(0.2)	(0.3)	(0.3)	(0.1)	(0.1)	(0.2)	(0.2)	(0.4)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100		9	9.1	7.6	6.3	6	15.7	17.6	19.7	15.1	17.5
		(0.8)	(1)	(1.2)	(1.1)	(0.8)	(0.8)	(1)	(1.5)	(1.6)	(1.9)
500		13.1	15.4	14.6	16.6	17.2	16.3	17.6	18.6	21.4	23.8
		(0.5)	(0.7)	(0.7)	(0.8)	(1.2)	(0.4)	(0.4)	(0.6)	(0.7)	(0.9)
1000		14.9	15.4	16.3	16.7	18.2	16.2	16.8	18.3	19.6	21.1
		(0.3)	(0.5)	(0.6)	(0.7)	(0.9)	(0.2)	(0.3)	(0.5)	(0.6)	(0.7)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100		9.6	9.2	9.5	7.6	7.6	26.6	25.4	18.3	12.3	10.2
		(1.1)	(1.3)	(1.4)	(1.3)	(1.5)	(1.7)	(2)	(2)	(1.9)	(1.4)
500		35.6	38.2	39.7	37.4	36.2	45.8	54.6	61.6	72.8	86.7
		(1.5)	(2.1)	(2.6)	(3)	(3.3)	(0.8)	(1.3)	(2.1)	(2.7)	(4)
1000		40.5	45.8	49	51.3	56.1	48.3	56.6	64.3	72.3	81.2
		(1.2)	(1.8)	(2.3)	(2.4)	(3.3)	(0.5)	(0.8)	(1.1)	(1.6)	(1.9)

Table A.31: Average number of selected variables 100 replications of α-modified CV 1SE with correlated predictors. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100		4.6	4.5	5.7	7.6	8	6.4	7.2	10.7	10.4	12.8
		(0.4)	(0.4)	(0.9)	(1.4)	(1.5)	(0.3)	(0.3)	(1.1)	(0.9)	(1.1)
500		4.9	5.4	5.2	5.3	6.2	5.9	5.9	7.2	6.9	7.1
		(0.2)	(0.2)	(0.3)	(0.3)	(0.5)	(0.2)	(0.2)	(0.3)	(0.4)	(0.3)
1000		5	5.5	5.2	5.6	5.4	5.5	5.7	6.2	6.6	7.1
		(0.1)	(0.2)	(0.2)	(0.3)	(0.3)	(0.1)	(0.1)	(0.2)	(0.3)	(0.4)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100		7.6	7.8	4.8	9.3	8.5	16.9	19.7	22	18.9	25.2
		(0.7)	(1)	(0.9)	(2)	(1.9)	(0.8)	(1)	(1.7)	(2)	(2.5)
500		13.4	15.1	14.7	16.4	15.4	17.4	18.6	20	23.5	25.4
		(0.6)	(0.7)	(0.8)	(0.8)	(0.9)	(0.3)	(0.5)	(0.7)	(0.7)	(1)
1000		15.5	15.4	16.3	16.6	18.4	16.9	17.6	20	21.3	22.5
		(0.3)	(0.5)	(0.6)	(0.8)	(0.8)	(0.3)	(0.3)	(0.5)	(0.7)	(0.8)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100		7.1	5	6	7.4	9.4	27.9	24.7	18.6	18.6	16.1
		(1)	(0.8)	(1.2)	(1.7)	(2.2)	(1.6)	(2.1)	(2.2)	(2.7)	(2.4)
500		35.3	36.2	35.4	28.1	25.2	46.7	56.3	64.2	77.4	90.5
		(1.5)	(2)	(2.4)	(2.6)	(2.9)	(0.6)	(1.3)	(2)	(2.8)	(4.1)
1000		40.7	45.2	48.1	47.5	47.4	48.5	58.2	66.7	75.7	86.6
		(1.3)	(1.9)	(2.3)	(2.3)	(2.8)	(0.4)	(0.7)	(1)	(1.5)	(2.1)

Table A.32: Average number of selected variables 100 replications of Relaxed Lasso with correlated predictors. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100		8.3	10.6	14.8	20.3	36.1	7.9	10.4	19.1	25.3	35.3
		(0.6)	(0.9)	(2.1)	(3.4)	(5.3)	(0.5)	(0.8)	(2.6)	(3.7)	(4.9)
500		7.6	8.4	9.1	9.7	12.7	6.7	6.6	7.6	8.2	8.7
		(0.4)	(0.6)	(0.6)	(0.9)	(1.7)	(0.3)	(0.4)	(0.5)	(0.6)	(1.2)
1000		6.8	7.1	8.4	8.6	10.8	6.1	6.4	6.9	7.4	6.4
		(0.4)	(0.4)	(0.8)	(0.7)	(1.8)	(0.3)	(0.4)	(0.5)	(0.7)	(0.4)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100		15.9	19.8	23.1	31.1	51.8	22.6	27.3	35.3	51.3	67.6
		(1)	(1.5)	(2.9)	(4.2)	(6.5)	(0.9)	(1.3)	(2.7)	(4.9)	(6.2)
500		19.4	21.9	22	22.5	27.5	18	18.9	20.6	21.5	22.2
		(0.7)	(1)	(1.2)	(1.5)	(1.9)	(0.5)	(0.7)	(1)	(1.3)	(1.1)
1000		17.9	20.9	23.7	23.4	24	16.4	17.7	19.2	19.9	19.9
		(0.5)	(0.9)	(1.6)	(1.4)	(1.6)	(0.3)	(0.6)	(0.8)	(0.9)	(1)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100		16.4	17.4	24.8	30.5	37.4	35.2	40.9	56.2	63	68
		(1.2)	(1.5)	(2.9)	(4.2)	(5.1)	(1.4)	(2.1)	(5)	(6)	(6.7)
500		44.3	54.3	62.5	65.2	70.7	49.1	60.1	67.4	84.5	98.8
		(0.9)	(2)	(3)	(4.1)	(5)	(0.3)	(1.3)	(2)	(3.6)	(4.9)
1000		47.9	60.1	70	69.8	81.7	49.7	59.4	60.2	64.5	71.9
		(0.5)	(1.5)	(2.5)	(3.1)	(4.5)	(0.2)	(1.2)	(1.5)	(2.6)	(3.6)

Table A.33: Average false discovery rate from 100 replications of CV 1SE with correlated predictors. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100	0.4 (0)	0.5 (0)	0.6 (0)	0.6 (0)	0.7 (0)	0.4 (0)	0.5 (0)	0.6 (0)	0.6 (0)	0.7 (0)
500	0.2 (0)	0.3 (0)	0.4 (0)	0.4 (0)	0.5 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.4 (0)	0.5 (0)
1000	0.2 (0)	0.2 (0)	0.3 (0)	0.3 (0)	0.3 (0)	0.1 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.3 (0)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100	0.3 (0)	0.5 (0)	0.6 (0)	0.7 (0)	0.6 (0)	0.4 (0)	0.5 (0)	0.6 (0)	0.6 (0)	0.7 (0)
500	0.2 (0)	0.4 (0)	0.4 (0)	0.5 (0)	0.5 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.6 (0)
1000	0.2 (0)	0.3 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.2 (0)	0.2 (0)	0.4 (0)	0.4 (0)	0.4 (0)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100	0 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.6 (0)	0 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.6 (0)
500	0 (0)	0.2 (0)	0.4 (0)	0.5 (0)	0.5 (0)	0 (0)	0.2 (0)	0.4 (0)	0.5 (0)	0.6 (0)
1000	0 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0 (0)	0.2 (0)	0.4 (0)	0.5 (0)	0.5 (0)

Table A.34: Average false discovery rate for 100 replications of AR2 CV 1SE with correlated predictors. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100	0.3 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.5 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.4 (0)	0.5 (0)
500	0.1 (0)	0.1 (0)	0.2 (0)	0.1 (0)	0.3 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0.1 (0)	0.2 (0)
1000	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100	0.3 (0)	0.4 (0)	0.5 (0)	0.6 (0)	0.6 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.5 (0)	0.6 (0)
500	0.2 (0)	0.2 (0)	0.3 (0)	0.3 (0)	0.4 (0)	0.1 (0)	0.2 (0)	0.2 (0)	0.3 (0)	0.4 (0)
1000	0.1 (0)	0.2 (0)	0.2 (0)	0.2 (0)	0.3 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0.2 (0)	0.3 (0)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100	0 (0)	0.3 (0)	0.5 (0)	0.6 (0)	0.7 (0)	0 (0)	0.3 (0)	0.3 (0)	0.4 (0)	0.6 (0)
500	0 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.5 (0)
1000	0 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.4 (0)	0 (0)	0.2 (0)	0.3 (0)	0.3 (0)	0.4 (0)

Table A.35: Average false discovery rate of 100 replications for α-modified CV 1SE with correlated predictors. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100	0.3 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.5 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.5 (0)
500	0.1 (0)	0.1 (0)	0.2 (0)	0.2 (0)	0.2 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0.2 (0)	0.2 (0)
1000	0 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.2 (0)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100	0.3 (0)	0.4 (0)	0.4 (0)	0.5 (0)	0.6 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.5 (0)	0.6 (0)
500	0.2 (0)	0.2 (0)	0.3 (0)	0.3 (0)	0.4 (0)	0.2 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.4 (0)
1000	0.1 (0)	0.2 (0)	0.2 (0)	0.2 (0)	0.3 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0.3 (0)	0.3 (0)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100	0 (0)	0.2 (0)	0.4 (0)	0.5 (0)	0.6 (0)	0 (0)	0.3 (0)	0.3 (0)	0.4 (0)	0.6 (0)
500	0 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.4 (0)	0 (0)	0.2 (0)	0.3 (0)	0.5 (0)	0.5 (0)
1000	0 (0)	0.2 (0)	0.3 (0)	0.3 (0)	0.4 (0)	0 (0)	0.2 (0)	0.3 (0)	0.4 (0)	0.5 (0)

Table A.36: Average false discovery rate for 100 replications of Relaxed Lasso with correlated predictors. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^{*}=5; SNR=1.25; \bar{\sigma}=4.71$					$p^{*}=5; SNR=5; \bar{\sigma}=2.35$				
100	0.5 (0)	0.5 (0)	0.6 (0)	0.6 (0)	0.7 (0)	0.3 (0)	0.4 (0)	0.5 (0)	0.5 (0)	0.6 (0)
500	0.2 (0)	0.3 (0)	0.3 (0)	0.3 (0)	0.3 (0)	0.2 (0)	0.1 (0)	0.2 (0)	0.2 (0)	0.2 (0)
1000	0.2 (0)	0.2 (0)	0.2 (0)	0.2 (0)	0.2 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.1 (0)
n	$p^{*}=15; SNR=1.25; \bar{\sigma}=8.3$					$p^{*}=15; SNR=5; \bar{\sigma}=4.15$				
100	0.4 (0)	0.6 (0)	0.6 (0)	0.8 (0)	0.8 (0)	0.4 (0)	0.5 (0)	0.6 (0)	0.7 (0)	0.8 (0)
500	0.3 (0)	0.3 (0)	0.4 (0)	0.4 (0)	0.5 (0)	0.2 (0)	0.2 (0)	0.2 (0)	0.2 (0)	0.3 (0)
1000	0.2 (0)	0.2 (0)	0.3 (0)	0.3 (0)	0.3 (0)	0.1 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0.2 (0)
n	$p^{*}=50; SNR=1.25; \bar{\sigma}=15.07$					$p^{*}=50; SNR=5; \bar{\sigma}=7.53$				
100	0 (0)	0.3 (0)	0.5 (0)	0.7 (0)	0.7 (0)	0 (0)	0.3 (0)	0.5 (0)	0.6 (0)	0.8 (0)
500	0 (0)	0.3 (0)	0.5 (0)	0.6 (0)	0.6 (0)	0 (0)	0.2 (0)	0.3 (0)	0.5 (0)	0.5 (0)
1000	0 (0)	0.3 (0)	0.4 (0)	0.4 (0)	0.5 (0)	0 (0)	0.2 (0)	0.2 (0)	0.2 (0)	0.3 (0)

Table A.37: Average number of false negatives from 100 replications of CV 1SE with correlated predictors. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100	1.8 (0.1)	2 (0.1)	2.4 (0.1)	2.6 (0.2)	2.9 (0.1)	0.4 (0.1)	0.4 (0.1)	0.5 (0.1)	0.8 (0.1)	0.9 (0.1)
500	0.3 (0.1)	0.4 (0.1)	0.5 (0.1)	0.6 (0.1)	0.6 (0.1)	0.1 (0)	0 (0)	0.1 (0)	0.1 (0)	0.1 (0)
1000	0.2 (0)	0.2 (0)	0.3 (0.1)	0.3 (0.1)	0.3 (0.1)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100	8.4 (0.4)	9.9 (0.4)	12.2 (0.3)	12.5 (0.3)	13.2 (0.2)	3.4 (0.4)	4.2 (0.3)	5.6 (0.4)	7.8 (0.4)	8.5 (0.4)
500	3.2 (0.4)	3 (0.3)	3.7 (0.4)	4 (0.4)	4.8 (0.4)	0.6 (0.1)	0.7 (0.1)	1 (0.2)	0.9 (0.2)	1 (0.2)
1000	1.2 (0.2)	1.6 (0.2)	1.9 (0.3)	2.4 (0.3)	2.5 (0.3)	0.1 (0)	0.2 (0.1)	0.3 (0.1)	0.4 (0.1)	0.4 (0.1)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100	39.3 (0.8)	43.8 (0.6)	46.1 (0.5)	47.6 (0.2)	48.2 (0.3)	20.8 (1.4)	30.9 (1.2)	38.6 (0.8)	42.9 (0.6)	45.4 (0.4)
500	13.5 (1.3)	18.7 (1.4)	22.5 (1.4)	28.4 (1.3)	31.6 (1.2)	3.1 (0.6)	5.2 (0.9)	7.6 (1)	9.1 (1.1)	10.4 (1.2)
1000	8.4 (1.1)	11.1 (1.3)	13.4 (1.3)	15.8 (1.3)	18.9 (1.4)	1.4 (0.4)	1.8 (0.4)	2.2 (0.5)	3 (0.5)	3.9 (0.7)

Table A.38: Average number of false negatives for 100 replications of AR2 CV 1SE with correlated predictors. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^{*}=5; SNR=1.25; \bar{\sigma}=4.71$					$p^{*}=5; SNR=5; \bar{\sigma}=2.35$				
100		2.4	2.6	2.9	3	3.3	0.8	0.8	1.1	1.4	1.4
		(0.2)	(0.2)	(0.2)	(0.2)	(0.2)	(0.1)	(0.1)	(0.1)	(0.1)	(0.2)
500		0.7	0.7	0.9	1.1	1	0.2	0.2	0.2	0.2	0.3
		(0.1)	(0.1)	(0.1)	(0.1)	(0.1)	(0.1)	(0)	(0)	(0)	(0.1)
1000		0.4	0.3	0.6	0.6	0.6	0.1	0 (0)	0.1	0.1	0.1
		(0.1)	(0.1)	(0.1)	(0.1)	(0.1)	(0)		(0)	(0)	(0)
n		$p^{*}=15; SNR=1.25; \bar{\sigma}=8.3$					$p^{*}=15; SNR=5; \bar{\sigma}=4.15$				
100		9.3	10.7	12.4	13.1	13.5	4.4	5.4	6.8	9.5	9.9
		(0.4)	(0.4)	(0.3)	(0.2)	(0.2)	(0.5)	(0.4)	(0.5)	(0.4)	(0.4)
500		4.2	3.9	4.8	4.8	5.8	1.2	0.9	1.5	1.2	1.3
		(0.4)	(0.4)	(0.5)	(0.5)	(0.5)	(0.3)	(0.2)	(0.3)	(0.2)	(0.2)
1000		1.7	2.2	2.5	3.2	3.4	0.2	0.3	0.5	0.7	0.6
		(0.3)	(0.3)	(0.4)	(0.4)	(0.4)	(0.1)	(0.1)	(0.2)	(0.2)	(0.2)
n		$p^{*}=50; SNR=1.25; \bar{\sigma}=15.07$					$p^{*}=50; SNR=5; \bar{\sigma}=7.53$				
100		40.4	43.7	45.7	47.8	48.5	23.4	32.2	40.1	45.3	47.1
		(1.1)	(0.8)	(0.6)	(0.3)	(0.3)	(1.7)	(1.3)	(1)	(0.6)	(0.4)
500		14.4	20.2	24.5	29.8	33.1	4.2	6.3	9	10.4	12.2
		(1.5)	(1.6)	(1.6)	(1.4)	(1.3)	(0.8)	(1)	(1.2)	(1.3)	(1.5)
1000		9.5	12.3	15.3	17.5	20.5	1.7	2.5	2.8	3.6	4.7
		(1.2)	(1.4)	(1.5)	(1.5)	(1.6)	(0.5)	(0.6)	(0.6)	(0.7)	(0.8)

Table A.39: Average number of false negatives of 100 replications for α-modified CV 1SE with correlated predictors. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100		2.3 (0.2)	2.6 (0.2)	3 (0.1)	3.1 (0.2)	3.3 (0.1)	0.7 (0.1)	0.7 (0.1)	0.8 (0.1)	1.1 (0.1)	1.3 (0.1)
500		0.6 (0.1)	0.7 (0.1)	1 (0.1)	1.1 (0.1)	1.1 (0.1)	0.1 (0)	0.1 (0)	0.1 (0)	0.2 (0.1)	0.3 (0.1)
1000		0.4 (0.1)	0.3 (0.1)	0.5 (0.1)	0.6 (0.1)	0.6 (0.1)	0 (0)	0 (0)	0.1 (0)	0 (0)	0.1 (0)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100		10 (0.4)	11.3 (0.4)	13.3 (0.2)	13.1 (0.3)	13.7 (0.2)	4.1 (0.4)	4.8 (0.4)	6.4 (0.5)	8.9 (0.5)	9.2 (0.4)
500		4 (0.4)	4 (0.4)	4.8 (0.5)	4.9 (0.5)	6 (0.5)	0.8 (0.2)	0.8 (0.2)	1.4 (0.3)	1.1 (0.2)	1.3 (0.2)
1000		1.4 (0.2)	2.1 (0.3)	2.6 (0.4)	3.3 (0.4)	3.3 (0.4)	0.1 (0.1)	0.2 (0.1)	0.4 (0.1)	0.5 (0.1)	0.6 (0.2)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100		42.9 (1)	46.5 (0.5)	47.3 (0.5)	48.1 (0.4)	48.6 (0.3)	22.1 (1.6)	32.8 (1.3)	40.2 (0.9)	43.9 (0.7)	46.3 (0.5)
500		14.7 (1.5)	21.3 (1.6)	26.1 (1.6)	33.2 (1.4)	36.9 (1.3)	3.3 (0.6)	5.7 (0.9)	8.2 (1.1)	9.9 (1.3)	11.9 (1.4)
1000		9.3 (1.3)	12.6 (1.5)	15.5 (1.5)	18.4 (1.5)	22.1 (1.6)	1.4 (0.4)	2 (0.4)	2.4 (0.5)	3.2 (0.6)	4.3 (0.7)

Table A.40: Average number of false negatives for 100 replications of Relaxed Lasso with correlated predictors. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100	1.5 (0.1)	1.7 (0.1)	2.2 (0.1)	2.3 (0.2)	2.4 (0.2)	0.4 (0.1)	0.4 (0.1)	0.5 (0.1)	0.8 (0.1)	0.7 (0.1)
500	0.1 (0)	0.2 (0.1)	0.3 (0.1)	0.4 (0.1)	0.6 (0.1)	0 (0)	0 (0)	0 (0)	0 (0)	0.1 (0)
1000	0 (0)	0.1 (0)	0.1 (0)	0.2 (0)	0.2 (0.1)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
n	$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100	6.5 (0.4)	8.3 (0.4)	9.7 (0.4)	10.7 (0.4)	11.2 (0.3)	2.7 (0.3)	3.4 (0.3)	4.9 (0.4)	6.4 (0.4)	7 (0.4)
500	2.2 (0.3)	2.4 (0.3)	3.3 (0.3)	3.7 (0.3)	4.3 (0.3)	0.5 (0.1)	0.6 (0.1)	0.8 (0.1)	0.9 (0.2)	1.2 (0.2)
1000	0.9 (0.1)	0.9 (0.2)	1.5 (0.2)	1.9 (0.2)	2.4 (0.2)	0 (0)	0.1 (0)	0.2 (0)	0.3 (0.1)	0.4 (0.1)
n	$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100	33.6 (1.2)	38.6 (0.9)	40.9 (0.9)	43.4 (0.7)	45.5 (0.5)	14.8 (1.4)	24 (1.2)	28.7 (1.4)	35.2 (1.1)	39.8 (0.8)
500	5.7 (0.9)	12.9 (1.2)	17.9 (1.3)	23.6 (1.3)	27 (1.3)	0.9 (0.3)	4 (0.7)	6.4 (0.7)	8.3 (1)	10.1 (1)
1000	2.1 (0.5)	6.5 (0.8)	9.4 (0.9)	13.1 (0.9)	16.1 (1.2)	0.3 (0.2)	1.4 (0.2)	2.7 (0.4)	4 (0.4)	4.9 (0.5)

Table A.41: Average prediction bias from 100 replications of CV 1SE with correlated predictors. Standard errors given in parentheses.

p	50	100	200	400	800	50	100	200	400	800
n	$p^{*} = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^{*} = 5; SNR = 5; \bar{\sigma} = 2.35$				
100	22.8	24.2	27.7	29.9	29.2	12	12.8	15.6	16.6	15.9
	(0.9)	(0.9)	(1)	(1.4)	(1.1)	(0.5)	(0.6)	(0.7)	(0.8)	(0.7)
500	29.2	30.9	35.2	35	38.7	15.1	15.8	17.8	17.4	19.8
	(1.1)	(1.4)	(1.5)	(1.6)	(1.8)	(0.7)	(0.7)	(0.8)	(0.8)	(1)
1000	33.7	33.2	37.5	39.7	38.8	17.6	16.6	19.1	19.2	19.1
	(1.5)	(1.4)	(1.7)	(2)	(1.8)	(0.8)	(0.7)	(0.9)	(0.9)	(1)
n	$p^{*} = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^{*} = 15; SNR = 5; \bar{\sigma} = 4.15$				
100	47.5	50	54.8	56.1	56.7	28	29.6	33.7	38.2	37.3
	(2)	(1.6)	(1.7)	(1.6)	(1.5)	(1.4)	(0.9)	(1.3)	(1.2)	(1.1)
500	67.6	67.7	73	72.9	79.6	35.3	35.7	39.3	38.5	42.7
	(3)	(3)	(3.1)	(3)	(2.8)	(1.7)	(1.7)	(2)	(1.8)	(1.9)
1000	65.8	71.8	78.6	88.7	90	34.1	37	39.6	45.3	46.8
	(3.2)	(3.4)	(3.6)	(4.6)	(3.3)	(1.8)	(1.7)	(1.9)	(2.3)	(2)
n	$p^{*} = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^{*} = 50; SNR = 5; \bar{\sigma} = 7.53$				
100	96.1	105.4	102.2	105.6	106.1	58.3	70.8	75.6	84.2	85.1
	(2.9)	(3.1)	(2)	(2.3)	(2.5)	(2.3)	(2.3)	(1.8)	(2.1)	(1.7)
500	128.9	146.7	156.8	172.6	180	69.8	80.2	90.9	99	103.2
	(5.8)	(5.5)	(5.1)	(5.2)	(4.7)	(3.5)	(4)	(3.9)	(4.4)	(4.1)
1000	141.9	164.4	178.9	187.1	207.8	73.1	86.1	94.6	99.3	112
	(6.8)	(7.3)	(7.4)	(6.4)	(7)	(3.9)	(4)	(4.2)	(4.3)	(4.7)

Table A.42: Average prediction bias for 100 replications of AR2 CV 1SE with correlated predictors. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100		21.9	24	26.2	27.9	26.9	12.9	13.9	15.4	16.8	16.7
		(0.8)	(1)	(1.1)	(1.3)	(1.1)	(0.6)	(0.7)	(0.7)	(0.8)	(0.8)
500		28.4	29.7	34.7	32.3	36.1	16.3	16.6	18	17	19.2
		(1.3)	(1.4)	(1.7)	(1.5)	(1.8)	(0.8)	(0.9)	(0.8)	(0.8)	(0.9)
1000		32.6	33.1	35.1	37.7	35.8	19	17.9	19.4	19.5	19.4
		(1.5)	(1.5)	(1.6)	(2.1)	(1.7)	(0.9)	(0.8)	(1)	(1)	(0.9)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100		47.9	48.8	53.5	52.7	52.3	29.8	31.7	35.4	38.9	38.4
		(2)	(1.5)	(1.6)	(1.4)	(1.1)	(1.6)	(1.2)	(1.4)	(1.3)	(1.1)
500		66.1	67.3	71.1	71.7	77.8	37.3	37.1	39.7	38.4	42.4
		(2.7)	(3.1)	(3)	(3.1)	(3.3)	(2)	(1.9)	(2.2)	(2)	(2.1)
1000		66.7	69.5	75.7	85.3	88.5	35.2	38.2	41.9	45.8	46.2
		(3.3)	(3.4)	(3.4)	(4.4)	(3.5)	(1.8)	(1.9)	(2.1)	(2.6)	(2.3)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100		95.2	101	95.6	100.9	103	61.2	71.5	75	85.5	85.9
		(2.7)	(3)	(1.7)	(2.1)	(2.6)	(2.5)	(2.6)	(1.9)	(1.9)	(1.9)
500		128.9	148.6	157.7	172.8	178.2	73.1	82.9	94.1	101.8	105.2
		(5.4)	(5.5)	(5)	(5.2)	(4.3)	(3.7)	(4.1)	(4.2)	(4.8)	(4.5)
1000		142.9	164.4	181	185.8	204.6	76.8	89.5	97	99.9	112.2
		(6.5)	(7)	(7.4)	(6.6)	(6.7)	(4)	(4.6)	(4.5)	(4.7)	(5.2)

Table A.43: Average number of prediction bias of 100 replications for α-modified CV 1SE with correlated predictors. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100		21.6	23.7	26.9	28.1	28.4	11.8	12.9	14.8	15.7	15.6
		(0.8)	(1)	(1.1)	(1.3)	(1.2)	(0.5)	(0.7)	(0.7)	(0.8)	(0.7)
500		28	29.5	34.2	32.4	37	14.9	15.2	16.6	16	18
		(1.2)	(1.4)	(1.6)	(1.5)	(2.1)	(0.7)	(0.8)	(0.8)	(0.8)	(0.9)
1000		32.1	31.7	34.8	37.4	35.8	17.1	16.2	17.9	17.9	18.3
		(1.4)	(1.4)	(1.6)	(2.1)	(1.7)	(0.8)	(0.7)	(1)	(0.9)	(0.9)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100		48.3	50.6	54.7	54.2	54.7	28.4	30.2	34.1	37.6	36.7
		(1.9)	(1.6)	(1.5)	(1.5)	(1.3)	(1.4)	(1.1)	(1.4)	(1.2)	(1)
500		65.2	67.1	71.1	72.1	79.2	34.8	34.9	38	36.8	41.2
		(2.6)	(3)	(3)	(3.2)	(3.2)	(1.8)	(1.8)	(2.1)	(1.9)	(2.1)
1000		65	69.2	75.8	85.7	88.7	33	36	39.1	43.2	44.3
		(3.2)	(3.2)	(3.3)	(4.5)	(3.5)	(1.7)	(1.8)	(2)	(2.4)	(2.2)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100		97	103.6	98	104.4	105.4	59.5	71.1	75.1	82.5	84.5
		(2.4)	(2.8)	(1.6)	(2.1)	(2.4)	(2.3)	(2.2)	(1.7)	(1.9)	(2.2)
500		128.7	150.9	161.4	179	185.8	69.5	80.5	91.8	99.4	103.6
		(5.3)	(5.4)	(4.8)	(5)	(4.1)	(3.5)	(4)	(4.1)	(4.7)	(4.4)
1000		141.3	166	182.3	190.1	210.6	73.1	85.7	94.3	97.3	109.3
		(6.6)	(7.2)	(7.4)	(6.7)	(6.7)	(3.8)	(4.2)	(4.3)	(4.6)	(5)

Table A.44: Average prediction bias for 100 replications of Relaxed Lasso with correlated predictors. Standard errors given in parentheses.

	p	50	100	200	400	800	50	100	200	400	800
n		$p^* = 5; SNR = 1.25; \bar{\sigma} = 4.71$					$p^* = 5; SNR = 5; \bar{\sigma} = 2.35$				
100		18.3	19.7	25.2	25.3	29.8	9.1	10.2	13.1	14.7	15.5
		(0.7)	(0.9)	(1.2)	(1.3)	(1.8)	(0.5)	(0.6)	(0.8)	(0.9)	(1)
500		15.3	19.3	21.3	20.9	24.9	8	8.4	9.8	9.3	10.6
		(1)	(1.4)	(1.3)	(1.3)	(1.7)	(0.4)	(0.6)	(0.7)	(0.7)	(0.8)
1000		14.6	16.1	18	20	19.5	7.7	7.6	8.4	8	7.8
		(1)	(1.2)	(1.5)	(1.6)	(1.4)	(0.5)	(0.5)	(0.6)	(0.6)	(0.5)
n		$p^* = 15; SNR = 1.25; \bar{\sigma} = 8.3$					$p^* = 15; SNR = 5; \bar{\sigma} = 4.15$				
100		40.4	44.3	47.9	50.6	53.4	24.8	27.3	30.6	33.9	34.7
		(1.5)	(1.6)	(1.6)	(1.7)	(1.9)	(1.1)	(1.1)	(1.2)	(1.3)	(1.1)
500		49.2	52.8	56.5	57.3	63.5	24.2	24.9	27	27.4	31.1
		(2.3)	(2.4)	(2.5)	(2.7)	(2.6)	(1.5)	(1.6)	(1.8)	(1.8)	(2)
1000		44	47.4	53.8	61.4	64.9	19.3	21	23.1	27.1	26.3
		(2.7)	(2.6)	(3)	(3.7)	(3.2)	(1.2)	(1.3)	(1.6)	(2.1)	(1.9)
n		$p^* = 50; SNR = 1.25; \bar{\sigma} = 15.07$					$p^* = 50; SNR = 5; \bar{\sigma} = 7.53$				
100		81.6	89.2	90.1	98	104.4	52.4	60.2	61.7	70.1	72.2
		(2)	(2.6)	(2)	(3.2)	(4.1)	(1.9)	(2)	(1.6)	(2.2)	(2.3)
500		102.7	125	136.1	148.7	154.6	57.2	69.7	79.4	88.5	93.2
		(4)	(4.4)	(4)	(4)	(3.7)	(2.8)	(3.4)	(3.4)	(3.8)	(3.5)
1000		105.5	134.2	151.1	160.2	178	55.2	69.5	76.4	81.1	90.5
		(4.6)	(5.6)	(5.5)	(5.4)	(5.6)	(2.8)	(3.4)	(4.1)	(4.1)	(4.8)

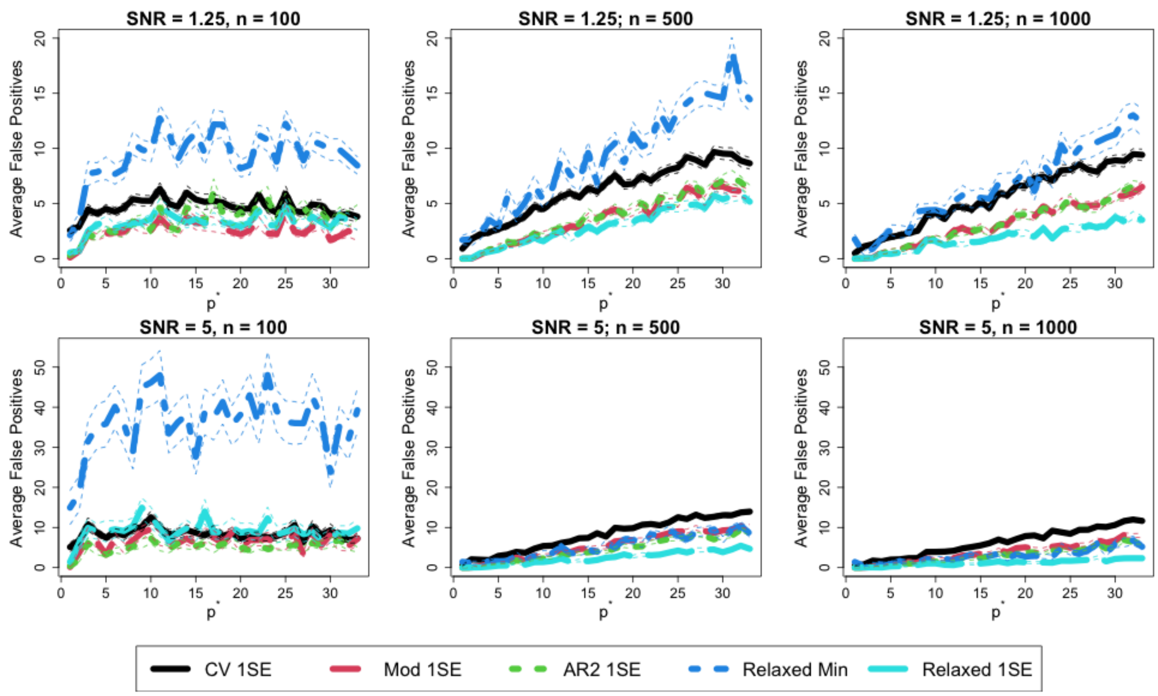

Figure A.8: Average false positives from 100 replications of simulated data with $n = \{100, 500, 1000\}$ and $p = 100$ for correlated predictors. Thin dotted lines represent the mean $\pm$ one standard error.

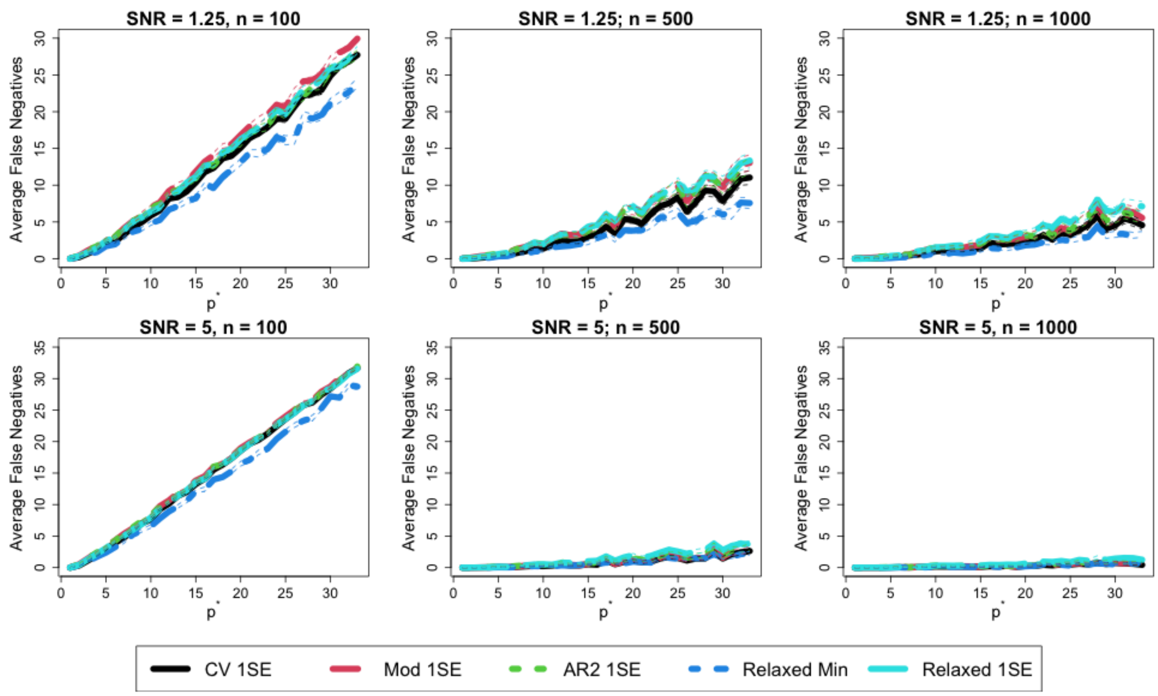

Figure A.9: Average false negatives from 100 replications of simulated data with $n = 100, n = 500, n = 1000$ and $p = 100$ with independent predictors for both SNR $= 1.25$ and SNR $= 5$. Thin dotted lines represent the mean $\pm$ one standard error.

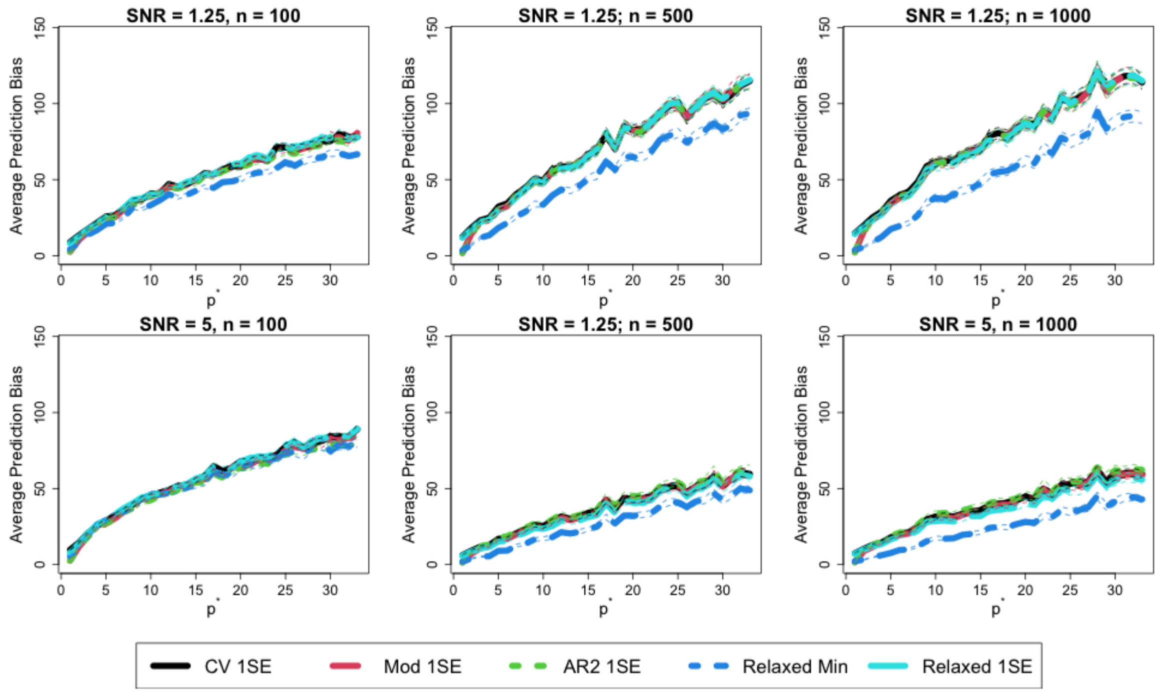

Figure A.10: Average prediction bias from 100 replications of simulated data with $n = \{100, 500, 1000\}$ and $p = 100$ for correlated predictors. Thin dotted lines represent the mean $\pm$ one standard error.

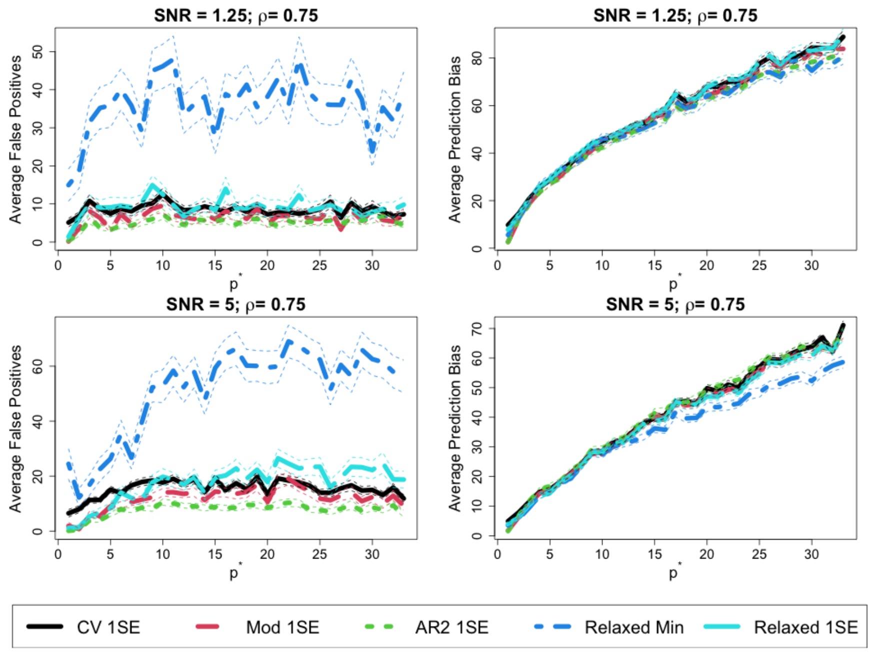

Figure A.11: Average false positives and prediction error plots from 100 replications of simulated data with $n = 100$ and $p = 800$ for both SNR $= 0.5$ and SNR $= 2$ with correlated predictors. Thin dotted lines represent the mean $\pm$ one standard error.

Table A.45: Average false discovery rate for 100 replications of simulated data with correlated predictors and $n = 100$. Standard errors given in parentheses.

	$p = 100$				$p = 800$			
	SNR = 1.25		SNR = 5		SNR = 1.25		SNR = 5	
	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$
CV Min	0.75 (0.01)	0.48 (0.01)	0.74 (0.01)	0.5 (0.01)	0.86 (0.01)	0.84 (0.01)	0.87 (0.01)	0.76 (0.02)
CV 1SE	0.53 (0.03)	0.34 (0.02)	0.52 (0.02)	0.4 (0.01)	0.65 (0.03)	0.64 (0.04)	0.7 (0.02)	0.61 (0.03)
Mod Min	0.63 (0.03)	0.4 (0.02)	0.67 (0.02)	0.48 (0.01)	0.7 (0.03)	0.77 (0.03)	0.78 (0.02)	0.7 (0.03)
Mod 1SE	0.35 (0.03)	0.28 (0.03)	0.34 (0.02)	0.35 (0.02)	0.45 (0.04)	0.64 (0.04)	0.53 (0.03)	0.49 (0.04)
AR2 Min	0.67 (0.02)	0.5 (0.02)	0.64 (0.02)	0.49 (0.01)	0.71 (0.03)	0.81 (0.02)	0.73 (0.02)	0.7 (0.02)
AR2 1SE	0.38 (0.03)	0.36 (0.03)	0.28 (0.02)	0.34 (0.02)	0.45 (0.04)	0.76 (0.03)	0.44 (0.03)	0.55 (0.03)
R2 Cut	0.94 (0)	0.66 (0)	0.91 (0)	0.57 (0.01)	0.95 (0)	0.9 (0)	0.91 (0)	0.84 (0)
AIC	0.68 (0.02)	0.43 (0.02)	0.7 (0.01)	0.48 (0.01)	0.79 (0.02)	0.79 (0.02)	0.81 (0.01)	0.7 (0.02)
BIC	0.5 (0.03)	0.33 (0.02)	0.5 (0.02)	0.39 (0.01)	0.66 (0.03)	0.67 (0.04)	0.67 (0.02)	0.59 (0.03)
ERIC	0.37 (0.03)	0.18 (0.02)	0.4 (0.02)	0.21 (0.02)	0.98 (0)	0.94 (0)	0.98 (0)	0.92 (0)
GCV	0.77 (0.01)	0.5 (0.01)	0.74 (0.01)	0.52 (0.01)	0.98 (0)	0.94 (0)	0.98 (0)	0.92 (0)
Relaxed Min	0.55 (0.03)	0.44 (0.02)	0.39 (0.03)	0.45 (0.01)	0.7 (0.03)	0.82 (0.02)	0.55 (0.03)	0.75 (0.02)
Relaxed 1SE	0.4 (0.03)	0.32 (0.03)	0.25 (0.03)	0.36 (0.02)	0.5 (0.04)	0.69 (0.04)	0.44 (0.03)	0.61 (0.03)

Table A.46: Average prediction bias for 100 replications of simulated data with correlated predictors and $n = 100$. Standard errors given in parentheses.

	$p = 100$				$p = 800$			
	SNR = 1.25		SNR = 5		SNR = 1.25		SNR = 5	
	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$
CV Min	20.35	63.73	11.82	43.31	25.17	73.02	15.93	56.79
	(0.79)	(1.34)	(0.45)	(1.38)	(1.06)	(2.39)	(0.79)	(1.43)
CV 1SE	25.74	78.01	14.04	50.31	28.69	88.89	15.65	71.1
	(1.1)	(1.83)	(0.63)	(1.69)	(0.95)	(2.69)	(0.69)	(1.47)
Mod Min	21.14	73.03	11.81	45.3	27.07	80.28	15.1	60.23
	(0.82)	(1.59)	(0.5)	(1.42)	(1.19)	(2.8)	(0.85)	(1.62)
Mod 1SE	23.74	80.66	13.73	51.53	26.92	83.82	15.29	68.23
	(0.89)	(1.81)	(0.64)	(1.74)	(0.86)	(2.13)	(0.74)	(1.4)
AR2 Min	21.65	73.11	11.92	45.59	26.4	78.8	14.32	57.97
	(0.89)	(1.75)	(0.52)	(1.43)	(1.06)	(2.22)	(0.76)	(1.27)
AR2 1SE	23.96	78	14.99	52.87	27.22	81.7	16.78	68.97
	(0.91)	(2.07)	(0.73)	(1.83)	(0.87)	(2)	(0.86)	(1.36)
R2 Cut	41.31	105.19	17.04	45.12	34.2	91.36	15.15	49.18
	(1.83)	(4.59)	(0.74)	(1.77)	(1.67)	(4.27)	(0.76)	(1.82)
AIC	20.17	65.73	11.87	44.71	23.98	73.47	14.07	59.81
	(0.71)	(1.22)	(0.47)	(1.42)	(0.78)	(1.98)	(0.61)	(1.15)
BIC	22.83	72.56	12.79	49.32	26.43	81.36	15.19	66.64
	(0.84)	(1.07)	(0.51)	(1.49)	(0.75)	(2.02)	(0.65)	(1.08)
ERIC	29.44	98.14	15.06	76.35	47.01	122.18	23.78	62.67
	(1.56)	(3.53)	(0.7)	(1.76)	(2.32)	(5.97)	(1.18)	(3.08)
GCV	22.53	67.48	12.43	44.43	47.01	122.18	23.78	62.67
	(0.99)	(1.81)	(0.49)	(1.61)	(2.32)	(5.97)	(1.18)	(3.08)
Relaxed Min	20.9	66.67	11.37	44.59	29.14	80.59	14.85	58.67
	(0.88)	(1.62)	(0.57)	(1.49)	(1.76)	(3.22)	(1.04)	(2.05)
Relaxed 1SE	25.38	77.77	13.27	50.73	29.07	89.21	14.46	67.88
	(1.14)	(1.83)	(0.69)	(1.76)	(1.21)	(3.14)	(0.84)	(1.8)

Table A.47: Average false negatives for 100 replications of simulated data with correlated predictors and $n = 100$. Standard errors given in parentheses.

	$p = 100$				$p = 800$			
	SNR = 1.25		SNR = 5		SNR = 1.25		SNR = 5	
	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$
CV Min	1.33 (0.14)	21.88 (0.65)	0.25 (0.05)	10.86 (0.77)	2 (0.14)	28.78 (0.44)	0.44 (0.08)	25.19 (0.51)
CV 1SE	2.1 (0.14)	27.72 (0.45)	0.55 (0.07)	16.5 (0.8)	2.85 (0.14)	31.65 (0.18)	0.69 (0.1)	29.09 (0.31)
Mod Min	1.57 (0.15)	25.05 (0.63)	0.38 (0.07)	11.82 (0.83)	2.39 (0.15)	29.13 (0.52)	0.54 (0.09)	25.76 (0.62)
Mod 1SE	2.68 (0.15)	29.94 (0.44)	0.87 (0.12)	18.17 (0.94)	3.27 (0.15)	31.69 (0.28)	0.95 (0.13)	30.01 (0.34)
AR2 Min	1.5 (0.16)	20.75 (0.89)	0.43 (0.08)	11.31 (0.81)	2.27 (0.15)	29.26 (0.4)	0.71 (0.11)	26.9 (0.41)
AR2 1SE	2.64 (0.15)	27.74 (0.6)	1.04 (0.13)	18.9 (0.98)	3.3 (0.15)	31.92 (0.17)	1.17 (0.14)	30.49 (0.26)
R2 Cut	0.35 (0.06)	5.21 (0.27)	0.07 (0.03)	6.2 (0.35)	1.56 (0.13)	25.88 (0.27)	0.39 (0.07)	22.5 (0.34)
AIC	1.49 (0.14)	24.55 (0.56)	0.29 (0.06)	12.39 (0.79)	2.27 (0.14)	30.6 (0.21)	0.59 (0.09)	27.3 (0.36)
BIC	2.17 (0.13)	28.63 (0.33)	0.55 (0.07)	17.23 (0.86)	3.02 (0.13)	31.83 (0.15)	0.72 (0.1)	29.23 (0.3)
ERIC	2.95 (0.12)	31.54 (0.17)	0.78 (0.09)	29.26 (0.3)	1.19 (0.11)	19.79 (0.29)	0.26 (0.06)	16.1 (0.33)
GCV	1.19 (0.13)	20.42 (0.72)	0.24 (0.05)	10.45 (0.7)	1.19 (0.11)	19.79 (0.29)	0.26 (0.06)	16.1 (0.33)
Relaxed Min	1.7 (0.14)	24.03 (0.62)	0.44 (0.08)	13.4 (0.9)	2.38 (0.14)	28.73 (0.48)	0.63 (0.09)	24.62 (0.61)
Relaxed 1SE	2.42 (0.14)	28.39 (0.48)	0.85 (0.1)	18.19 (0.88)	3.09 (0.14)	31.57 (0.25)	0.99 (0.13)	28.83 (0.42)

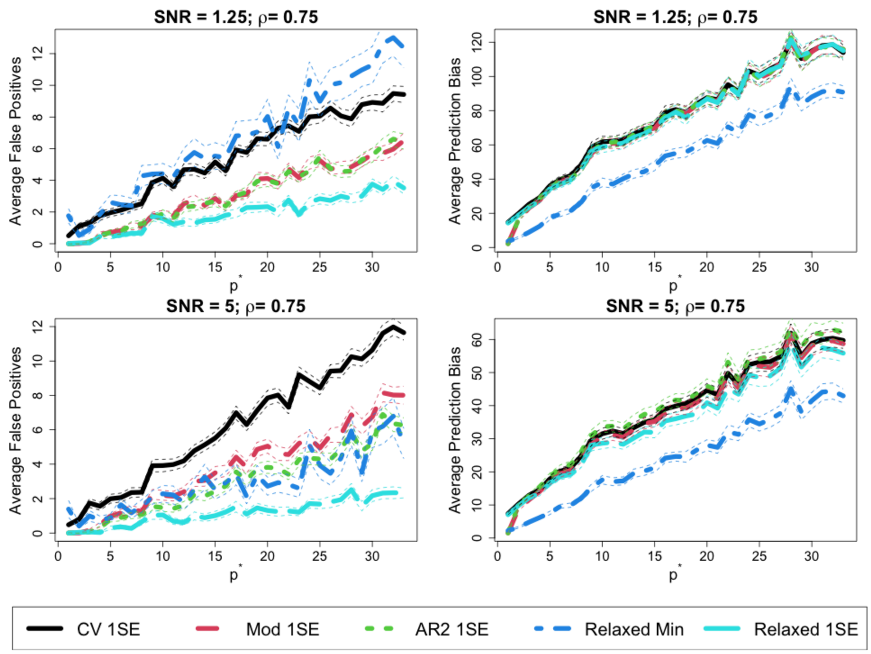

Figure A.12: Average false positives and prediction error plots from 100 replications of simulated data with $n = 1000$ and $p = 100$ for both SNR $= 0.5$ and SNR $= 2$ with correlated predictors. Thin dotted lines represent the mean $\pm$ one standard error.

Table A.48: Average false discovery rate for 100 replications of simulated data with correlated predictors and $n = 1000$. Standard errors given in parentheses.

	SNR $= 1.25$		SNR $= 5$	
	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$
CV Min	0.7 (0.01)	0.48 (0.01)	0.7 (0.01)	0.5 (0.01)
CV 1SE	0.23 (0.02)	0.24 (0.01)	0.21 (0.02)	0.26 (0.01)
Mod Min	0.58 (0.02)	0.45 (0.01)	0.56 (0.02)	0.45 (0.01)
Mod 1SE	0.08 (0.02)	0.18 (0.01)	0.12 (0.02)	0.19 (0.01)
AR2 Min	0.59 (0.02)	0.47 (0.01)	0.56 (0.02)	0.45 (0.01)
AR2 1SE	0.08 (0.02)	0.17 (0.01)	0.09 (0.02)	0.15 (0.01)
R2 Cut	0.1 (0.02)	0.19 (0.01)	0.14 (0.02)	0.2 (0.01)
AIC	0.7 (0.01)	0.47 (0.01)	0.69 (0.01)	0.49 (0.01)
BIC	0.35 (0.02)	0.18 (0.01)	0.35 (0.02)	0.23 (0.01)
ERIC	0.26 (0.02)	0.08 (0.01)	0.25 (0.02)	0.13 (0.01)
GCV	0.71 (0.01)	0.48 (0.01)	0.71 (0.01)	0.5 (0.01)
Relaxed Min	0.2 (0.03)	0.24 (0.02)	0.09 (0.02)	0.11 (0.01)
Relaxed 1SE	0.05 (0.01)	0.11 (0.01)	0.04 (0.01)	0.06 (0.01)

Table A.49: Average prediction bias for 100 replications of simulated data with correlated predictors and $n = 1000$. Standard errors given in parentheses.

	SNR = 1.25		SNR = 5	
	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$
CV Min	22.47 (1.11)	91.73 (3.45)	11.27 (0.52)	48.34 (1.83)
CV 1SE	36.11 (1.77)	113.94 (4.37)	17.87 (0.87)	59.71 (2.39)
Mod Min	21.09 (1.17)	93.33 (3.54)	10.24 (0.55)	47.5 (1.92)
Mod 1SE	34.13 (1.7)	115.84 (4.52)	16.99 (0.91)	58.69 (2.43)
AR2 Min	21.11 (1.17)	93.62 (3.62)	10.27 (0.56)	47.53 (1.92)
AR2 1SE	34.79 (1.74)	115.86 (4.65)	18.7 (0.96)	62.24 (2.68)
R2 Cut	67.19 (6.77)	143.47 (11.71)	29.92 (4.11)	64.7 (2.06)
AIC	22.7 (1.11)	92.76 (3.49)	11.28 (0.51)	48.61 (1.84)
BIC	26.92 (1.25)	120.22 (4.63)	13.21 (0.58)	59.85 (2.4)
ERIC	30.27 (1.51)	155.85 (5.53)	14.84 (0.73)	74.46 (3.78)
GCV	23.18 (1.26)	92.49 (3.44)	11.37 (0.51)	48.7 (1.85)
Relaxed Min	17.52 (1.34)	90.79 (3.71)	7.61 (0.49)	42.96 (2.1)
Relaxed 1SE	34.35 (1.97)	115.03 (4.66)	16.61 (0.84)	55.85 (2.53)

Table A.50: Average false negatives for 100 replications of simulated data with correlated predictors and $n = 1000$. Standard errors given in parentheses.

	SNR = 1.25		SNR = 5	
	$p^* = 5$	$p^* = 33$	$p^* = 5$	$p^* = 33$
CV Min	0.04 (0.02)	1.38 (0.27)	0 (0)	0.12 (0.05)
CV 1SE	0.24 (0.05)	4.56 (0.6)	0.02 (0.01)	0.43 (0.14)
Mod Min	0.06 (0.03)	1.54 (0.29)	0 (0)	0.12 (0.05)
Mod 1SE	0.39 (0.07)	5.56 (0.71)	0.04 (0.02)	0.48 (0.14)
AR2 Min	0.06 (0.03)	1.4 (0.27)	0 (0)	0.12 (0.05)
AR2 1SE	0.4 (0.07)	5.58 (0.73)	0.06 (0.03)	0.72 (0.2)
R2 Cut	0.38 (0.07)	5.27 (0.71)	0.04 (0.02)	0.46 (0.14)
AIC	0.03 (0.02)	1.68 (0.3)	0 (0)	0.11 (0.04)
BIC	0.16 (0.04)	6.52 (0.72)	0 (0)	0.52 (0.13)
ERIC	0.22 (0.05)	12.45 (0.95)	0 (0)	1.83 (0.44)
GCV	0.03 (0.02)	1.58 (0.29)	0 (0)	0.11 (0.04)
Relaxed Min	0.09 (0.03)	3.31 (0.39)	0 (0)	0.66 (0.11)
Relaxed 1SE	0.37 (0.06)	7.12 (0.66)	0.03 (0.02)	1.25 (0.2)

A.2 EMG Application Approximations and Results

The model given in the main document does not involve raw data, so some manipulations are required in order to perform estimation. The response, velocity (y_i), must be estimated from hand position, z_i and covariate functions in the model must be created from discrete observations. These tasks are performed exactly as described in Stallrich et al. (2020).

Furthermore, the model also requires approximations for γ_j. We represent all γ_j using: $\gamma_j(t,z) \approx \sum_{l=1}^{L}\sum_{m=1}^{M}\omega_l(t)\tau_m(z)\beta_{jlm}$ where $\omega_l(t)$ and $\tau_m(z)$ are the same univariate basis functions for all γ_j, and the β_{jlm}'s are basis coefficients that must be estimated. In matrix form, we have $\gamma_j(t,z) \approx \boldsymbol{\omega}^T(t)\mathbf{B}_j\boldsymbol{\tau}(z)$ where $\boldsymbol{\omega}^T(t) = \big(\omega_1(t), ..., \omega_L(t)\big)$, $\boldsymbol{\tau}^T(z) = \big(\tau_1(z), ..., \tau_M(z)\big)$, and $\mathbf{B}_j = \big(\beta_{jlm}\big)$ is the $L \times M$ coefficient matrix.

The integral in the model is approximated:

$$\int_{-\delta}^{0} X_{ij}(t)\gamma_j(t,z_i)dt \approx \left(\sum_{k=-\delta}^{0} x_{(i+k)j}\boldsymbol{\omega(k)^T}\right)\mathbf{B}_j\boldsymbol{\tau}(z_i) = \mathbf{X}_{ij\omega}^T\mathbf{B}_j\boldsymbol{\tau}(z_i). \tag{A.4}$$

As noted in the main document, we use the penalized least squares approach to functional estimation proposed in Gertheiss et al. (2013) and adapted in Stallrich et al. (2020).

Tables A.51 through A.54 show the results from multiple stages of adaptive weighting, the SAFE procedure. After only one round, the differences in tuning parameter selection approaches disappear, giving the same results for APE, AR2, and Mod CV. The differences reappear again after three stages, but only for AR2 CV, which still overselects by a single coefficient for FR2, even after a fourth stage.

Table A.51: Variable selection results for EMG finger movements after one stage of adaptive weighting. FP indicates the total number of false positives in the model and Size is the total number of EMG signals contained in the model.

		FC1	FC2	FC3	FR1	FR2	FR3	Mean
APE	FP	0	0	0	0	1	1	0.33
	Size	2	2	2	2	4	3	2.5
AR2	FP	0	0	1	0	1	0	0.33
	Size	2	2	3	2	4	2	2.5
Mod	FP	0	0	0	0	1	1	0.33
	Size	2	2	2	2	4	3	2.5

Table A.52: Variable selection results for EMG finger movements after two stages of adaptive weighting. FP indicates the total number of false positives in the model and Size is the total number of EMG signals contained in the model..

		FC1	FC2	FC3	FR1	FR2	FR3	Mean
APE	FP	0	0	0	0	1	0	0.17
	Size	2	2	2	2	4	2	2.33
AR2	FP	0	0	0	0	1	0	0.17
	Size	2	2	2	2	4	2	2.33
Mod	FP	0	0	0	0	1	0	0.17
	Size	2	2	2	2	4	2	2.33

Table A.53: Variable selection results for EMG finger movements after three stages of adaptive weighting. FP indicates the total number of false positives in the model and Size is the total number of EMG signals contained in the model.

		FC1	FC2	FC3	FR1	FR2	FR3	Mean
APE	FP	0	0	0	0	0	0	0
	Size	2	2	2	2	3	2	2.17
AR2	FP	0	0	0	0	1	0	0.17
	Size	2	2	2	2	4	2	2.33
Mod	FP	0	0	0	0	0	0	0
	Size	2	2	2	2	3	2	2.17

Table A.54: Variable selection results for EMG finger movements after the four stages of adaptive weighting recommended by Stallrich et al. (2020). FP indicates the total number of false positives in the model and Size is the total number of EMG signals contained in the model.

		FC1	FC2	FC3	FR1	FR2	FR3	Mean
APE	FP	0	0	0	0	0	0	0
	Size	2	2	2	2	3	2	2.17
AR2	FP	0	0	0	0	1	0	0.17
	Size	2	2	2	2	4	2	2.33
Mod	FP	0	0	0	0	0	0	0
	Size	2	2	2	2	3	2	2.17

APPENDIX

B

SUPPLEMENTARY MATERIAL FOR CHAPTER 3

The supplementary material in the following appendix includes two sections. Section B.1 gives details in applying Algorithm 1 to find $\hat{\alpha}_\lambda$ and $\hat{\alpha}_{\lambda,\nu}$. This section also includes the general algorithm for Newton's Method that was used to develop Algorithm 1. In Section B.2, we provide additional simulation results that were left out of the main document in the interest of brevity. These results include additional methods considered for the logistic and Poisson regression settings covered in the main document such as the implementation of a 1SE Rule and Consistent Cross Validation. Also included are results of applying the α- and α_ν-Modifications to the MCP penalty in logistic regression.

B.1 Algorithm Formulae

In this section, we give the formulae for $\nabla h(x)$ and $\nabla^2 h(x)$ as can be found in Algorithm 1. This algorithm was developed using Newton's Method, found in Algorithm 4.

Algorithm 4 Newton's Method

Given: Start at some x; set some tolerance $\epsilon > 0$
repeat
  Compute Newton step: $\Delta x_{nt} \leftarrow \nabla^2 h(x)^{-1} \nabla h(x)$
  Compute Newton decrement: $\lambda^2 \leftarrow \nabla h(x)^T \nabla^2 h(x)^{-1} \nabla h(x)$
  Choose stepsize t by backtracking line search ▷ Pure: $t = 1$
  $x \leftarrow x - t\Delta x_{nt}$
until $\frac{\lambda^2}{2} \leq \epsilon$

We begin with the unpenalized α-Modification:

$$h(\beta_0, \alpha) = -\frac{\sum_i y_i(\beta_0 + \alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)} + \frac{\sum_i b(\beta_0 + \alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)} - \sum_i c(y_i, \phi) + P_\lambda(\hat{\boldsymbol{\beta}}_\lambda). \tag{B.1}$$

Now we find the first and second derivatives of $h(\beta_0, \alpha)$ with respect to both β_0 and α:

$$\frac{\partial h(\beta_0, \alpha)}{\partial \alpha} = \frac{\sum_i \Big(g^{-1}(\beta_0 + \alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda) - y_i\Big) \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda}{a(\phi)} \tag{B.2}$$

$$\frac{\partial h(\beta_0, \alpha)}{\partial \beta_0} = \frac{\sum_i \big(g^{-1}(\beta_0 + \alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda) - y_i\big)}{a(\phi)} \tag{B.3}$$

$$\frac{\partial^2 h(\beta_0, \alpha)}{\partial \alpha^2} = \frac{\sum_i (\mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)^2 (g^{-1})'(\beta_0 + \alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)} \tag{B.4}$$

$$\frac{\partial^2 h(\beta_0, \alpha)}{\partial \beta_0^2} = \frac{\sum_i \big((g^{-1})'(\beta_0 + \alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda) - y_i\big)}{a(\phi)} \tag{B.5}$$

$$\frac{\partial^2 h(\beta_0, \alpha)}{\partial \beta_0 \partial \alpha} = \frac{\sum_i (\mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)(g^{-1})'(\beta_0 + \alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)}. \tag{B.6}$$

Then:

$$\nabla h = \begin{bmatrix} \frac{\sum_i \left(g^{-1}(\beta_0+\alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)-y_i\right)}{a(\phi)} \\ \frac{\sum_i \left(g^{-1}(\beta_0+\alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)-y_i\right)\mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda}{a(\phi)} \end{bmatrix}, \tag{B.7}$$

and

$$\nabla^2 h = \begin{bmatrix} \frac{\sum_i \left((g^{-1})'(\beta_0+\alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)-y_i\right)}{a(\phi)} & \frac{\sum_i (\mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)(g^{-1})'(\beta_0+\alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)} \\ \frac{\sum_i (\mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)(g^{-1})'(\beta_0+\alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)} & \frac{\sum_i (\mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)^2(g^{-1})'(\beta_0+\alpha \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)} \end{bmatrix}. \tag{B.8}$$

Then the $(k+1)$th step for α and β_0 is:

$$\Delta_{k+1} = \nabla^2 h(\beta_{0,k}, \alpha_k)^{-1} \nabla h(\beta_{0,k}, \alpha_k). \tag{B.9}$$

Now, let $h(x) = h_\nu(x)$ to define the expressions in Algorithm 1 for the penalized α_ν-Modification. The objective function for our new penalized optimization problem (for an exponential family with canonical link function) when $P(\alpha) = |\alpha - 1|$ is:

$$h(\beta_0,\alpha) = -\ell(\alpha|\beta_0,\hat{\boldsymbol{\beta}}_\lambda,\mathbf{y},\phi) + \nu|\alpha-1| \tag{B.10}$$

$$= -\frac{\sum_i y_i(\beta_0+\alpha\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)} + \frac{\sum_i b(\beta_0+\alpha\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)} - \sum_i c(y_i,\phi) + \nu|\alpha-1| \tag{B.11}$$

$$\frac{\partial h(\beta_0,\alpha)}{\partial\alpha} = \frac{\sum_i \Big(g^{-1}(\beta_0+\alpha\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda) - y_i\Big)\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda}{a(\phi)} + \nu\text{sign}(\alpha-1) \tag{B.12}$$

$$\frac{\partial h(\beta_0,\alpha)}{\partial\beta_0} = \frac{\sum_i \big(g^{-1}(\beta_0+\alpha\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda) - y_i\big)}{a(\phi)} \tag{B.13}$$

$$\frac{\partial^2 h(\beta_0,\alpha)}{\partial\alpha^2} = \frac{\sum_i(\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda)^2(g^{-1})'(\beta_0+\alpha\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)} \tag{B.14}$$

$$\frac{\partial^2 h(\beta_0,\alpha)}{\partial\beta_0^2} = \frac{\sum_i \big((g^{-1})'(\beta_0+\alpha\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda) - y_i\big)}{a(\phi)} \tag{B.15}$$

$$\frac{\partial^2 h(\beta_0,\alpha)}{\partial\beta_0\partial\alpha} = \frac{\sum_i(\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda)(g^{-1})'(\beta_0+\alpha\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)}. \tag{B.16}$$

Then we have:

$$\nabla h(\beta_0,\alpha) = \begin{bmatrix} \frac{\sum_i \left((g^{-1})'(\beta_0+\alpha\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda)-y_i\right)}{a(\phi)} \\ \frac{\sum_i \left(g^{-1}(\beta_0+\alpha\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda)-y_i\right)\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda}{a(\phi)} + \nu\text{sign}(\alpha-1) \end{bmatrix}, \tag{B.17}$$

and

$$\nabla^2 h(\beta_0,\alpha) = \begin{bmatrix} \frac{\sum_i \left((g^{-1})'(\beta_0+\alpha\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda)-y_i\right)}{a(\phi)} & \frac{\sum_i(\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda)(g^{-1})'(\beta_0+\alpha\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)} \\ \frac{\sum_i(\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda)(g^{-1})'(\beta_0+\alpha\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)} & \frac{\sum_i(\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda)^2(g^{-1})'(\beta_0+\alpha\mathbf{x}_i^T\hat{\boldsymbol{\beta}}_\lambda)}{a(\phi)} \end{bmatrix}. \tag{B.18}$$

B.2 Additional Simulation Results

In this section, additional results from the logistic and Poisson regression simulation studies can be found. In several of the figures in the following results, the Hamming Distance between $\boldsymbol{\beta}^*$ and $\hat{\boldsymbol{\beta}}_\lambda$ was used to evaluate the variable selection.

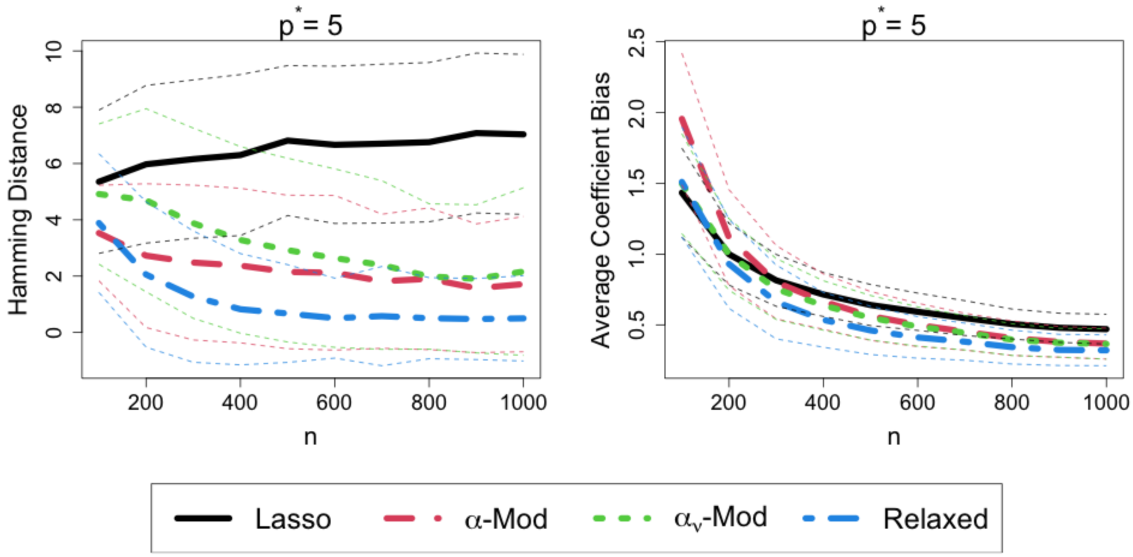

Figure B.1: Results from 500 replications of Logistic regression with independent predictors, using the Lasso penalty.

B.2.1 Logistic Regression Simulation Results

This section contains additional results from the logistic regression simulations described in the main document. Figure B.1 show the performance of the α- and α_ν-Modifications in competition with the traditional and Relaxed Lasso as n increases. Finally, Table B.3 shows the results from employing the α- and α_ν-Modifications on the Minimax Concave Penalty (Zhang 2010), rather than the Lasso penalty.

We also consider an even more difficult problem: logistic regression with correlated predictors. Table B.2 and Figure B.2 give these results.

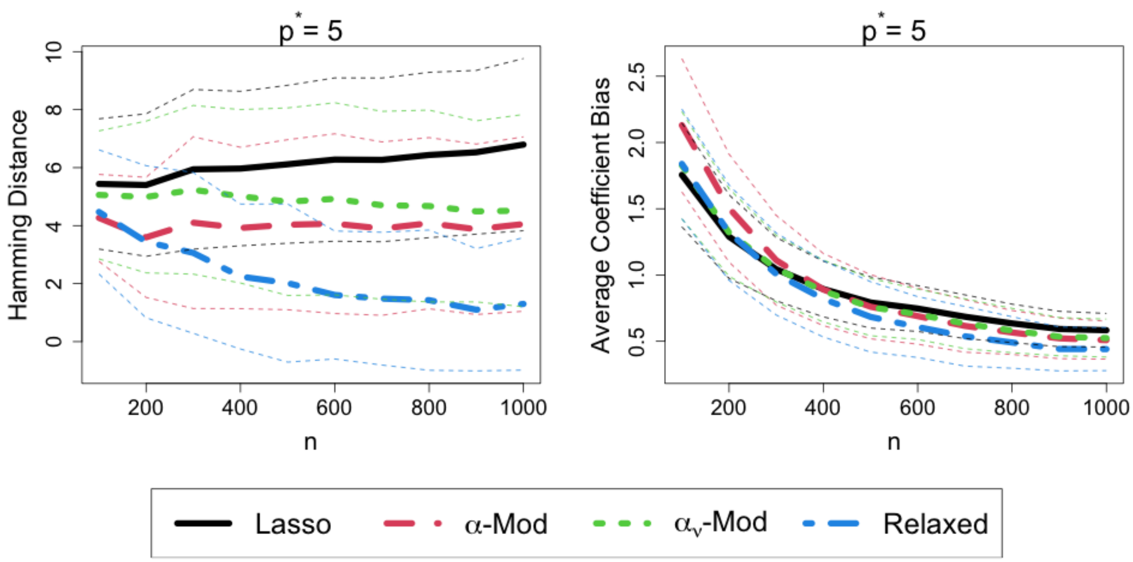

Figure B.2: Results from 500 replications of Logistic regression with correlated predictors ($\rho = 0.5$), using the Lasso penalty.

Table B.1: Average results from 500 replications of logistic regression data with $p^* = 5, p = 20$. Optimal models chosen with 5-fold CV using a 1SE Rule. Standard errors given in parentheses.

		α	CB	Dev	Size	FN	Time
$n = 100$	Lasso(Min)		1.43 (0.31)	2.87 (0.15)	9.46 (3.38)	0.45 (0.91)	0.24 (0.2)
	Lasso(1SE)		1.86 (0.32)	3.08 (0.14)	3.92 (2.85)	2.06 (1.81)	0.24 (0.2)
	EBIC		2.18 (0.23)	3.21 (0.12)	0.49 (1.16)	4.55 (1.04)	0 (0)
	α-Mod(Min)	3.55 (4.89)	1.96 (0.46)	2.99 (0.2)	3.67 (3.39)	2.43 (1.94)	1.87 (0.58)
	α-Mod(1SE)	8.38 (38.67)	2.12 (0.25)	3.15 (0.15)	0.95 (1.54)	4.13 (1.32)	1.87 (0.58)
	α_ν-Mod(Min)	1.45 (1.56)	1.5 (0.35)	2.9 (0.16)	8.37 (3.95)	0.77 (1.33)	43.94 (2.15)
	α_ν-Mod(1SE)	1.05 (0.37)	1.83 (0.4)	3.04 (0.18)	3.03 (2.76)	2.66 (1.8)	43.94 (2.15)
	Relaxed(Min)		1.51 (0.39)	2.89 (0.16)	6.63 (3.9)	1.13 (1.39)	12.47 (10.35)
	Relaxed(1SE)		1.79 (0.43)	3.03 (0.19)	2.66 (2.18)	2.78 (1.61)	12.47 (10.35)
	CCV(Min)		1.94 (0.52)	2.98 (0.19)	2.99 (2.43)	2.57 (1.7)	0.59 (0.17)
	CCV(1SE)		2.08 (0.33)	3.12 (0.18)	1.22 (1.64)	3.89 (1.42)	0.59 (0.17)
	Oracle ML		1.01 (0.46)	2.89 (0.11)			
	Lasso FPFN				3.16 (1.25)	1.86 (1.24)	
	Lasso FNFP				8.54 (3.46)	0 (0)	
$n = 500$	Lasso(Min)		0.64 (0.15)	2.93 (0.05)	11.81 (2.66)	0 (0)	0.36 (0.02)
	Lasso(1SE)		1.1 (0.18)	2.99 (0.05)	5.81 (1.2)	0 (0)	0.36 (0.02)
	EBIC		0.91 (0.2)	2.97 (0.05)	5.88 (1.06)	0 (0)	0.01 (0)
	α-Mod(Min)	1.65 (0.21)	0.56 (0.17)	2.93 (0.05)	7.15 (2.73)	0 (0)	5.36 (0.25)
	α-Mod(1SE)	3.29 (0.75)	0.82 (0.25)	2.96 (0.05)	4.97 (0.43)	0.08 (0.29)	5.36 (0.25)
	α_ν-Mod(Min)	1.43 (0.23)	0.55 (0.16)	2.93 (0.05)	7.92 (3.26)	0 (0)	126.15 (3.66)
	α_ν-Mod(1SE)	1.01 (0.06)	0.58 (0.23)	2.94 (0.05)	5.02 (0.48)	0.06 (0.27)	126.15 (3.66)
	Relaxed(Min)		0.46 (0.17)	2.93 (0.05)	5.67 (1.74)	0 (0.04)	12.29 (1.17)
	Relaxed(1SE)		0.55 (0.32)	2.95 (0.05)	4.78 (0.5)	0.24 (0.45)	12.29 (1.17)
	CCV(Min)		0.46 (0.19)	2.93 (0.05)	5.41 (1.33)	0 (0.04)	0.72 (0.05)
	CCV(1SE)		0.56 (0.32)	2.95 (0.05)	4.77 (0.51)	0.25 (0.47)	0.72 (0.05)
	Oracle ML		0.41 (0.14)	2.94 (0.05)			
	Lasso FPFN				4.99 (0.09)	0.01 (0.09)	
	Lasso FNFP				5.01 (0.09)	0 (0)	
$n = 1000$	Lasso(Min)		0.47 (0.11)	2.94 (0.03)	12.03 (2.84)	0 (0)	0.58 (0.03)
	Lasso(1SE)		0.91 (0.14)	2.98 (0.03)	5.53 (0.97)	0 (0)	0.58 (0.03)
	EBIC		0.69 (0.15)	2.96 (0.03)	5.88 (1.07)	0 (0)	0.01 (0)
	α-Mod(Min)	1.47 (0.16)	0.37 (0.11)	2.94 (0.03)	6.72 (2.4)	0 (0)	15.02 (0.85)
	α-Mod(1SE)	3.42 (0.75)	0.61 (0.18)	2.96 (0.03)	5.01 (0.18)	0 (0.04)	15.02 (0.85)
	α_ν-Mod(Min)	1.38 (0.17)	0.37 (0.11)	2.94 (0.03)	7.16 (2.97)	0 (0)	342.41 (17.12)
	α_ν-Mod(1SE)	1.01 (0.06)	0.36 (0.12)	2.95 (0.03)	5 (0.06)	0 (0.04)	342.41 (17.12)
	Relaxed(Min)		0.32 (0.11)	2.94 (0.03)	5.5 (1.51)	0 (0)	17.34 (1.41)
	Relaxed(1SE)		0.35 (0.23)	2.95 (0.04)	4.92 (0.3)	0.09 (0.28)	17.34 (1.41)
	CCV(Min)		0.32 (0.12)	2.94 (0.03)	5.42 (1.28)	0 (0)	0.98 (0.06)
	CCV(1SE)		0.35 (0.23)	2.95 (0.04)	4.92 (0.29)	0.09 (0.29)	0.98 (0.06)
	Oracle ML		0.29 (0.09)	2.95 (0.03)			
	Lasso FPFN				5 (0)	0 (0)	
	Lasso FNFP				5 (0)	0 (0)	

Table B.2: Average results from 500 replications of logistic regression data with $p^* = 5, p = 20$ and correlated predictors. Optimal models chosen with 5-fold CV using a 1SE Rule. Standard errors given in parentheses.

		α	CB	Dev	Size	FN	Time
$n = 100$	Lasso(Min)		1.76 (0.39)	2.9 (0.17)	7.81 (3.53)	1.31 (1.32)	0.31 (0.29)
	Lasso(1SE)		2.03 (0.24)	3.09 (0.15)	2.99 (2.72)	3.07 (1.58)	0.31 (0.29)
	EBIC		2.13 (0.22)	3.14 (0.17)	1.03 (1.59)	4.21 (1.14)	0.01 (0)
	α-Mod(Min)	3.61 (4.86)	2.13 (0.5)	2.99 (0.19)	3.29 (3.09)	2.99 (1.58)	2.3 (0.96)
	α-Mod(1SE)	9.11 (27.5)	2.25 (0.29)	3.11 (0.18)	0.92 (1.58)	4.31 (1)	2.3 (0.96)
	α_ν-Mod(Min)	1.68 (2.38)	1.83 (0.4)	2.92 (0.17)	6.76 (3.94)	1.65 (1.51)	52.31 (8.05)
	α_ν-Mod(1SE)	1.1 (0.91)	2.09 (0.31)	3.05 (0.18)	2.05 (2.31)	3.58 (1.44)	52.31 (8.05)
	Relaxed(Min)		1.84 (0.41)	2.92 (0.18)	5.58 (3.53)	1.95 (1.46)	17.08 (15.56)
	Relaxed(1SE)		2.05 (0.33)	3.04 (0.2)	2.11 (2.01)	3.49 (1.3)	17.08 (15.56)
	CCV(Min)		2.21 (0.6)	2.97 (0.19)	3.03 (2.56)	3.02 (1.42)	0.72 (0.25)
	CCV(1SE)		2.24 (0.32)	3.1 (0.18)	0.9 (1.21)	4.26 (0.93)	0.72 (0.25)
	Oracle ML		1.27 (0.61)	2.9 (0.13)			
	Lasso FPFN				2.35 (1.2)	2.83 (1.21)	
	Lasso FNFP				11.6 (4.34)	0 (0)	
$n = 500$	Lasso(Min)		0.79 (0.19)	2.94 (0.1)	11.11 (2.72)	0 (0.06)	0.43 (0.09)
	Lasso(1SE)		1.3 (0.23)	3 (0.1)	5.98 (1.54)	0.2 (0.42)	0.43 (0.09)
	EBIC		1.07 (0.31)	2.98 (0.11)	6.04 (1.71)	0.22 (0.59)	0.01 (0)
	α-Mod(Min)	1.38 (0.18)	0.76 (0.24)	2.94 (0.1)	9 (2.96)	0.02 (0.13)	5.91 (1.04)
	α-Mod(1SE)	2.8 (1.98)	1.24 (0.36)	3 (0.1)	4.76 (1.46)	0.65 (0.92)	5.91 (1.04)
	α_ν-Mod(Min)	1.16 (0.19)	0.76 (0.22)	2.94 (0.1)	9.8 (3.26)	0.01 (0.12)	141.04 (20.03)
	α_ν-Mod(1SE)	1 (0.05)	1 (0.37)	2.97 (0.1)	5.04 (1.43)	0.5 (0.75)	141.04 (20.03)
	Relaxed(Min)		0.68 (0.27)	2.94 (0.1)	6.92 (2.78)	0.05 (0.23)	15.89 (3.22)
	Relaxed(1SE)		0.96 (0.44)	2.97 (0.1)	4.51 (1.2)	0.76 (0.81)	15.89 (3.22)
	CCV(Min)		0.71 (0.31)	2.94 (0.1)	6.37 (2.17)	0.07 (0.27)	0.83 (0.15)
	CCV(1SE)		1 (0.45)	2.97 (0.1)	4.35 (1.28)	0.87 (0.92)	0.83 (0.15)
	Oracle ML		0.5 (0.18)	2.95 (0.1)			
	Lasso FPFN				4.57 (0.67)	0.43 (0.67)	
	Lasso FNFP				5.67 (1.3)	0 (0)	
$n = 1000$	Lasso(Min)		0.58 (0.13)	2.95 (0.09)	11.79 (2.97)	0 (0)	0.66 (0.1)
	Lasso(1SE)		1.08 (0.17)	2.99 (0.09)	5.89 (1.21)	0.04 (0.19)	0.66 (0.1)
	EBIC		0.79 (0.2)	2.97 (0.09)	6.33 (1.4)	0.02 (0.14)	0.01 (0)
	α-Mod(Min)	1.27 (0.16)	0.51 (0.14)	2.95 (0.09)	9.05 (3.01)	0 (0)	15.9 (2.81)
	α-Mod(1SE)	2.19 (1.1)	0.93 (0.28)	2.98 (0.09)	5.09 (0.9)	0.19 (0.43)	15.9 (2.81)
	α_ν-Mod(Min)	1.18 (0.16)	0.52 (0.14)	2.95 (0.09)	9.52 (3.31)	0 (0)	371.38 (49.97)
	α_ν-Mod(1SE)	1 (0.05)	0.69 (0.29)	2.97 (0.09)	5.13 (0.88)	0.17 (0.39)	371.38 (49.97)
	Relaxed(Min)		0.44 (0.16)	2.95 (0.09)	6.3 (2.28)	0 (0.04)	21.01 (3.21)
	Relaxed(1SE)		0.62 (0.4)	2.96 (0.09)	4.79 (0.79)	0.35 (0.52)	21.01 (3.21)
	CCV(Min)		0.44 (0.17)	2.95 (0.09)	6.08 (1.9)	0 (0)	1.07 (0.17)
	CCV(1SE)		0.63 (0.41)	2.96 (0.09)	4.75 (0.8)	0.38 (0.58)	1.07 (0.17)
	Oracle ML		0.35 (0.11)	2.95 (0.09)			
	Lasso FPFN				4.84 (0.37)	0.16 (0.37)	
	Lasso FNFP				5.24 (0.65)	0 (0)	

Table B.3: Average results from 500 replications of MCP logistic regression data with $p^* = 5, p = 20$.

		α	CB	Dev	Size	FN	Time
$n = 100$	MC+		2.07 (0.45)	2.98 (0.18)	3.02 (1.95)	2.81 (1.33)	0.16 (0.07)
	α-Mod MC+	2.6 (5.24)	2.24 (0.5)	3 (0.19)	2.2 (1.96)	3.31 (1.38)	1.2 (0.3)
	α_ν-Mod MC+	1.53 (1.96)	2.1 (0.52)	2.99 (0.18)	2.81 (2)	2.94 (1.35)	21.42 (3.25)
	Oracle OLS		1.22 (0.56)	2.9 (0.14)			
$n = 500$	MC+		0.63 (0.27)	2.94 (0.1)	5.96 (1.24)	0.04 (0.2)	0.32 (0.03)
	α-Mod MC+	1.02 (0.03)	0.64 (0.27)	2.94 (0.1)	6.02 (1.28)	0.02 (0.15)	3.21 (0.99)
	α_ν-Mod MC+	1 (0.02)	0.63 (0.26)	2.94 (0.1)	5.99 (1.26)	0.02 (0.15)	63.21 (11.6)
	Oracle OLS		0.49 (0.18)	2.95 (0.1)			

B.2.2 Poisson Regression Simulation Results

Figure B.3 and Table B.4 give additional results from the simulation study described in the main document.

Table B.4: Average results from 500 replications of Poisson regression data with $p^* = 5, p = 20$. Optimal models chosen with 5-fold CV using a 1SE Rule. Standard errors given in parentheses.

		α	CB	Dev	Size	FN	Time
$n = 100$	Lasso(Min)		0.68 (0.19)	0.84 (0.14)	9.63 (3.99)	0.67 (1.08)	0.16 (0.02)
	Lasso(1SE)		0.8 (0.19)	1.04 (0.18)	4.18 (3.19)	2.07 (1.84)	0.16 (0.02)
	EBIC		0.9 (0.18)	1.14 (0.19)	1.6 (2.06)	3.6 (1.7)	0 (0)
	α-Mod(Min)	2.09 (1.6)	0.83 (0.22)	0.89 (0.18)	6.23 (4.84)	1.66 (1.78)	1.19 (0.04)
	α-Mod(1SE)	4.62 (23.29)	0.88 (0.19)	1.03 (0.19)	2.69 (2.89)	2.93 (1.89)	1.19 (0.04)
	α_ν-Mod(Min)	1.57 (1.51)	0.72 (0.2)	0.87 (0.16)	7.96 (4.68)	1.11 (1.52)	38.76 (0.86)
	α_ν-Mod(1SE)	1.17 (0.52)	0.83 (0.21)	1.01 (0.2)	3.11 (2.89)	2.66 (1.83)	38.76 (0.86)
	Relaxed(Min)		0.72 (0.21)	0.86 (0.15)	6.37 (3.71)	1.24 (1.35)	5.98 (1.07)
	Relaxed(1SE)		0.8 (0.22)	0.97 (0.2)	3.35 (2.38)	2.37 (1.55)	5.98 (1.07)
	CCV(Min)		0.82 (0.24)	0.89 (0.17)	4.8 (3.35)	1.79 (1.65)	0.47 (0.06)
	CCV(1SE)		0.87 (0.22)	1.01 (0.2)	2.46 (2.43)	2.97 (1.74)	0.47 (0.06)
	Oracle ML		0.47 (0.18)	2.98 (0.2)			
	Lasso FPFN				3.13 (1.23)	1.94 (1.26)	
	Lasso FNFP				9.62 (4.38)	0 (0)	
$n = 500$	Lasso(Min)		0.31 (0.08)	0.92 (0.08)	10.97 (2.95)	0 (0)	0.28 (0.02)
	Lasso(1SE)		0.47 (0.11)	0.98 (0.08)	5.69 (1.09)	0.02 (0.16)	0.28 (0.02)
	EBIC		0.4 (0.12)	0.96 (0.08)	5.76 (1.11)	0.04 (0.39)	0.01 (0)
	α-Mod(Min)	1.34 (0.15)	0.29 (0.1)	0.92 (0.08)	8.14 (3.17)	0.01 (0.15)	4.32 (0.23)
	α-Mod(1SE)	2.39 (0.8)	0.41 (0.14)	0.95 (0.08)	4.96 (0.81)	0.2 (0.53)	4.32 (0.23)
	α_ν-Mod(Min)	1.28 (0.15)	0.28 (0.09)	0.92 (0.08)	8.21 (3.26)	0 (0.04)	119.41 (4.09)
	α_ν-Mod(1SE)	2 (0.41)	0.32 (0.15)	0.94 (0.08)	5.01 (0.82)	0.17 (0.48)	119.41 (4.09)
	Relaxed(Min)		0.24 (0.1)	0.93 (0.08)	5.97 (2.09)	0.02 (0.15)	8.22 (0.62)
	Relaxed(1SE)		0.3 (0.18)	0.94 (0.08)	4.66 (0.74)	0.4 (0.65)	8.22 (0.62)
	CCV(Min)		0.25 (0.12)	0.93 (0.08)	5.81 (2.08)	0.05 (0.24)	0.65 (0.06)
	CCV(1SE)		0.29 (0.18)	0.94 (0.08)	4.66 (0.74)	0.39 (0.7)	0.65 (0.06)
	Oracle ML		0.2 (0.07)	3.11 (0.11)			
	Lasso FPFN				4.95 (0.3)	0.05 (0.3)	
	Lasso FNFP				5.05 (0.25)	0 (0)	
$n = 1000$	Lasso(Min)		0.22 (0.06)	0.93 (0.07)	11.2 (3.19)	0 (0)	0.41 (0.02)
	Lasso(1SE)		0.38 (0.08)	0.97 (0.07)	5.37 (0.74)	0 (0)	0.41 (0.02)
	EBIC		0.28 (0.08)	0.95 (0.07)	5.68 (0.84)	0 (0)	0.01 (0)
	α-Mod(Min)	1.28 (0.11)	0.18 (0.06)	0.93 (0.07)	7.18 (2.83)	0 (0)	12.33 (0.96)
	α-Mod(1SE)	2.44 (0.5)	0.29 (0.1)	0.95 (0.07)	4.99 (0.18)	0.02 (0.15)	12.33 (0.96)
	α_ν-Mod(Min)	1.25 (0.11)	0.18 (0.06)	0.93 (0.07)	7.31 (3.01)	0 (0)	332.93 (26.18)
	α_ν-Mod(1SE)	2.24 (0.41)	0.19 (0.08)	0.94 (0.07)	5 (0.22)	0.02 (0.14)	332.93 (26.18)
	Relaxed(Min)		0.15 (0.06)	0.94 (0.07)	5.6 (1.86)	0 (0)	11.01 (0.62)
	Relaxed(1SE)		0.16 (0.11)	0.94 (0.07)	4.9 (0.32)	0.1 (0.32)	11.01 (0.62)
	CCV(Min)		0.15 (0.07)	0.94 (0.07)	5.4 (1.63)	0 (0.06)	0.82 (0.06)
	CCV(1SE)		0.17 (0.12)	0.94 (0.07)	4.89 (0.34)	0.11 (0.34)	0.82 (0.06)
	Oracle ML		0.13 (0.05)	3.12 (0.09)			
	Lasso FPFN				5 (0)	0 (0)	
	Lasso FNFP				5 (0)	0 (0)	

Table B.5 and Figure B.4 give the results from Poisson regression with correlated predictors.

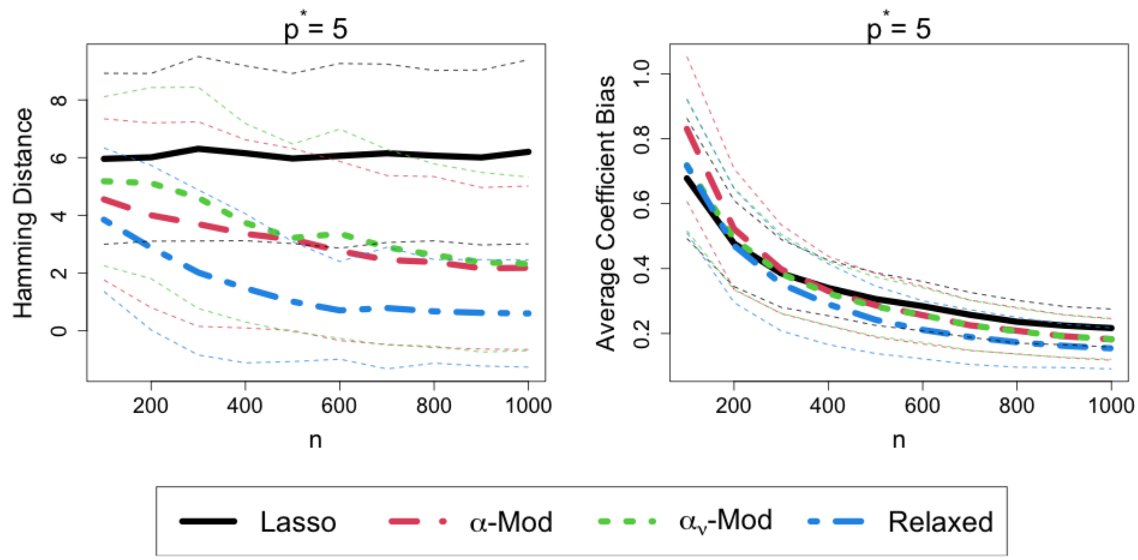

Figure B.3: Results from 500 replications of Poisson regression with independent predictors, using the Lasso penalty.

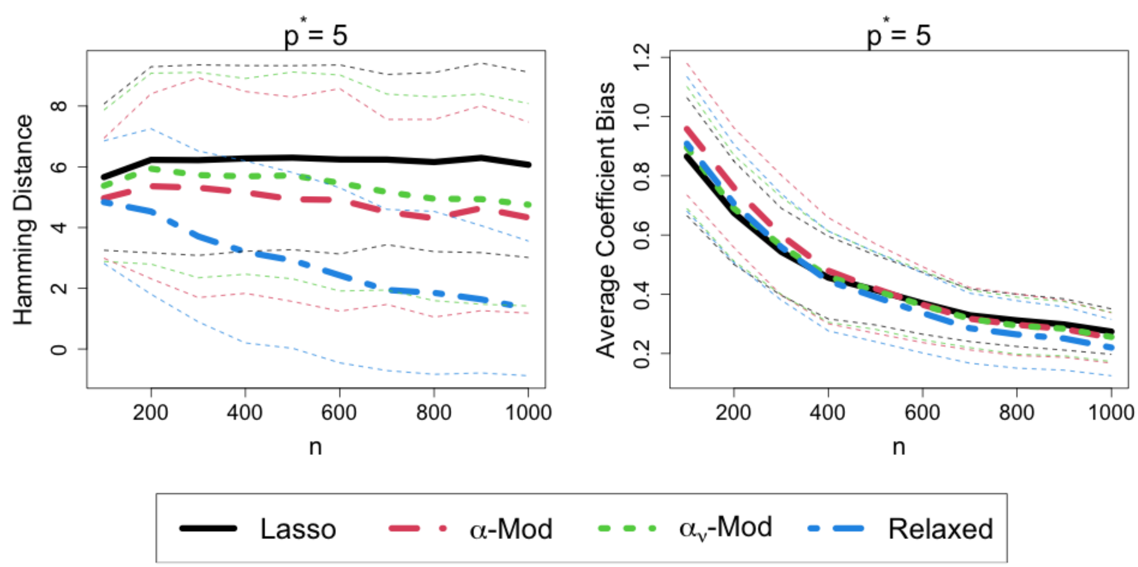

Figure B.4: Results from 500 replications of Poisson regression with correlated predictors ($\rho = 0.5$), using the Lasso penalty.

Table B.5: Average results from 500 replications of poisson regression data with $p^* = 5, p = 20$ and correlated predictors. Optimal models chosen with 5-fold CV using a 1SE Rule. Standard errors given in parentheses.

		α	CB	Dev	Size	FN	Time
$n = 100$	Lasso(Min)		0.87 (0.2)	0.88 (0.16)	6.78 (4.32)	1.94 (1.63)	0.19 (0.03)
	Lasso(1SE)		0.93 (0.14)	1.03 (0.17)	2.46 (2.96)	3.5 (1.63)	0.19 (0.03)
	EBIC		0.95 (0.13)	1.05 (0.16)	1.36 (2.04)	4.06 (1.35)	0 (0)
	α-Mod(Min)	2.05 (1.95)	0.96 (0.22)	0.92 (0.17)	4.13 (4.3)	2.92 (1.79)	1.23 (0.07)
	α-Mod(1SE)	4.55 (22.1)	1 (0.14)	1.02 (0.18)	1.53 (2.28)	3.96 (1.38)	1.23 (0.07)
	α_ν-Mod(Min)	1.48 (1.32)	0.9 (0.21)	0.9 (0.17)	5.74 (4.61)	2.32 (1.73)	39.55 (1.33)
	α_ν-Mod(1SE)	1.11 (0.73)	0.97 (0.15)	1 (0.18)	1.83 (2.59)	3.83 (1.43)	39.55 (1.33)
	Relaxed(Min)		0.91 (0.23)	0.9 (0.17)	4.78 (3.72)	2.53 (1.63)	7.76 (6.3)
	Relaxed(1SE)		0.96 (0.18)	1 (0.2)	1.84 (2.27)	3.77 (1.43)	7.76 (6.3)
	CCV(Min)		0.99 (0.25)	0.91 (0.17)	3.7 (3.44)	2.95 (1.61)	0.51 (0.08)
	CCV(1SE)		1 (0.17)	1 (0.18)	1.53 (2.13)	3.97 (1.34)	0.51 (0.08)
	Oracle ML		0.57 (0.22)	2.97 (0.65)			
	Lasso FPFN				2.16 (1.15)	3.11 (1.15)	
	Lasso FNFP				13.45 (4.54)	0 (0)	
$n = 500$	Lasso(Min)		0.41 (0.12)	0.93 (0.08)	11.26 (3.07)	0.02 (0.16)	0.33 (0.03)
	Lasso(1SE)		0.63 (0.16)	0.99 (0.09)	5.38 (1.85)	0.71 (0.97)	0.33 (0.03)
	EBIC		0.59 (0.23)	0.98 (0.08)	4.81 (2.46)	1.17 (1.6)	0.01 (0)
	α-Mod(Min)	1.26 (0.32)	0.42 (0.15)	0.93 (0.08)	9.69 (3.6)	0.12 (0.44)	4.29 (0.28)
	α-Mod(1SE)	2.82 (5.76)	0.62 (0.2)	0.98 (0.09)	4.23 (2.03)	1.3 (1.49)	4.29 (0.28)
	α_ν-Mod(Min)	1.1 (0.2)	0.41 (0.13)	0.93 (0.08)	10.6 (3.54)	0.06 (0.3)	120.45 (5.67)
	α_ν-Mod(1SE)	1.56 (0.6)	0.57 (0.21)	0.97 (0.09)	4.51 (2.07)	1.16 (1.38)	120.45 (5.67)
	Relaxed(Min)		0.39 (0.15)	0.93 (0.08)	7.59 (3.11)	0.16 (0.45)	10.98 (3.73)
	Relaxed(1SE)		0.53 (0.24)	0.96 (0.09)	4.2 (1.68)	1.24 (1.24)	10.98 (3.73)
	CCV(Min)		0.42 (0.18)	0.93 (0.08)	6.97 (3.11)	0.3 (0.7)	0.69 (0.05)
	CCV(1SE)		0.55 (0.23)	0.97 (0.09)	4.05 (1.8)	1.39 (1.33)	0.69 (0.05)
	Oracle ML		0.24 (0.09)	3.05 (0.53)			
	Lasso FPFN				4.02 (1.07)	0.99 (1.07)	
	Lasso FNFP				6.34 (1.75)	0 (0)	
$n = 1000$	Lasso(Min)		0.27 (0.08)	0.93 (0.07)	11.07 (3.05)	0 (0)	0.48 (0.05)
	Lasso(1SE)		0.48 (0.12)	0.97 (0.07)	5.73 (1.24)	0.1 (0.32)	0.48 (0.05)
	EBIC		0.36 (0.13)	0.95 (0.07)	6.17 (1.46)	0.07 (0.35)	0.01 (0)
	α-Mod(Min)	1.17 (0.11)	0.25 (0.08)	0.93 (0.07)	9.33 (3.14)	0 (0)	12.45 (1)
	α-Mod(1SE)	1.71 (0.6)	0.42 (0.15)	0.96 (0.07)	5.1 (1.07)	0.27 (0.63)	12.45 (1)
	α_ν-Mod(Min)	1.12 (0.09)	0.26 (0.08)	0.93 (0.07)	9.75 (3.34)	0 (0.04)	338.47 (24.55)
	α_ν-Mod(1SE)	1.55 (0.41)	0.35 (0.16)	0.95 (0.07)	5.14 (1.07)	0.26 (0.56)	338.47 (24.55)
	Relaxed(Min)		0.22 (0.09)	0.93 (0.07)	6.32 (2.22)	0.01 (0.1)	14.89 (2.35)
	Relaxed(1SE)		0.32 (0.2)	0.95 (0.07)	4.73 (0.91)	0.47 (0.66)	14.89 (2.35)
	CCV(Min)		0.22 (0.11)	0.93 (0.07)	6.22 (2.35)	0.03 (0.17)	0.91 (0.07)
	CCV(1SE)		0.3 (0.2)	0.95 (0.07)	4.76 (0.9)	0.43 (0.66)	0.91 (0.07)
	Oracle ML		0.16 (0.06)	3.04 (0.49)			
	Lasso FPFN				4.75 (0.51)	0.25 (0.51)	
	Lasso FNFP				5.32 (0.72)	0 (0)	

APPENDIX

C

SUPPLEMENTARY MATERIAL FOR CHAPTER 4

Chapter 4 gives the results from using the APE and AR2 approximation for the EMG data in the initial stage of SAFE—without any adaptive weighting—and the final stage—after four rounds of adaptive weighting. Here, the results of the intervening stages of SAFE using the approximated APE and AR2 approaches are given.

Table C.1: Model size, number of false positives (FP), and number of tuning parameter sampling points (Sampled) evaluated for the EMG data one stage of adaptive weighting. Performance is evaluated for observing the APE and AR2 curves over a grid of tuning parameters (Grid Search APE, Grid Search AR2), the approximation of the APE (Approximated APE) curve and the approximation of the AR2 curve (Approximated AR2) across a denser grid of tuning parameters.

		FC1	FC2	FC3	FR1	FR2	FR3	Mean
Size	Grid Search APE	2	2	2	2	4	3	2.5
	Sequential APE	2	3	5	4	4	3	3.5
	Grid Search AR2	2	2	3	2	4	2	2.5
	Sequential AR2	2	4	5	2	4	3	3.333
FP	Grid Search APE	0	0	0	0	1	1	0.333
	Sequential APE	0	1	2	2	1	1	1.167
	Grid Search AR2	0	0	1	0	1	0	0.333
	Sequential AR2	0	1	2	0	1	1	0.833
Min APE	Grid Search APE	0.073	0.084	0.093	0.197	0.15	0.179	0.13
	Sequential APE	0.065	0.075	0.079	0.181	0.133	0.167	0.117
	Grid Search AR2	0.076	0.089	0.107	0.204	0.142	0.193	0.135
	Sequential AR2	0.064	0.086	0.081	0.195	0.139	0.173	0.123
Min AR2	Grid Search APE	0.064	0.084	0.094	0.188	0.149	0.221	0.133
	Sequential APE	0.057	0.075	0.075	0.188	0.134	0.173	0.117
	Grid Search AR2	0.064	0.074	0.082	0.184	0.134	0.215	0.126
	Sequential AR2	0.055	0.076	0.07	0.184	0.133	0.176	0.116
Sampled	Grid Search APE	2000	2000	2000	2000	2000	2000	2000
	Sequential APE	84	89	75	80	89	84	83.5
	Grid Search AR2	2000	2000	2000	2000	2000	2000	2000
	Sequential AR2	82	104	91	75	84	88	87.333

Table C.2: Model size, number of false positives (FP), and number of tuning parameter sampling points (Sampled) evaluated for the EMG data with two stages of adaptive weighting. Performance is evaluated for observing the APE and AR2 curves over a grid of tuning parameters (Grid Search APE, Grid Search AR2), the approximation of the APE (Approximated APE) curve and the approximation of the AR2 curve (Approximated AR2) across a denser grid of tuning parameters.

		FC1	FC2	FC3	FR1	FR2	FR3	Mean
Size	Grid Search APE	2	2	2	2	4	2	2.333
	Sequential APE	2	2	3	2	3	2	2.333
	Grid Search AR2	2	2	2	2	4	2	2.333
	Sequential AR2	2	3	5	2	4	2	3
FP	Grid Search APE	0	0	0	0	1	0	0.167
	Sequential APE	0	0	1	0	0	0	0.167
	Grid Search AR2	0	0	0	0	1	0	0.167
	Sequential AR2	0	1	2	0	1	0	0.667
Min APE	Grid Search APE	0.073	0.085	0.093	0.198	0.146	0.183	0.13
	Sequential APE	0.062	0.073	0.077	0.182	0.134	0.184	0.119
	Grid Search AR2	0.077	0.086	0.101	0.204	0.164	0.278	0.152
	Sequential AR2	0.063	0.087	0.079	0.182	0.133	0.19	0.122
Min AR2	Grid Search APE	0.064	0.078	0.088	0.186	0.143	0.218	0.129
	Sequential APE	0.054	0.073	0.068	0.173	0.134	0.195	0.116
	Grid Search AR2	0.066	0.072	0.086	0.184	0.128	0.195	0.122
	Sequential AR2	0.055	0.076	0.069	0.178	0.13	0.191	0.117
Sampled	Grid Search APE	2000	2000	2000	2000	2000	2000	2000
	Sequential APE	94	89	91	84	80	75	85.5
	Grid Search AR2	2000	2000	2000	2000	2000	2000	2000
	Sequential AR2	87	94	113	98	124	87	100.5

Table C.3: Model size, number of false positives (FP), and number of tuning parameter sampling points (Sampled) evaluated for the EMG data with three stages of adaptive weighting. Performance is evaluated for observing the APE and AR2 curves over a grid of tuning parameters (Grid Search APE, Grid Search AR2), the approximation of the APE (Approximated APE) curve and the approximation of the AR2 curve (Approximated AR2) across a denser grid of tuning parameters.

		FC1	FC2	FC3	FR1	FR2	FR3	Mean
Size	Grid Search APE	2	2	2	2	3	2	2.167
	Sequential APE	2	2	3	2	3	2	2.333
	Grid Search AR2	2	2	2	2	4	2	2.333
	Sequential AR2	2	2	5	2	4	2	2.833
FP	Grid Search APE	0	0	0	0	0	0	0
	Sequential APE	0	0	1	0	0	0	0.167
	Grid Search AR2	0	0	0	0	1	0	0.167
	Sequential AR2	0	0	2	0	1	0	0.5
Min APE	Grid Search APE	0.073	0.084	0.093	0.198	0.144	0.176	0.128
	Sequential APE	0.063	0.073	0.086	0.184	0.137	0.168	0.119
	Grid Search AR2	0.076	0.086	0.096	0.203	0.148	0.183	0.132
	Sequential AR2	0.063	0.091	0.081	0.198	0.141	0.178	0.126
Min AR2	Grid Search APE	0.064	0.074	0.094	0.186	0.136	0.222	0.129
	Sequential APE	0.055	0.07	0.083	0.178	0.139	0.173	0.116
	Grid Search AR2	0.064	0.073	0.086	0.184	0.125	0.197	0.122
	Sequential AR2	0.055	0.078	0.07	0.192	0.133	0.178	0.118
Sampled	Grid Search APE	2000	2000	2000	2000	2000	2000	2000
	Sequential APE	90	83	87	77	93	74	84
	Grid Search AR2	2000	2000	2000	2000	2000	2000	2000
	Sequential AR2	102	85	78	75	90	90	86.667