Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Anthony V. Pileggi Date

Penalized Linear Regression and a Flexible Alternative

By

Anthony V. Pileggi
Master of Science

Biostatistics

Lance A. Waller, Ph.D.
Advisor

Vicki Hertzberg, Ph.D.
Committee Member

Mary Kelley, Ph.D.
Committee Member

Hunter Glanz, Ph.D.
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the Graduate School

Date

Penalized Linear Regression and a Flexible Alternative

By

Anthony V. Pileggi
B.S., Carnegie Mellon University, 2008

Adviser: Lance A. Waller, Ph.D.

An Abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Master of Science
in Biostatistics
2016

Abstract

Penalized Linear Regression and a Flexible Alternative

By Anthony V. Pileggi

We provide a comprehensive review of traditional and modern approaches to estimation and model selection in the linear regression framework. We are particularly interested in methods that estimate regression coefficients under various constraints, and the impact these constraints have on the resulting coefficient estimates and models selected. We propose a novel approach that allows for a more flexible penalty structure, and provide an estimation algorithm that utilizes linear programming. Finally, our flexible estimator is illustrated in various applications that exhibit spatial structure.

Penalized Linear Regression and a Flexible Alternative

By

Anthony V. Pileggi
B.S., Carnegie Mellon University, 2008

Adviser: Lance A. Waller, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Master of Science
in Biostatistics
2016

Contents

List of Figures

List of Tables

Chapter 1

Traditional and Modern Approaches to Estimation and Model Selection in Linear Regression

1.1 Introduction

Collecting data is easier than ever. As the amount of data we collect increases, we must either accept increased complexity of our models or develop more complex procedures to trim them. Given a large pool of potential predictors, we would like to identify the subset that explains the majority of the variability in a particular response variable. But traditional model selection techniques are rapidly becoming computationally infeasible as the number of variables increase, promoting the development of alternative approaches. We consider a class of penalized regression functions that encourage sparsity in the coefficient estimates by placing constraints on coefficient norms, yielding simultaneous parameter estimation and model selection. We assess the properties of these methods and consider recent extensions. Finally, we conduct detailed simulations to compare methods and make practical recommendations.

1.2 Notation

We begin by defining notation necessary for our development.

Vector Norms

The p-norm of a vector is defined as

$$||x||_p = \left(\sum_i |x_i|^p \right)^{1/p}$$

Special Cases:

L_1 Norm (i.e., 1-norm, Manhattan norm):

$$||x||_1 = \sum_i |x_i|$$

L_2 Norm (i.e., 2-norm, Euclidean norm):

$$||x||_2 = \left(\sum_i |x_i|^p \right)^{1/2} = \left(\mathbf{x}^{\mathrm{T}} \mathbf{x} \right)^{1/2}$$

L_∞ Norm (i.e., Maximum norm, Supremum norm):

$$||x||_\infty = \lim_{p \to \infty} \left\{ \left(\sum_i |x_i|^p \right)^{1/p} \right\} = \max \left\{ |x_1|, |x_2|, \ldots, |x_n| \right\}$$

1.3 Linear Regression (ordinary least squares)

1.3.1 Definition

Suppose we observe data $\mathcal{D}_i = \{y_i, x_{i1}, x_{i2}, \ldots, x_{ip}\}$ for $i = 1, 2, \ldots, n$, where y_i and x_i. represent the i^{th} values of the response and predictor variables, respectively. Let $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{ip}]^{\mathrm{T}}$ represent the $p \times 1$ predictor vector corresponding to the i^{th} observation. The standard linear regression model assumes the response variable is related to a linear

combination of predictors,

$$y_i = \beta_0 + \sum_{i=1}^{n} x_{ij}\beta_j + \epsilon_i = \beta_0 + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta} + \epsilon_i \qquad (i = 1, \ldots, n) \tag{1.1}$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_p]^{\mathrm{T}}$ is the $p \times 1$ parameter vector and β_0 represents the intercept term. By assuming each predictor is standardized ($\sum_i x_{ij}/n = 0$, $\sum_i x_{ij}^2/n = 1$) and y is centered ($\sum_i y_i/n = 0$), we can effectively ignore the intercept; under these assumptions, when all predictors equal zero the average value of the y_i's is also zero.

Consider a matrix formation of (1.1), where $\mathbf{y} = [y_1, y_2, \ldots, y_n]^{\mathrm{T}}$ is the $n \times 1$ response vector and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^{\mathrm{T}}$ is the $n \times p$ matrix of predictors. The linear regression model in matrix form is

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

or, if we want to make explicit that the model is conditional on $\mathbf{X}$ (i.e., $\mathbf{X}$ is assumed fixed and known without appreciable error),

$$E(\mathbf{y}|\mathbf{X}) = \mathbf{X}\boldsymbol{\beta}$$

where the error term has vanished because we assume $E(\epsilon_i) = 0$.

1.3.2 Estimation

To find the best estimates for the regression coefficients, we first need to determine how to quantify what is *best*. According to ordinary least squares (OLS) this criteria is the error (i.e., residual) sums of squares,

$$\mathrm{SSE}(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \tag{1.2}$$

There are plenty of other alternatives, but there are a few good reasons why least squares regression is popular. First, the derivative with respect to $\boldsymbol{\beta}$ can be easily derived because

SSE $(\boldsymbol{\beta})$ is smooth and continuous, leading to the familiar *normal equations*,

$$\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)\boldsymbol{\beta} = \mathbf{X}^{\mathrm{T}}\mathbf{y}. \tag{1.3}$$

Assuming that $\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$ exists and rearranging terms yields the closed form solution for the regression coefficients,

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}. \tag{1.4}$$

This leads to a second reason why least squares is so popular - the OLS solution is *identical* to the solution obtained via maximum likelihood estimation, assuming the errors follow a Normal distribution with homoscedastic variance (i.e., $\epsilon_i \sim^{\text{i.i.d.}} \text{Normal}\left(0, \sigma^2\right)$). Therefore, we know least squares solutions share enticing statistical properties that have been studied extensively in the literature, including: consistency, efficiency, and unbiasedness. Briefly, consistency indicates that the estimator will hit the target as the sample size approaches the population ($n \to \infty$), efficiency indicates that the estimator has the smallest variance among all possible unbiased estimators (i.e., it achieves the Cramer-Rao lower bound), and unbiasedness indicates that $E\left(\hat{\boldsymbol{\beta}}\right) = \boldsymbol{\beta}$. It is important to note that a biased estimator can be consistent; in fact, recent interest has shifted towards biased estimators as a means of ultimately obtaining more stable estimates.

Another important point is that OLS is only unbiased if the model is correct (i.e., $\mathbf{X}$ contains the relevant covariates). Obviously, it becomes more difficult to identify the correct model as the number of potential predictors increase. Although unbiasedness is an appealing property, the difficulty associated with guaranteeing this property in practice is cause for concern; as the number of potential predictors grows large, recovering the true model becomes increasingly difficult. Given a likely incorrect model, allowing for some bias may help to reduce the variability associated with OLS estimates.

In general, there is a bias-variance tradeoff; the error in model-based predictions can be decomposed into three main components, bias, variance, and the irreducible error. As we cannot hope to remove error inherent to the problem, we focus on the bias and variance components. Unbiased estimates have the largest variance, such that small changes to

the data will lead to large changes in the coefficient estimates; however, the expectation of unbiased estimates is exactly the targeted paramter. Allowing for some bias in the estimates can help to avoid overfitting, such that the estimates do not react so drastically to small changes in the observed data. Biased estimates will tend to be less variable, but the resulting estimates will deviate from the truth. There is a trade-off here: unbiased estimates have to much variability, relying heavily on the data, while completely biased estimates have no variability but ignore the data completely. There is conceivably a sweet spot, such that estimates are slightly biased to reduce variability while still yielding accurate and interpretable estimates. Variable selection techniques are one approach for tackling the bias-variance trade-off; predictions from a model with all possible predictors will vary considerably (i.e., overfit), while an intercept-only model will yield predictions that are stable against minor changes in the data.

1.3.3 Variable Selection

In general, least squares regression will never yield coefficient estimates that are *exactly* equal to zero. Therefore, all predictors are assumed to have some estimated degree of association (albeit small) with the outcome, and every predictor contributes to subsequent predictions. However, coefficient estimates that are sufficiently small may be extraneous, unimportant, or negligible. In other words, not all potential predictors must be related to the outcome. Such variables associated with 'small' parameter estimates may be removed from consideration in the model selection process, yielding a more parsimonious model. This is the basic idea behind model selection – reducing model complexity by estimating coefficients for only the subset of variables that are significantly related to the outcome.

Some analyses are only interested in prediction, and therefore one could argue that it would be best to provide as much information as possible, regardless of how useless or redundant it may be. After all, the model fit always appears to improve (or remain unchanged) as additional predictors are added when model fit is measured using R^2 (i.e., the proportion of variability in the response variable that is explained by the linear relationship with the set of predictors), regardless of their association with the response. A common misconception, this seemingly positive result is isolated to the set of data from which the

estimates were obtained (i.e., the training set); if there is any interest in using the model for predicting future observations, then overfitting (i.e., including unimportant predictors) should be avoided. According to the bias-variance tradeoff, overly complex models are sensitive to the data such that small changes in the data can lead to large changes in the coefficient estimates. This instability does not bode well for subsequent predictions made outside of the training data. The value of a parsimonious model should not be overlooked, even when building predictive models.

For simply comparing two nested models (i.e., a reduced model containing a subset of predictors in the full model), formal hypothesis testing can be used in the form of the Partial F-test

$$F = \frac{(\text{SSE}_r - \text{SSE}_f) \,/\, (p_f - p_r)}{\text{SSE}_f / \,(n - p_f - 1)} \tag{1.5}$$

where SSE_f and SSE_r represent the error sums of squares for the full and reduced model, respectively. Of course, rarely are we simply comparing between two nested models in practice. Further, as the number of models under consideration increases, so does the number of hypothesis tests and concerns regarding type I error inflation.

Often we are interested in comparing non-nested models. If the number of potential predictors is small, we can actually fit all possible regression models and choose the one that fits best. Unfortunately, if we use R^2 as our criteria then we will always pick the model with the most covariates; ideally, the fit would be evaluated on a separate test data set. Popular alternatives for evaluating fit (sans test data) include a penalty term for model complexity, and include: Akaike information criterion (AIC), Bayesian information criterion (BIC), and Mallows' C_p. Although not interpretable as stand-alone values, relatively lower values of AIC/BIC imply a better fit. While both are based on a penalized log-likelihood value (with different 0-norm penalties), under appropriate assumptions the AIC is consistent in estimating the regression function [1] while the BIC is consistent in model selection [58]. Moreover, the BIC has a more aggressive penalty for complexity relative to AIC. AIC is based on asymptotic results, so a correction is needed for small-samples (AICc) [33] In contrast, when using Mallow's C_p the models considered to be best have values close to the number of variables included in the model [46], causing some difficultly when making

comparisons between models of varying complexity.

Alternatively, we can use an iterative approach to selecting the best model based on adding or removing predictors and assessing improvements in the model fit; automated algorithms for variable selection were proposed as early as 1960 [18]. These p-value driven methods rely on F-statistics (Equation 1.5) derived from the potential inclusion/exclusion of a variable, given all other variables already in the model. More specifically, for a predefined p-value threshold and starting point, variables are iteratively added or removed from the model and refit until no coefficient's p-value meets the threshold. Forward selection begins with no variables and sequentially adds them while backward selection starts will all variables and removes one at a time; stepwise selection starts with no variables, but can both add or remove a variable at each step. These iterative selection schemes are considered *greedy* algorithms because the optimal choice is made at each step toward the overall goal of finding the optimal solution. A similar but more computationally intensive method, forward stagewise uses the same approach with smaller 'steps' at each iteration; although still *greedy*, stagewise selection is designed to mitigate some of the problems associated with correlated predictors [17].

1.3.4 Shortcomings

There are numerous issues associated with traditional model selection techniques. First, the aforementioned techniques are based on OLS estimates; therefore, we can anticipate problems whenever OLS estimates behave poorly. This includes situations where X is not full rank, which is always true when there are more predictors than observations (i.e., $p > n$). Not only are estimates not unique, but they are also susceptible to sign-flipping; i.e., there exists solutions across which any given coefficient will change sign [64]. As a consequence, the resulting interpretation of regression coefficients can be misleading. Of course, this lack of a unique solution may not apply to some linear combinations of coefficients (i.e., contrasts). An additional concern regarding OLS estimates is that correlated predictor variables (i.e., multicollinearity) can result in wildly variable parameter estimates. Obviously, this causes model selection results to be comparably unreliable.

For problems with a large number of predictors, traditional methods are not acceptable.

Considering all possible regression models is not computationally feasible; for example, with just 30 predictors there are over one billion possible models from which to chose. Further, p-value driven methods are influenced by multicollinearity, which itself is increasingly likely as the number of predictors increases. This sensitivity is partially ameliorated by methods that reduce the step size between iterations (e.g., forward stagewise), decreasing the likelihood that correlated predictors counterweight themselves. However, there is a trade-off between computational demand and step size, which can make these methods less appealing in high dimensional settings.

If the true interest is in the resulting predictions of outcomes, as opposed to accurate estimation of associations between the outcome and individual features, then OLS can be used regardless of the rank of $\mathbf{X}$. However, OLS-based predictions tend to have poor external validity; in the context of the bias-variance trade-off, OLS estimators sacrifice increases in variability in order to provide unbiased estimates. This makes them particularly sensitive to the data, especially when ignoring model selection (i.e., including all predictors), and leads to overfitting and poor out-of-sample prediction. Therefore, appealing alternative approaches to model selection may consider allowing for some bias in estimates, in order to stabilize both estimates and predictions with regard to small changes in the data. For example, the James-Stein estimator is a biased estimator capable of outperforming the maximum likelihood estimator [59].

1.4 Penalized Linear Regression

1.4.1 Definition

In contrast to OLS, consider a class of loss functions of the form

$$loss(\beta) = ||y - X\beta||_2^2 + \lambda P(\beta), \tag{1.6}$$

where tuning parameter $\lambda > 0$ and penalty function $P(\cdot)$ may both be vectors. Note that OLS is actually a special case when $\lambda = 0$. In general, this class of loss functions will yield biased estimates for any $\lambda > 0$; however, these biased estimates should be less variable than

those based on OLS, thanks to the bias-variance trade-off. Further, the exact form of $P(\beta)$ will influence the type of bias induced, which can have a substantial impact on the resulting structure of the regression coefficients.

1.4.2 Ridge Regression (L_2)

Ridge regression [31] was originally proposed as an alternative to OLS when predictors are correlated. The Ridge regression estimator takes the form:

$$\hat{\boldsymbol{\beta}}_\lambda = \underset{\beta}{\operatorname{argmin}} \left\{ ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda \, ||\boldsymbol{\beta}||_2^2 \right\} \ , \quad \lambda \geq 0 \tag{1.7}$$

Actually, a closed-form solution exists:

$$\hat{\boldsymbol{\beta}}_\lambda = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda \mathbf{I}_p\right)^{-1} \mathbf{X}^{\mathrm{T}}\mathbf{y} \tag{1.8}$$

As $\lambda \to 0$, the Ridge estimate approaches the OLS estimate; increasing values of λ shrink the coefficients towards 0, introducing bias in the estimates while reducing variance. However, coefficients will never become exactly zero as λ increases, so Ridge regression cannot be used as a tool for variable selection.

1.4.3 Lasso (L_1)

1.4.3.1 Motivation

If model selection is the goal, then an intuitive approach would be to impose a penalty based on the number of nonzero coefficients; specifically, $P(\beta) = \sum_i \beta_i^0 = \#\{\beta \neq 0\}$. This penalty function, also called a L_0 penalty, is the basic approach behind the standard AIC and BIC criteria; model complexity is penalized directly. As previously discussed, this approach is not computationally feasible for a large number of predictors. Imposing a L_0 penalty yields a non-convex loss function with discontinuity at the origin and potential identifiability problems, resulting in optimization challenges. Similar optimization difficulties exist for any L_q penalty, $q < 1$. However, any $q \leq 1$ will encourage sparsity in the coefficient estimates [40], by forcing smaller, insignificant estimates to be exactly zero. This suggests

utilizing a L_1 penalty ($P(\beta) = \sum_i |\beta_i|$) for model selection.

1.4.3.2 Definition

Originally introduced by Tibshirani in 1996 [60], the Lasso imposes an L_1 penalty on the loss function such that the estimator takes the form:

$$\hat{\boldsymbol{\beta}}_\lambda = \underset{\beta}{\operatorname{argmin}} \left\{ ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda \, ||\boldsymbol{\beta}||_1 \right\} \ , \quad \lambda \geq 0$$

By controlling λ, the Lasso performs simultaneous estimation and variable selection, setting some coefficient estimates to exactly zero to indicate unimportant variables. It is specifically suited for problems where there is a small-to-moderate number of significant predictors, i.e., situations where parsimony is desirable. Further, it can be utilized in high dimensional settings by selecting λ such that $\sum_i \mathrm{I}\left(\hat{\beta}_{j,\lambda} \neq 0\right) \leq rank(\mathbf{X})$.

In the special case where X is orthonormal, Lasso estimates are equivalent to soft threshold estimates: $\hat{\boldsymbol{\beta}}_\lambda = \text{sign}(\hat{\boldsymbol{\beta}}_{ols})(|\hat{\boldsymbol{\beta}}_{ols}| - \lambda)$. This provides some intuition behind the Lasso – coefficients that are sufficiently small are forced to be exactly zero. This is not unlike standard model selection techniques, where sufficiently insignificant predictors are removed from the model. And while there is no closed form solution for the Lasso due to the non-differentiable L_1 penalty, the problem remains convex. It was originally formulated as a quadratic programming problem (QP), but the solution algorithm could take as many as 2^p iterations to converge for a single value of λ [60]. Because the optimal solution to the Lasso essentially involves joint optimization of (β, λ), the QP approach is not computationally feasible for large values of p. Thus, the Lasso was not utilized for large p or high dimensional settings until the development of the least angle regression algorithm [17].

1.4.3.3 Computation

Least angle regression (LARS) is an efficient algorithm for computing the regression coefficients along a path of values for tuning parameter λ [17]. Least angle regression is conceptually similar to forward stagewise selection, but significantly faster thanks to math-

ematical calculation of the size of each step (as opposed to taking a large number of very small steps). Briefly, at each step the predictor most correlated with the current residuals (i.e., the residuals from a model fit based on the previous step) is entered into the model, and the solution vector then moves in a direction equiangular to all included predictors (Table 1.1). This approach is in stark contrast to stepwise selection, which takes a single large step towards the unconstrained OLS solution for each predictor introduced into the model; LARS is a less greedy version of stepwise selection [17]. A modified version of LARS yields the entire Lasso solution path in a comparable number of computations to a full OLS fit. Conceptually, LARS is a 'smarter' version of forward stagewise regression that utilizes mathematical formulae to reduce repetitive steps, often occurring when many small increments are taken for a single variable. This provides a compromise between the stability of stagewise regression and the speed of stepwise regression. In fact, the entire LARS solution path requires an equivalent number of computations to OLS. Modification of LARS, by imposing a sign-consistency constraint on nonzero coefficients and corresponding current correlations, yields the entire Lasso solution path [17].

Table 1.1: Least Angle Regression (LARS) Algorithm

1. Begin with all coefficients equal to zero ($\beta_j = 0, \ \forall j$).
2. Identify the predictor ($\mathbf{x}_j$) most correlated with response ($\mathbf{y}$).
3. Move β_j in the direction of $sign\left(corr\left(\mathbf{x}_j, \mathbf{y}\right)\right)$
   - Take the residuals along the way ($\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}}$).
   - Stop when another predictor ($\mathbf{x}_k$) is as correlated with $\mathbf{r}$ as ($\mathbf{x}_j$).
4. Move (β_j, β_k) in their joint least squares direction until another predictor ($\mathbf{x}_m$) has as much correlation with the residual ($\mathbf{r}$).
5. Continue until all predictors are in the model, terminating at the full OLS solution.

There are numerous software packages available for fitting the Lasso. In R, the `lars` and

`glmnet` packages are available. There is also a `glmnet` package for MATLAB. For fitting the Lasso in SAS, the `GLMSELECT` procedure is recommended.

1.4.3.4 Theoretical Details

There is substantial work regarding theoretical properties of the Lasso. In general, there are three main considerations: (1) Does it provide accurate predictions?, (2) Does it provide accurate coefficient estimates?, and (3) Does it consistently select the true model?

Results originally by Greenshtein and Ritov [27], and expounded upon by Buhlmann and van de Geer [12], suggest the Lasso is consistent for prediction when $p \gg n$, assuming: fixed X with $n \to \infty$, mild error assumptions, and a sparse coefficient vector such that $||\boldsymbol{\beta}_0||_1 = o\left(\sqrt{n/\log p}\right)$. This result, called *persistency*, suggests that introducing additional, potentially extraneous covariates does not adversely affect prediction estimates. Therefore, no harm can come from considering all potential predictors, even when p is large. Further, it is especially appealing because no strict conditions are placed on the design matrix. In contrast, L_q estimation consistency, $q \in \{1, 2\}$, requires restricted eigenvalue assumptions [9]; while not overly restrictive, they can be difficult to check in practice.

Model selection consistency requires relatively more strict design assumptions, in the form of the neighborhood stability condition [47] or the irrepresentable condition [76]. Nearly equivalent, these conditions require correlations between important and unimportant predictors to be relatively small, and are applicable when p grows (at a rate) larger than n. An additional assumption for model selection consistency requires the important predictors to have coefficients that are sufficiently large (i.e., a β-min assumption). It has been pointed out that these assumptions are unlikely to hold in practice [62]. For example, multicollinearity issues are prevalent in genomic and neuroimaging data. Alternatively, we can assume the design matrix conforms to the more relaxed restricted eigenvalue assumptions (retaining the β-min assumption) and conclude that the Lasso has the *variable screening property*. More formally, if $\hat{S}$ denotes the selected model and S_0 denotes the true model, then $S_0 \subseteq \hat{S}$ with high probability [62].

Results regarding model selection consistency make the assumption that tuning parameter λ follows a certain rate, while it is typically treated as fixed in practice. For example,

results from Knight and Fu [40] suggest λ should decay as the number of observations increase, assuming a fixed number of variables. When attempting to select the optimal λ in practice, results suggest that utilizing cross-validated prediction accuracy yields the true model with a probability less than 1 [41, 47]; due to estimation bias, the inclusion of additional noise variables can often improve prediction. Therefore, while theoretical results make rate-assumptions for λ that are not reasonable in practice, we can still be reasonably confident in the variable screening property of the Lasso; although we may include negligible predictors, we are almost guaranteed to not miss the important covariates.

The uniqueness of Lasso estimates has been studied more recently. Similar to OLS, Lasso estimates are always unique when X is full rank. However, when $p > n$ the Lasso is not a strictly convex problem. Fortunately, a unique solution is guaranteed when predictors are continuous; this result is extended to any differentiable, strictly convex loss function, including logistic and Poisson regression [64]. If predictors are discrete, there are some additional conditions under which uniqueness can be satisfied [64]. In contrast to OLS, non-unique Lasso solutions do not suffer from sign-inconsistency, mitigating the potential for an incorrect interpretation of a coefficient direction [64]. However, some coefficients may be nonzero for some solutions and zero for others; in these cases, bounds can be calculated to determine if a variable is 'dispensable' [64].

1.4.3.5 Choosing λ

By utilizing LARS, the entire Lasso solution path can be easily computed. However, the appropriate value for the tuning parameter λ must still be selected, which is essentially a surrogate for model selection. Selecting λ is not trivial, and how the selection is made will have an influence on the resulting estimates. If the dataset is large enough, then a simple approach is to partition the data such that one set is only used for estimation (training), a separate set is used to select the optimal value of λ (tuning), and the last set is used to evaluate the model's prediction accuracy (test). These partitions are selected randomly, so the results will be slightly different each time.

A common approach to selecting λ is via k-fold cross-validation, where $\hat{\lambda}$ is chosen as the minimizer of the mean squared prediction error (MSPE). Specifically, for each value of λ

along a coefficient path, estimates from the $k-1$ folds (excluding the i^{th}) are used to predict observations in the i^{th} fold, $i = 1, \ldots, k$. Essentially, cross-validation is an iterative version of the simple approach based on partitioning the data. Results suggest this approach is consistent with the variable screening property of the Lasso [41]. A general heuristic of selecting the optimal λ, called the 'one standard error rule', is to set it equal to a value that achieves the least complex (i.e., smallest) model within one standard error of the estimated minimum [28].

It is recommended that 5 or 10 folds are used for cross-validation, although these recommendations are more empirical than based on theoretical derivations. While leave-one-out (n-fold) cross-validation may seem like the optimal choice, the resulting error estimates are obtained by averaging many positively-correlated quantities. As a consequence, the error estimates from leave-one-out CV are highly variable. In the spirit of the bias-variance tradeoff, allowing for less overlap in the training data (by setting $k = 5$ or 10) yields less variable error estimates.

A concern that is especially relevant when analyzing high-dimensional data, cross-validation can be time-consuming because it requires k separate Lasso fits (i.e., one fit per fold). And if the number of observations is small relative to p, both cross-validation and data-splitting techniques may be untenable for selecting the tuning parameter. To avoid overfitting and computational bottlenecks , one approach is to directly select the optimal value for λ based on a penalized information criteria (e.g., AIC, BIC).

Another approach for selecting λ is via generalized cross-validation [60]. For linear models in general, generalized cross-validation is defined as:

$$\text{GCV} = \frac{1}{n} \sum_{i=1}^{n} \left[\frac{y_i - \hat{f}(x_i)}{1 - \text{trace}(\mathbf{S})/n} \right]^2$$

where $\hat{f}$ refers to estimates obtained from a linear fitting method and $\mathbf{S}$ is the orthogonal projection onto the column space of $\mathbf{X}$, or $\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$. For the Lasso, this approach is derived using a linear approximation to the Lasso estimate, and involves writing the solution as a Ridge regression estimator, $\hat{\beta} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda \mathbf{W}^{-}\right)^{-1} \mathbf{X}^{\mathrm{T}}\mathbf{y}$, where $\mathbf{W} = diag\left(\left|\hat{\beta}_j\right|\right)$ and $\mathbf{W}^{-}$ denotes a generalized inverse. Noting that $\mathbf{S} = \mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda \mathbf{W}^{-}\right)^{-1} \mathbf{X}^{\mathrm{T}}$, the generalized

cross-validation style statistic for the Lasso is obvious.

One concern regarding cross-validation and other data-partitioning approaches is that the resulting solutions are not guaranteed to be identical, even when applied to identical sets of data. This fact raises concerns regarding the reliability of the resulting coefficient estimates, made more evident by the fact that there is not a generally accepted approach for obtaining standard error estimates. There are approaches for selecting the tuning parameter that involve data resampling schemes as a means of evaluating estimate stability. Stability selection [48] involves subsampling the data (typically, taking many samples of size $n/2$ without replacement), repeatedly fitting the Lasso to a sequence of λ values, and ultimately yielding selection probabilities for each covariate as a function of λ. Primarily used for selecting the appropriate model, a subsequent fit to the selected covariates can yield coefficient estimates. Meinshausen also proposed a Randomized Lasso, which involves randomly reweighting the predictors at each iteration; although a seemingly naive idea, it performs quite well if a reasonable number of iterations are considered [48]. The BoLasso is a nearly identical idea to stability selection, but instead considers taking bootstrap samples of size n with replacement [7]. The main difference appears to be that the theoretical results of the BoLasso regarding consistent variable selection rely on a value of λ that vanishes at a rate that is much faster than is typically assumed; ostensibly, the BoLasso runs the risk of including irrelevant noise variables when λ is chosen too large. In contrast, if λ is chosen too large when applying stability selection to the Randomized Lasso, it is more likely that a few important predictors will be missed [48]. Obviously, the preferable outcome depends on the specific application and analysis goals.

1.4.3.6 Drawbacks

In discussing properties of the Lasso, a number of drawbacks have been brought to light. First, the design matrix must be well-behaved – typically, this assumption is satisfied when predictor correlations are sufficiently small. However, in the presence of high multicollinearity much of Lasso theory breaks down; predictions may be accurate, but there is no guarantee of accuracy for estimation or variable screening. This is particularly concerning because multicollinearity pervades many high-dimensional applications (e.g., genomics, imaging).

Empirical studies show the Lasso performs poorly when variables are correlated, arbitrarily selecting a small subset of predictors from a correlated group [78]. A common approach to alleviate predictor collinearity concerns is to consider orthogonalizing the design matrix prior to fitting (e.g., principal components analysis); however, this yields coefficient estimates that are difficult to interpret in the context of the original predictors and useless if model selection is our goal. Although the Lasso solution is sparse in the orthogonalized space, sparsity is not retained after back-transforming the coefficient estimates to the original predictor space.

Additionally, the Lasso tends to select additional noise variables, such that the true model is not captured with probability one. This may be partly due to the combination of the Lasso providing biased coefficient estimates and λ being optimized via cross-validated MSPE [41]. This bias also causes the Lasso to perform poorly when the true coefficient vector is not sparse or nonzero coefficients are sufficiently small. In addition, although applicable in high dimensional settings, the number of nonzero coefficients is bounded by the sample size.

Finally, the Lasso cannot properly account for structure in the variables. For example, when modeling a categorical variable we use a set of dummy variables, which should all be included or excluded simultaneously. Alternatively, we might want to impose order restrictions when considering interaction terms, encourage similarity among predictors within a neighborhood, or directly incorporate predictor correlations. These properties can be imposed by altering the form of the coefficient penalty, $P(\beta)$.

1.4.4 L_1 Extensions

There have been many proposed extensions and modifications to the Lasso, with the goal of improving upon the aforementioned drawbacks. Coverage of these methods is meant to be targeted rather than exhaustive, with a focus on four important extensions and their associated properties: (1) Adaptive Lasso, (2) Elastic Net, (3) Group Lasso, and (4) Fused Lasso.

1.4.4.1 Adaptive Lasso

The Adaptive Lasso was proposed to achieve model selection consistency in situations where the Lasso cannot, by introducing a data-dependent weight that controls the degree of penalty on each coefficient [77]. Given an initial estimate $\hat{\boldsymbol{\beta}}^*$, the Adaptive Lasso takes the form:

$$\hat{\boldsymbol{\beta}}_\lambda = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda \sum_{j=1}^{p} w_j \left|\beta_j\right| \right\}, \quad \lambda \geq 0, \tag{1.9}$$

where $\mathbf{w}$ is a $p \times 1$ vector of positive-valued weights defined as

$$w_j = \left|\hat{\beta}_j^*\right|^{-\gamma}.$$

For the Adaptive Lasso to achieve oracle properties (i.e., perform as well as if the true model were known in advance), it requires an initial estimate, $\hat{\boldsymbol{\beta}}^*$, that is zero-consistent. In general, estimates are considered zero-consistent if zero coefficients converge to zero while nonzero coefficients do not [32]. Thus, a typical choice for $\hat{\boldsymbol{\beta}}^*$ is the OLS estimate; however, this is not zero-consistent when $p > n$. Huang et al. [32] suggest utilizing coefficients from univariate regression, which are zero-consistent in high-dimensional situations under the assumption of partial orthogonality (i.e., important and unimportant coefficients are only weakly correlated). When partial orthogonality is not satisfied, there are some non-theoretical indications that the Lasso or Ridge estimates may suffice as initial estimators [75]. However, in situations where the Lasso does not have the variable screening property, utilizing Lasso initial estimates will not yield consistent model selection; important predictors with coefficients initially set to zero will never become nonzero, due to an infinite weight. Empirical results suggest iteratively reweighting the Adaptive Lasso can provide further improvements, decreasing bias and improving model selection consistency [11, 13].

Given an appropriate initial estimate, an additional appeal of the Adaptive Lasso is that the entire solution path can be fit using LARS. Thus, we can expect improved performance at an equivalent computational cost; of course, the Adaptive Lasso is has a greater computational expense in the high dimensional case, where an initial estimate is not trivial. For this reason, the idea of introducing adaptive, data-dependent weights is prevalent among

other Lasso extensions; for example, the Adaptive Elastic Net [80] and the Adaptive Group Lasso [69]. These adaptive extensions boast improvements with regard to selecting the true model consistently.

1.4.4.2 Elastic Net

The Elastic Net addresses problems associated with correlated variables by simultaneously utilizing the Lasso (L_1) and the Ridge (L_2). It takes the form:

$$\hat{\boldsymbol{\beta}}_{\lambda} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda_1 ||\boldsymbol{\beta}||_1 + \lambda_2 ||\boldsymbol{\beta}||_2^2 \right\} , \quad \lambda_1, \lambda_2 \geq 0, \tag{1.10}$$

where the resulting naive coefficient estimates are multiplied by $(1+\lambda_2)$ to overcome double-shrinkage [78]. Applying separate parameters to each penalty allows for a great deal of flexibility, as the Elastic Net can degenerate into a pure Ridge or Lasso regression by setting λ_1 or λ_2 equal to zero, respectively. However, this flexibility comes at the additional computational cost of optimizing over a two-dimensional grid of tuning parameters. Fortunately, a modified LARS algorithm can be used to compute the entire solution path for λ_1, given a fixed value for λ_2. By considering a grid of values for λ_2, the optimal pair $\left(\hat{\lambda}_1, \hat{\lambda}_2\right)$ can be chosen via k-fold cross-validated MSPE.

Given a set of correlated predictors, the Elastic Net will tend to include or exclude them as a group; this is in contrast to the Lasso, which will include at most only a few members of the correlated group. Another advantage of the Elastic Net is that, due to the L_2 penalty on coefficients, the maximum number of nonzero coefficients is no long limited by the number of observations. Simulation results suggest the Elastic Net has improved prediction relative to the Lasso, especially in situations with collinearity [78]. Further, under the appropriate conditions the Elastic Net has model selection consistency when $p \gg n$ [37]; the Adaptive Elastic Net achieves model selection consistency under weaker assumptions [80].

1.4.4.3 Group Lasso

The Group Lasso addresses the Lasso's inability to incorporate covariate groups, which commonly occurs when modeling categorical predictors (e.g., ethnicity, age groups). It

takes the form:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \left|\left| \mathbf{y} - \sum_{l=1}^{G} X_l \beta_l \right|\right|_2^2 + \lambda \sum_l \sqrt{p_l} \, ||\beta_l||_2 \right\} , \quad \lambda \geq 0, \tag{1.11}$$

where G represents the total number of predictor groups and p_l indicates the number of predictor members in group l; when all groups include a single member, the Group Lasso is equivalent to the Lasso [74]. The penalty imposed in the Group Lasso can be considered an L_2/L_1 penalty; an L_2 penalty is applied within each group, and an L_1 penalty is applied across groups. As a consequence, entire groups are included or excluded simultaneously. Conditions have been established for estimation and model selection consistency [51].

The original solution algorithm proposed by Yuan and Lin [74] utilized a blockwise coordinate descent procedure. However, it was based on a groupwise orthonormal $\mathbf{X}$; if X_l is not orthonormal then it can be orthonormalized, but this does not always yield a solution to the original problem [23]. Friedman et al. propose an algorithm for the nonorthogonal case, and additionally propose the Sparse Group Lasso [23]; by incorporating an additional L_1 penalty on coefficients, sparsity is also encouraged within groups. However, these methods require disjoint groups; more recent approaches consider potentially overlapping group structures [34].

1.4.4.4 Fused Lasso

The Fused Lasso [63] is designed for situations where the ordering of predictors is informative, and takes the form:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda_1 \, ||\boldsymbol{\beta}||_1 + \lambda_2 \sum_{j=2}^{p} |\beta_j - \beta_{j-1}| \right\} , \quad \lambda_1, \lambda_2 \geq 0, \tag{1.12}$$

By imposing an L_1 penalty on pairwise differences between variable neighbors, their coefficients are encouraged to be similar. The Fused Lasso has proven useful in signal approximation, which is essentially a regression problem where each observation has its own coefficient (i.e., $X = I_n$); the goal is to estimate the signal with fewer than n unique coefficients by 'fusing' neighboring coefficients. Other useful application areas include mass spectrometry

and genetics, where proximal information can improve performance.

Tibshirani et al. propose formulating the Fused Lasso as a quadratic program for fixed values of the tuning parameter [63]; however, this can be slow for large p. Fused Lasso solutions can be obtained much faster by applying a generalized coordinate-wise descent procedure [21]; unfortunately, the procedure is only guaranteed to converge for the signal approximation case (i.e., $X = I_n$). In addition, Hoefling developed a path algorithm for the signal approximation case [30]. Under a unification of the Lasso and Fused Lasso penalties (i.e., a single λ controls both), Tibshirani and Taylor provide an algorithm to compute the solution path for the situation with general $\mathbf{X}$ [65]; however, it is slow and not intended for high dimensional situations. An alternative approach for large p, employing an iterative algorithm based on the split Bregman method can significantly improve computation time for a fixed λ [73].

Rinaldo considers asymptotic properties of the Fused Lasso, and proposes an adaptive version that has improved performance for recovery of a block-sparse signal [55]. Another extension, the Generalized Lasso, considers a general fusion penalty, where a user-specified set of pairwise difference penalties is imposed [65]. Alternatively, the Pairwise Fused Lasso imposes penalties on all possible predictor pairs, regardless of proximity [54]; these penalties can incorporate weights, informed by the data or prior knowledge of predictor dependencies.

1.4.5 Lasso and Generalized Linear Models

The Lasso, and discussed extensions, are generally restricted to the typical linear regression setting, with a residual sum of squares plus penalty loss function (Equation 1.6). However, many interesting problems that involve non-Normal outcomes may benefit from Lasso-type penalties. Much work has been done to provide extensions beyond the standard linear model.

The Lasso has been extended to generalized linear models (GLMs), which take the form:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ -\log L(\mathbf{y}; \boldsymbol{\beta}) + \lambda \left|\left|\boldsymbol{\beta}\right|\right|_1 \right\} \ , \quad \lambda \geq 0, \tag{1.13}$$

where $L(\mathbf{y}; \boldsymbol{\beta})$ represents the likelihood function. Using a predictor-corrector algorithm

(def: a general algorithm that proceeds in two steps, (1) a prediction step to calculate an approximation to the desired quantity, and (2) a correction step to refine the initial prediction), the entire solution path can computed; although the paths are not exactly piecewise linear, assuming piecewise linearity between λ 'knots' is a reasonable assumption in practice [53]. A Lasso-type penalty can also be applied to Cox regression models [61], and other algorithms are available for applying the Group Lasso to GLMs [57].

The idea of L_1 regularization has been applied to principal components analysis to promote sparse loadings, which can simplify the often convoluted interpretation of components [38, 79]; it has similarly been applied to linear discriminant analysis [67]. Another common area for L_1 regularization is in graphical models derived from sparse inverse covariance matrices [47, 22], which can be useful for describing a variable dependency structure. Finally, there have been extensions to multivariate models; for example, Obozinski et al. consider L_1/L_2 regularization, where sparsity is encouraged within outcomes while grouping is encouraged across outcomes, yielding a similar set of predictors across outcomes [52].

1.5 Results

In this section, we first illustrate some of the methods we have previously introduced by using both simulated and real data. Next, we generate a set of interesting scenarios and conduct simulations to evaluate the comparative performance of these methods, plus some traditional methods, based on their ability to accurately: (1) predict, (2) estimate, and (3) select the correct model.

1.5.1 Illustrations

A Simple Simulated Dataset

We generate a single dataset according to the rules defined in Scenario 1(a) (see details of the data below, in *Simulations*), and obtain the OLS estimate in addition to the Lasso, Elastic Net, and Ridge coefficient paths (Figure 2.4). The estimates are plotted alongside each other, where the L_1-norm is chosen as 6 for purposes of comparison and to help visualize the effect of each penalty under nearly equivalent constraints (Figure 2.5).

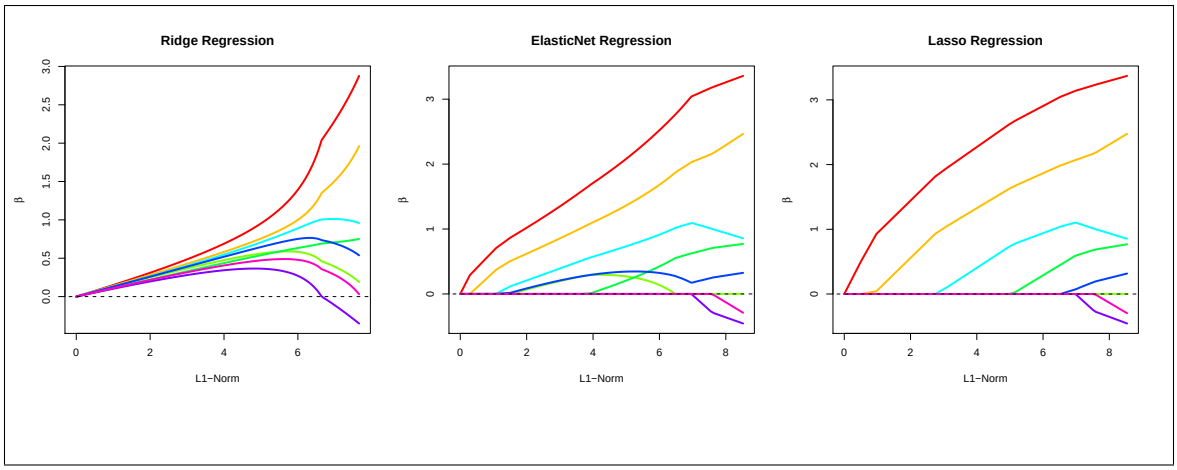

Figure 1.1: Coefficient estimates as a function of their L_1-norm for Ridge regression (left), Elastic Net (middle), and Lasso (right). Moving from left to right, coefficient paths become less smooth and more estimates are set exactly equal to zero.

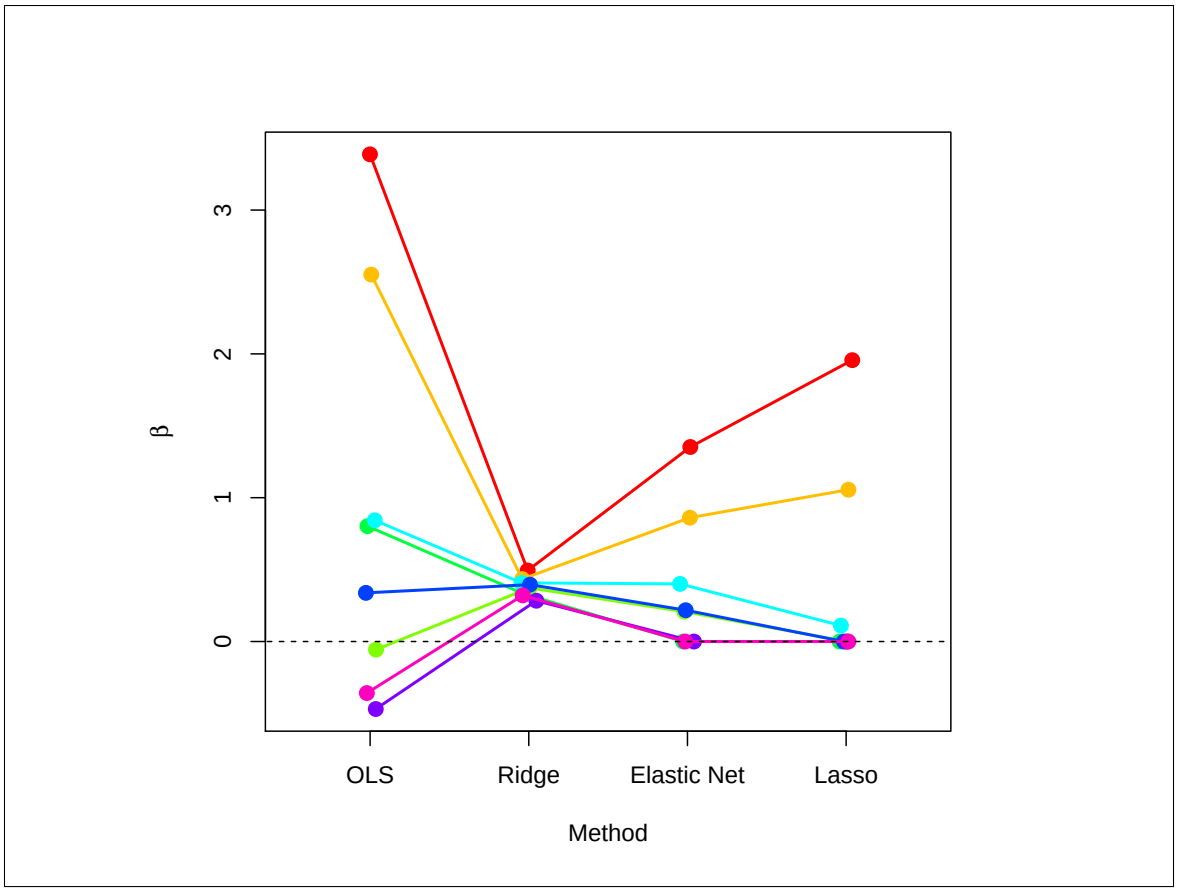

Figure 1.2: Coefficient estimates for OLS, Ridge regression, Elastic Net, and Lasso. For the latter three methods, tuning parameters were chosen such that $\sum ||\boldsymbol{\beta}||_1 \approx 6$ for each. The grouping effect of Ridge regression is evident, while the Lasso and Elastic Net force some coefficients to be exactly zero.

Real Data

To illustrate the application of the aforementioned methods, we apply these methods to following datasets: (1) Prostate, (2) Baseball, and (3) ADNI [Voxel-based morphometry]. Details regarding the data can be found in the Appendix.

The coefficient paths are displayed in Figure 2.6, where each row corresponds to a single dataset and each column corresponds to the method indicated. Notably, Ridge regression has smooth coefficient paths that appear to behave similarly, a visual confirmation of its 'grouping' property. In contrast, the Lasso has jagged paths that rarely seem to behave in concert once they diverge from zero. We found it interesting that Ridge coefficient paths

sometimes cross zero, indicating that the direction of their relationship with the outcome has changed; this highlights one benefit of the Lasso, as sign-flipping coefficients are likely negligible and should have estimates of zero.

Next, we use 5-fold cross-validation to choose the tuning parameter for each method. In general, the coefficient estimates have been shrunk relative to the OLS estimate (Figure 2.7). The Lasso sets some of the smaller coefficients equal to zero, but not all; as suggested in the literature, the Adaptive Lasso may be capable of further reducing our models by setting more of the negligible coefficients to exactly zero.

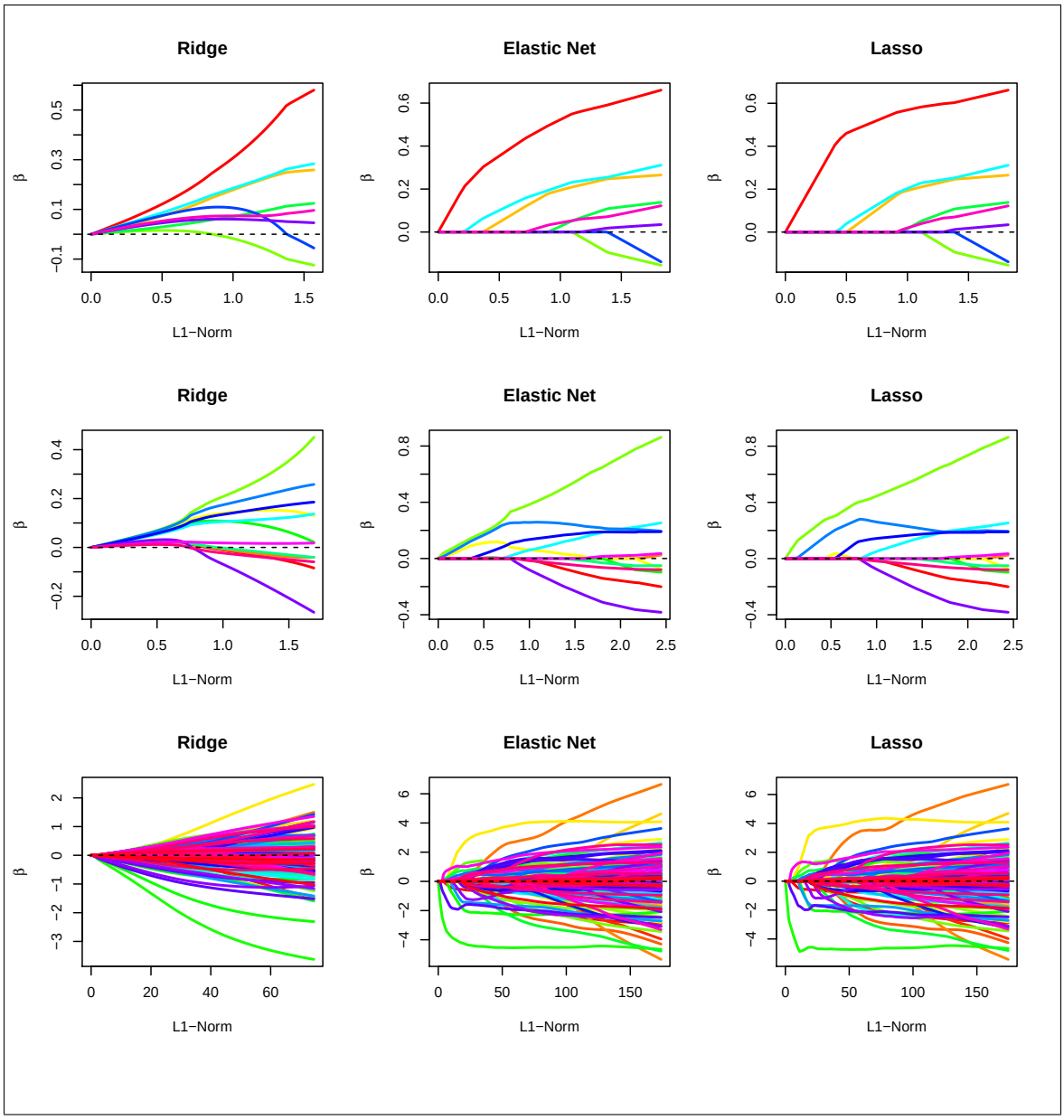

Figure 1.3: Coefficient estimates as a function of their L_1-norm for Ridge regression (left), Elastic Net (middle), and Lasso (right) applied to the Prostate (top), Baseball (middle), and ADNI [voxel-based morphometry] (bottom) datasets. Moving from left to right, coefficient paths become less smooth and more estimates are set exactly equal to zero.

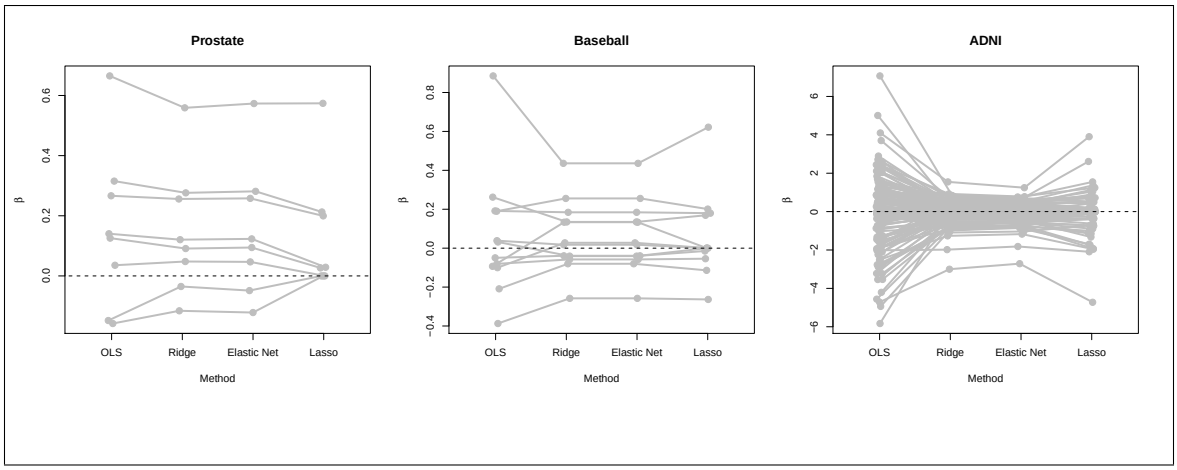

Figure 1.4: Coefficient estimates for the Prostate (left), Baseball (middle), and ADNI [voxel-based morphometry] (right). Estimates were obtained for OLS, and 5-fold cross-validation was used to obtain estimates for: Ridge regression, Elastic Net, and Lasso. All methods induce shrinkage relative to OLS; the grouping effect of Ridge regression is evident, while Lasso forces some coefficients to be exactly zero.

1.5.2 Simulations

We consider simulation scenarios to assess the comparative performance of these methods based on their ability to: (1) Predict, (2) Estimate, and (3) Select the correct model.

In each simulation, at every iteration (B total) we generate three datasets: (1) training, (2) tuning, and (3) testing. The training set is used to get estimates for each method. For methods that yield estimates across a range of tuning parameter values, we apply these estimates to the tuning set and select the one that minimizes the prediction error. At this point, we can compute estimation and model selection error. Finally, the estimates are applied to the test set to measure prediction accuracy.

Table 1.2: Simulations – Measures of Error

Type	**Formula**
Prediction Error	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
Estimation Error	$\sqrt{\frac{1}{p}\sum_{j=1}^{p}\left(\beta_j - \hat{\beta}_j\right)^2}$
Model Selection Error	$\frac{1}{p}\sum_{j=1}^{p}\left(I\left\{\beta_j \neq 0\right\} - I\left\{\hat{\beta}_j \neq 0\right\}\right)^2$

Table 1.3: Simulations – Methods for Comparison

Simulation 1	**Simulation 2**
Oracle	Oracle
Full OLS	Full OLS
Stepwise OLS (Intercept-only)	Ridge
Stepwise OLS (Full)	Adaptive Ridge ($\gamma = \{0.5, 1.0, 2.0\}$)
Ridge	Lasso
Lasso	Adaptive Lasso ($\gamma = \{0.5, 1.0, 2.0\}$)
Elastic Net	Elastic Net
	Adaptive Elastic Net ($\gamma = \{0.5, 1.0, 2.0\}$)

1.5.2.1 Description of Scenarios

Scenario 1: The first scenario is based on Example 1 from the original Lasso paper [60]. We consider $n = 20$ observations on a set of $p = 8$ predictors, where the true coefficient vector is defined as $\boldsymbol{\beta} = (3, 1.5, 0, 0, 2, 0, 0, 0)^{\mathrm{T}}$ (and $\beta_0 = 2.3$). The matrix of predictors, $\mathbf{X}$, is assumed to come from a multivariate Normal distribution with an exchangeable (i.e., compound-symmetric) correlation structure; formally, $\mathbf{X} \sim MVN(\mathbf{0}, \boldsymbol{\Sigma})$, where $\Sigma_{ij} = \rho$ for $i \neq j$ and $\rho \in [0, 1)$. We consider varying the strength of predictor correlations (ρ).

Scenario 2: The second scenario is similar to Scenario 1, except we have more observations ($n = 200$) and we consider varying the number of nuisance predictors (i.e., predictors unrelated to the outcome). An autoregressive correlation structure is used for the design

matrix, where $\Sigma_{ij} = \rho^{|i-j|}$ for $\rho \in [0, 1)$ and we set $\rho = 0.5$.

Scenario 3: The third scenario mimics Example 4 from the original Lasso paper [60]. There are $p = 40$ predictors that have a compound symmetric correlation structure, and the true coefficient vector is block-sparse with two blocks of size 10 and magnitude 2. We set $n = 100$ and $\sigma = 15$, and consider a range of values for $\rho = \{0, 0.2, 0.4, 0.6, 0.8\}$.

Scenario 4: The fourth scenario has origins in the Elastic Net paper [78]. There are three groups of important predictors, with each group containing five highly correlated ($\rho \approx 0.99$) variables. We evaluate this scenario assuming there are $n = 50$ observations and consider a range of noise levels, $\sigma = \{15, 30, 50\}$.

Scenario 5: The fifth scenario is designed to mimic a neuroimaging example. Specifically, we have collected data regarding $p = 1044$ regions of interest (ROIs), only 50 of which actually have non-zero coefficients. As is typical of neuroimaging data, the ROIs (i.e., predictors) are highly correlated ($\Sigma_{ij} = \rho = 0.7$) and there is a considerable amount of noise in the data ($\sigma = 100$). We assume, perhaps optimistically, that there are $n = 100$ observations.

Table 1.4: Simulations – Scenario Details

	Scenario 1					**Scenario 2**				**Scenario 3**					**Scenario 4**			**Scenario 5**
	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	5
B	500					500				500					500			500
$n_{train/valid/test}$	20/20/200					200/200/1000				100/100/400					50/50/400			100/100/400
p	8					10	40	70	100	40					40			1044
$\sum_j I(\beta_j \neq 0)$	3					3				20					15			50
σ	3					3				15					15	30	50	–
Σ_{ij}	ρ					$\rho^{\lvert i-j \rvert}$				ρ					-			ρ
ρ	0	0.2	0.4	0.6	0.8	0.5				0	0.2	0.4	0.6	0.8	-			0.7
$\approx R^2$	-					-				-					-			

1.5.2.2 Simulation 1

The purpose of Simulation 1 is to compare popular methods for variable selection based on their ability to predict, estimate, and select the correct model across the various scenarios. In particular, our goal is to identify methods that are universally optimal for achieving these goals.

Across all scenarios, the Oracle is expected to out perform all other methods, and therefore provides a baseline for comparison. Regarding model selection, the Full OLS estimates and Ridge regression will always perform poorly because they do not produce sparse estimates. We expect the Lasso or Elastic Net to outperform stepwise selection across the board, depending on the degree of correlation between predictors. The performance of Lasso versus Ridge will depend on predictor-correlation and coefficient sparsity. Although Elastic Net is a more flexible version of both the Lasso and Ridge, we actually expect this added flexibility to be a detriment in some circumstances – while the Lasso anticipates sparsity and the Ridge grouping, the Elastic Net must first use the data to decide on the degree of sparsity versus grouping before providing estimates. And these selections are made on the basis of prediction accuracy in the tuning set.

Scenario 1 involves a small number of predictors, a semi-sparse coefficient vector, and increasing between-predictor correlation; we anticipate the Lasso to perform best across the board when ρ is small, and Elastic Net when ρ is large. Our results indicate that the penalized approaches provide the best prediction and estimation, with Elastic Net slightly outperforming Ridge and Lasso. As expected, Lasso estimation accuracy deteriorates as the between-predictor correlation increases; however, we were surprised to see that prediction accuracy actually improved, suggesting the Lasso may be a viable option for building predictive models even when the design is correlated. The Lasso also outperformed Ridge and Elastic Net with regard to selecting the correct model. But we were surprised to see that Stepwise OLS actually performed the best, with the intercept-only starting point only slightly edging out the full model start.

Scenario 2 is similar to Scenario 1, but considers an increasing number of noise variables. Given the relatively small between-predictor correlations, we expect Lasso to outperform

all other methods. Ridge regression should perform very poorly, although perhaps not as bad as the Full OLS estimates. Simulation results suggest that we are correct with regard to prediction and estimation, as the Lasso performs favorably. We found it interesting that the Elastic Net performed equally well, even with regard to model selection; likely, this is a result of the large n in this Scenario. The most surprising result is that Stepwise OLS yields the most accurate models, even when 97% of the coefficient vector is sparse; although it suffers when starting with the Full OLS estimates, there are negligible error increases with an intercept-only starting point as the number of noise variables increase. Given the trends, it is conceivable that the Lasso may also become the model-selection champion as the coefficient vector becomes increasingly sparse.

As Scenario 3 is essentially a larger version of Scenario 1 (i.e., $p = 40$ vs. $p = 8$), we anticipate the Lasso and Elastic Net to perform best for small and large ρ, respectively. The larger p should also cause some problems for the non-penalized approaches. Our simulation results were unexpected. Most interesting, all three penalized approaches outperformed the *Oracle OLS* in terms of prediction. Further, Ridge and Elastic Net exhibit improved prediction relative to the Lasso. Similarly, the penalized approaches yield improved estimation relative to the Oracle; Ridge regression provides the most accurate estimates, regardless of the predictor-correlation. Unlike results from Scenario 1, the large p helps highlight benefits of penalized regression as the OLS-based approaches suffer from as much as a 3-fold increase in estimation error relative to penalized techniques. Finally, the Lasso does outperform all other methods when it comes to accuracy of the selected variables. Elastic Net is only marginally inferior to Ridge based on prediction and estimation, but it actually performs worse than the stepwise approaches when it comes to model selection.

Scenario 4 is complex and designed to highlight the drawbacks of the Lasso; Zou and Hastie [78] found that the Elastic Net provided the best predictions, but they did not consider estimation and model selection accuracy. Given the strong correlation between the three predictor groups, we expect the Elastic Net will outperform all other methods in terms of estimation and model selection. For the most part, our simulation results were contrary to our expectations. Similar to Scenario 3, the penalized approaches have improved prediction and estimation accuracy relative to OLS-based estimates, including the Oracle. As the noise

in the data increases, Ridge provides the best prediction and estimation. The OLS-based methods provide horrendous estimation, with errors that are as much as twenty times as large as the penalized approaches. Despite this performance, Stepwise OLS (intercept-only) is optimal when it comes to variable selection, and even Stepwise OLS (Full) is comparable to the Lasso, which outperforms Ridge and Elastic Net.

Scenario 5 is conceived as a neuroimaging example, and we suspect that the Elastic Net will perform optimally regarding our three criteria. The fact that we can actually get estimates for the penalized approaches is enough to outperform the OLS-based estimates, because $p > n$; we can obtain estimates for Oracle OLS because the number of nonzero coefficients is less than n. Our results suggest that Oracle OLS has the least accurate estimates and predictions. For the penalized approaches, performance regarding prediction and estimation is as follows (best-to-worst): (1) Ridge, (2) Elastic Net, (3) Lasso. Although seemingly outperformed, the Lasso actually improves substantially over the Elastic Net when it comes to variable selection; while the Lasso incorrectly identifies the status (zero vs. non-zero) of less than 10% of variables, the Elastic Net is incorrect about almost 70%.

Overall, the results of our simulation were surprising; our expectations, rooted in the literature, were inconsistent with what we observed. First, we expected the Oracle OLS to always outperform all other methods; intuitively, knowing the correct model should yield the best estimates and predictions. However, it was outperformed by penalized (i.e., *biased*) approaches in Scenarios 3-5. This may be because the Oracle estimates are unbiased and these scenarios have a large number of correlated predictors; even though our model is correct, the specific estimates are highly variable. It was also surprising how well Stepwise OLS performed as a method for variable selection, handily beating the Lasso in 3 out of 5 Scenarios. However, it would be difficult to recommend Stepwise OLS if there is any interest in prediction or interpreting the resulting coefficient estimates.

As we might have expected, there is no panacea; while some methods performed consistently well, there is no scenario in which a single method proved optimal for prediction, estimation, and model selection. As suggested in the literature, the Lasso appears to have trouble trimming down a model and often includes extraneous noise variables; explored in our second simulation, adaptive weighting may cause improvements. Similarly, the Elastic

Net tends to select models that are closer in size to Ridge than Lasso, which makes it difficult to recommend as a procedure for variable selection. When it comes to prediction, differences between the three methods were negligible. This was true even as the magnitude of predictor-correlations increased. However, we found that increasing ρ had a negative impact on the estimation and variable selection accuracy of the Lasso.

Table 1.5: Results for Simulation 1 (based on $B = 500$ replicates)

	Scenario 1					Scenario 2				Scenario 3					Scenario 4			Scen.
(Prediction Error)	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	5
Oracle	3.3	3.3	3.3	3.3	3.3	3.0	3.0	3.0	3.0	16.9	16.8	16.8	16.9	16.8	18.1	36.5	60.6	45.2
Full OLS	3.9	4.0	4.0	3.9	4.0	3.1	3.4	3.7	4.3	19.5	19.5	19.3	19.5	19.4	35.2	70.0	118.4	
Stepwise OLS (0)	3.9	3.9	3.8	3.8	3.7	3.2	3.2	3.3	3.4	19.0	19.1	18.6	18.2	17.4	37.4	83.7	105.9	
Stepwise OLS (Full)	3.8	3.9	3.9	3.8	3.8	3.0	3.1	3.3	3.5	18.8	19.1	18.7	18.3	17.7	31.3	62.0	106.4	
Ridge	3.6	3.6	3.6	3.5	3.4	3.1	3.3	3.4	3.6	16.6	16.2	16.0	15.7	15.5	17.3	32.7	53.1	32.1
Lasso	3.6	3.6	3.5	3.5	3.4	3.1	3.1	3.1	3.1	17.0	16.9	16.6	16.3	16.0	16.9	33.5	54.7	33.0
ElasticNet	3.6	3.5	3.5	3.4	3.4	3.1	3.1	3.1	3.1	16.6	16.3	16.0	15.7	15.5	16.8	32.8	53.3	32.3
(Estimation Error)																		
Oracle	0.47	0.49	0.54	0.59	0.78	0.14	0.07	0.05	0.04	1.21	1.29	1.50	1.86	2.54	13.82	28.30	46.94	1.97
Full OLS	0.90	0.97	1.08	1.31	1.84	0.28	0.30	0.34	0.39	1.96	2.15	2.44	3.08	4.27	27.49	54.80	93.85	
Stepwise OLS (0)	0.86	0.87	0.95	1.14	1.49	0.34	0.20	0.17	0.16	1.82	2.04	2.21	2.52	3.01	6.67	13.21	16.02	
Stepwise OLS (Full)	0.81	0.86	0.98	1.20	1.61	0.18	0.16	0.17	0.20	1.79	2.03	2.22	2.60	3.23	18.84	37.21	65.38	
Ridge	0.72	0.73	0.77	0.83	0.95	0.27	0.24	0.23	0.22	1.11	0.97	0.98	1.02	1.09	1.07	1.56	2.04	0.25
Lasso	0.66	0.67	0.72	0.81	1.04	0.20	0.11	0.09	0.08	1.25	1.27	1.38	1.54	1.83	2.81	3.02	3.17	0.51
ElasticNet	0.66	0.66	0.70	0.78	0.94	0.20	0.11	0.09	0.08	1.12	0.99	1.01	1.05	1.14	1.97	1.91	2.30	0.31
(Model Selection Error)																		
Oracle	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Full OLS	0.62	0.62	0.62	0.62	0.62	0.70	0.92	0.96	0.97	0.50	0.50	0.50	0.50	0.50	0.62	0.62	0.62	
Stepwise OLS (0)	0.17	0.17	0.18	0.22	0.28	0.04	0.05	0.05	0.05	0.42	0.38	0.40	0.43	0.46	0.34	0.35	0.37	
Stepwise OLS (Full)	0.19	0.19	0.21	0.25	0.31	0.05	0.07	0.09	0.12	0.40	0.38	0.40	0.43	0.46	0.46	0.46	0.46	
Ridge	0.62	0.62	0.62	0.62	0.62	0.70	0.92	0.96	0.97	0.50	0.50	0.50	0.50	0.50	0.62	0.62	0.62	0.95
Lasso	0.37	0.36	0.34	0.36	0.37	0.33	0.19	0.13	0.10	0.38	0.33	0.34	0.37	0.40	0.44	0.46	0.45	0.07
ElasticNet	0.43	0.44	0.43	0.45	0.49	0.35	0.20	0.13	0.11	0.46	0.47	0.47	0.48	0.48	0.38	0.52	0.56	0.67

1.5.2.3 Simulation 2

The literature suggests that incorporating adaptive weighting can help to reduce bias and remove additional noise variables. In particular, the weights yield stronger penalties on coefficients with smaller initial estimates, pushing them to exactly zero while relaxing weights on coefficients with a larger magnitude of effect. Although we anticipate improvements in model selection, it is not clear what impact this will have on prediction and estimation. Further, there is some uncertainty in the literature regarding what power (γ) to use in the weights; as γ increases, so does the influence of the initial estimates on the resulting coefficients. We expect that adaptive weighting will prove most useful in situations where the OLS estimate has favorable performance, because the OLS estimate is used to compute initial weights; if the initial estimate is poor, so too should be the results based on adaptive weighting. The purpose of Simulation 2 is to evaluate whether adaptive weighting is effective across the trifecta of fit criteria (i.e., prediction, estimation, and model selection), and determine if there is an optimal value of γ to use for computing these weights.

In Scenario 1, we previously noted that as predictor-correlations increase, prediction accuracy increases while estimation accuracy decreases. We expected the inclusion of adaptive weighting would lead to improved performances, but weights actually lead to mild inflation of the estimation and prediction errors; as γ increases, so do the errors. However, we see the benefits of adaptive weighting when focusing on model selection, as increasing γ leads to improvements in model selection accuracy. These improvements are most substantial for small values of ρ, but nearly non-existent as the correlation between predictors grows large.

Scenario 2 is designed to showcase the Lasso, and we expect adaptive weighting to further improve the accuracy of model selection. Predictor-correlations are less substantial in this scenario, so we expect the OLS estimates to perform well and therefore anticipate adaptive weighting to prove useful. Because we trust the initial estimates, we should see improvements in prediction and estimation as γ (i.e., the influence of the weights) grows large. The simulation results tended to agree with our expectations, as we found that increasing γ led to improvements in prediction accuracy. These improvements were negligible for a small number of noise variables, but were substantial as the number of noise variables

grew. The largest improvements were seen for Ridge, where adaptive weighting yielded prediction accuracies on par with Lasso and Elastic Net. We previously noticed that Ridge performed poorly in this scenario, but found here that adaptive weighting provides substantial improvements. While Lasso and Elastic Net benefit from adaptive weighting, they approach the prediction accuracy of the Oracle without it; in other words, they perform so well without adaptive weighting that we cannot hope for substantial improvements in prediction beyond this. Estimation results mimic those of prediction; adaptive weighting is most beneficial for Ridge when there are plenty of noise variables, although weighting for all methods with any γ leads to improvements. Once again, model selection is where Lasso and Elastic Net benefit the most from adaptive weighting. For example, we see Adaptive Lasso ($\gamma = 2.0$) yield as much as a seven-fold improvement in accuracy, relative to Lasso; the Elastic Net sees similar benefits, but to a lesser degree. The results of this simulation make it quite clear that adaptive weighting does a good job of pushing coefficients estimates associated with those negligible 'noise' variables to exactly zero.

Similar to Scenario 1 but with a larger number of predictors, we expect the results of Scenario 3 to be similar. As ρ increases, the OLS estimates should become less accurate and this error will propagate into the weights; adaptive weighting should prove less useful in this scenario. As expected, our simulation results indicate that adaptive weighting leads to predictions with increased error, and these errors grow with γ. We found it interesting that this was true even when $\rho = 0$, although we would have expected to see a benefit to adaptive weighting in the situation; when OLS estimates are accurate and trustworthy, we generally expect to see improvements when considering an adaptive weighting scheme. Estimation results behave similarly, where we see only drawbacks associated with adaptive weighting. Unlike with prediction, adaptive weighting incurs substantial estimation errors that grow with γ and ρ. Even more surprising, the negatives of adaptive weighting continue into model selection, as errors here also tend to grow with γ. While adaptive weighting can usually trim excess noise variables from our model, it actually leads to accuracy reductions in the scenario. This is even true when $\rho = 0$. Overall, these results are concerning; our previous scenarios paint adaptive weighting as an approach that, while variable in its influence on prediction and estimation, leads to consistent improvements to model selection. However,

here we have provided evidence that blindly incorporating weighting, even when relying on seemingly trustworthy initial estimates, can have a potentially detrimental impact on analysis results.

Scenario 4 is designed to be difficult for methods that do not include an L_2-norm (i.e., 'grouping' effect). We expect the OLS estimates to be poor, and this should make it difficult for the adaptive versions to perform well. Our simulation results suggest that weighting provides minor increases to prediction accuracy as γ increases; although these improvements are small, we would have expected the weights to incur additional errors in prediction. Estimation results for Ridge were particularly interesting, as we found that $\gamma = 0.5$ lead to improvements over standard Ridge, but further increasing γ had a negative impact. Small values of γ imply less influence on the initial estimate, so we found that using only a little information from a unstable estimate ultimately led to minor improvements in estimation accuracy; however, putting too much weight on an unstable initial estimate is unwise and can reduce the accuracy of the resulting coefficients. Lasso and Elastic Net did not behave like Ridge, and saw increased estimation error when incorporating any adaptive weights. However, using adaptive weights proved beneficial for model selection, as accuracy increased with γ for both Elastic Net and Lasso. Although this situation is designed to be difficult for Lasso, we found the Adaptive Lasso ($\gamma = 2.0$) to be the top performer for model selection in all three versions of Scenario 4.

Ultimately, we decided to ignore Scenario 5 for the second set of simulations. The primary reason for this is the lack of an initial estimate, as we cannot obtain a unique OLS estimate in this scenario; without an initial estimate, it is not clear how to proceed with adaptive weighting. One possibility is to use, for example, the Lasso to obtain an initial estimate for the adaptive version. However, this could be considered double-dipping into the data. Thus, without a clear path forward we decided to forgo Simulation 2 on Scenario 5.

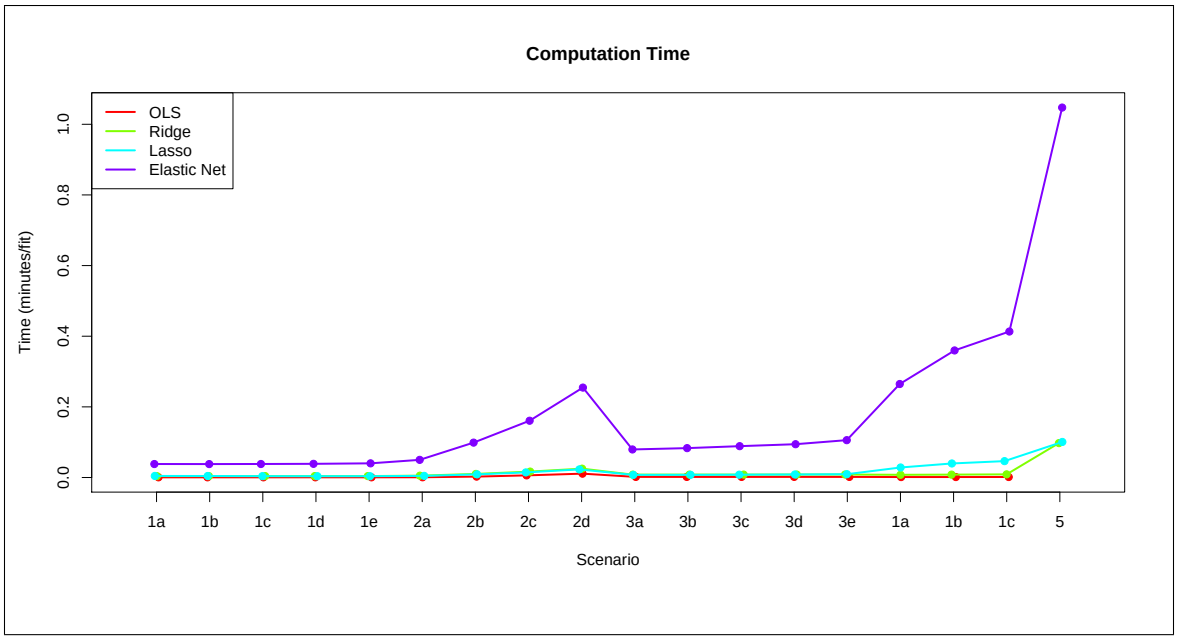

Figure 1.5: Computation times across all 18 scenarios for Simulation 2. The adaptive versions are not shown, but have comparable times to their unweighted counterparts. Notably, the Elastic Net demands far greater computation time than all other methods combined.

1.6 Discussion

We have conducted a detailed review of popular methods for variable selection when the number of important predictors is unknown, including both traditional and modern techniques. While traditional approaches are capable for small problems with simple predictor dependency structures, techniques like penalized regression have been developed primarily to handle complex high-dimensional problems. Not only are these methods capable of managing a large number of predictors, but they can also leverage correlation in the design. And with a suitable initial estimate, further gains can be made by utilizing adaptive weighting.

However, the literature is scarce when it comes to drawbacks and recommendations. We know that Ridge regression is suited for situations with correlated predictors, and the Lasso for sparsity, but which method should we use if we expect a little of both? And while the Elastic Net may be optimal for this situation, are there any drawbacks? Computation time will be increased because we must consider an additional tuning parameter, so is there a correlation or sparsity threshold below which we can do just as well by simply applying the Lasso or Ridge, respectively? Or, because the Elastic Net contains both the Lasso and Ridge as special cases, is it optimal in every situation? And when we say *optimal*, are were

referring to prediction, estimation, or model selection?

The results of our simulations suggest that there is not one single method that always outperforms all others. Just like with traditional approaches to model selection, the statistician must think carefully about the specific application, and the strengths and weaknesses of each potential approach, before making a decision about how to proceed. When it comes to prediction and estimation, modern techniques consistently outperform traditional approaches. However, modern techniques appear underwhelming when it comes to model selection. A common target for criticism, stepwise selection was the optimal performer in many scenarios and performed comparably with Lasso in others. Based on our simulations, stepwise selection may not be all that bad as a tool for automated model selection; however, we recommend against it if there is interest in interpreting the coefficient estimates or predicting future observations.

As illustrated in the second simulation, adaptive weighting led to further reductions in model selection error for Lasso and Elastic Net. But in general, Elastic Net tends to choose models that are much larger, and misclassify more variables, than Lasso. Even when considering adaptive versions of the Elastic Net, it cannot hope to achieve Lasso-levels of model selection accuracy. In most situations, the Lasso appears preferable to the Elastic Net for purposes of model selection; for estimation and prediction, we would recommend Elastic Net over Lasso.

One of the most surprising results of our simulations, knowing the true model and obtaining the corresponding OLS fit (i.e., Oracle OLS) did not always yield the most accurate estimates and predictions. Modern approaches seem to outperform the Oracle OLS when there are many important predictors that are reasonably correlated. This is quite a shocking revelation, as it would make intuitive sense that knowing the true model should make it much easier to provide accurate estimates and predictions; in some applications, there appears to be little hope for building an accurate model using least squares. Thus, for complex models it may be wise to allow for some bias in the coefficient estimates.

Our simulation results regarding the impact of adaptive weighting, using OLS as an initial estimate, weave a tale of caution. Across the simulation scenarios, and in agreement with the literature, adaptive weighting seemed to have the most positive impact on model

selection accuracy. Confusingly, it had a negative impact in Scenario 3, leading to more misidentification of significant predictors. Given that Scenarios 1 and 3 are quite similar, it is not clear why adaptive weighting has an overall negative impact on Scenario 3 while it reduces model selection error in Scenario 1.

It is also difficult to make a universal recommendation regarding the value of γ when considering adaptive weighting. In general, larger values of γ imply a stronger influence of the initial estimate on the final coefficients. Therefore, when a good and reliable initial estimate is available, we should consider using larger values of γ; in other words, we recommend that γ reflect the quality of the initial estimate. However, if prediction or estimation are primarily of interest, as opposed to variable selection, then it may be best to forgo weighting (i..e, $\gamma = 0$); reductions in model selection error do not imply the adaptive weighting scheme will yield concurrent improvements to prediction and estimation accuracy. We are not claiming that adaptive weighting will not lead to better predictions; however, we are suggesting that it may result in poorer predictions.

We have previously discuss a number of alternatives that have been proposed in the literature. Not included in our simulations, fusion penalties are another alternative for promoting 'grouping' among coefficient estimates. In Chapter 2, we explore the application of fusion penalties in more detail.

Table 1.6: Results for Simulation 2 (Scenarios 1-2)

	Scenario 1 ($B = 5000$)					Scenario 2 ($B = 5000$)			
(Prediction Error)	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)
Oracle	3.31	3.32	3.31	3.31	3.32	3.03	3.03	3.03	3.03
Full OLS	3.94	3.96	3.96	3.95	3.97	3.09	3.36	3.73	4.25
Ridge	3.64	3.61	3.55	3.48	3.37	3.08	3.27	3.44	3.59
aRidge ($\gamma = 0.5$)	3.61	3.58	3.53	3.47	3.38	3.07	3.19	3.29	3.38
aRidge ($\gamma = 1.0$)	3.59	3.58	3.54	3.48	3.39	3.06	3.13	3.19	3.25
aRidge ($\gamma = 2.0$)	3.59	3.60	3.58	3.53	3.45	3.07	3.08	3.11	3.14
Lasso	3.57	3.55	3.52	3.48	3.43	3.05	3.08	3.09	3.10
aLasso ($\gamma = 0.5$)	3.58	3.58	3.57	3.54	3.50	3.05	3.06	3.06	3.07
aLasso ($\gamma = 1.0$)	3.59	3.60	3.60	3.58	3.53	3.04	3.05	3.05	3.06
aLasso ($\gamma = 2.0$)	3.60	3.63	3.63	3.61	3.57	3.04	3.04	3.05	3.06
ElasticNet	3.56	3.53	3.49	3.45	3.37	3.06	3.08	3.09	3.10
aElasticNet ($\gamma = 0.5$)	3.55	3.54	3.51	3.47	3.40	3.05	3.06	3.06	3.07
aElasticNet ($\gamma = 1.0$)	3.56	3.56	3.54	3.50	3.43	3.04	3.05	3.06	3.06
aElasticNet ($\gamma = 2.0$)	3.57	3.59	3.58	3.55	3.48	3.04	3.04	3.05	3.06
(Estimation Error)									
Oracle	0.48	0.49	0.53	0.60	0.81	0.14	0.07	0.05	0.04
Full OLS	0.88	0.95	1.08	1.31	1.85	0.28	0.31	0.34	0.39
Ridge	0.72	0.72	0.77	0.84	0.96	0.26	0.24	0.23	0.22
aRidge ($\gamma = 0.5$)	0.70	0.70	0.74	0.81	0.95	0.24	0.19	0.18	0.17
aRidge ($\gamma = 1.0$)	0.69	0.69	0.73	0.81	0.97	0.21	0.15	0.14	0.13
aRidge ($\gamma = 2.0$)	0.68	0.70	0.75	0.85	1.04	0.21	0.12	0.10	0.10
Lasso	0.67	0.67	0.72	0.81	1.04	0.19	0.11	0.09	0.08
aLasso ($\gamma = 0.5$)	0.67	0.69	0.75	0.87	1.14	0.17	0.09	0.07	0.07
aLasso ($\gamma = 1.0$)	0.67	0.70	0.77	0.91	1.20	0.17	0.09	0.07	0.06
aLasso ($\gamma = 2.0$)	0.68	0.72	0.80	0.94	1.25	0.15	0.08	0.07	0.06
ElasticNet	0.66	0.66	0.70	0.78	0.94	0.20	0.11	0.09	0.08
aElasticNet ($\gamma = 0.5$)	0.66	0.66	0.71	0.80	0.98	0.17	0.09	0.07	0.07
aElasticNet ($\gamma = 1.0$)	0.66	0.67	0.73	0.82	1.02	0.17	0.09	0.07	0.06
aElasticNet ($\gamma = 2.0$)	0.66	0.69	0.76	0.88	1.11	0.16	0.08	0.07	0.06
(Model Selection Error)									
Oracle	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Full OLS	0.62	0.62	0.62	0.63	0.62	0.70	0.92	0.96	0.97
Ridge	0.62	0.62	0.62	0.62	0.62	0.70	0.92	0.96	0.97
aRidge ($\gamma = 0.5$)	0.62	0.62	0.62	0.62	0.62	0.70	0.92	0.96	0.97
aRidge ($\gamma = 1.0$)	0.62	0.62	0.62	0.63	0.62	0.70	0.92	0.96	0.97
aRidge ($\gamma = 2.0$)	0.62	0.62	0.62	0.62	0.62	0.68	0.90	0.93	0.94
Lasso	0.37	0.35	0.35	0.35	0.37	0.35	0.18	0.13	0.10
aLasso ($\gamma = 0.5$)	0.30	0.30	0.31	0.32	0.35	0.20	0.09	0.06	0.05
aLasso ($\gamma = 1.0$)	0.27	0.27	0.29	0.31	0.35	0.14	0.06	0.04	0.04
aLasso ($\gamma = 2.0$)	0.24	0.24	0.27	0.29	0.33	0.05	0.04	0.03	0.03
ElasticNet	0.44	0.42	0.44	0.45	0.49	0.38	0.19	0.13	0.11
aElasticNet ($\gamma = 0.5$)	0.38	0.38	0.41	0.44	0.48	0.23	0.10	0.06	0.05
aElasticNet ($\gamma = 1.0$)	0.36	0.37	0.40	0.44	0.48	0.19	0.08	0.05	0.04
aElasticNet ($\gamma = 2.0$)	0.33	0.34	0.38	0.41	0.47	0.12	0.08	0.06	0.05

Table 1.7: Results for Simulation 2 (Scenarios 3-5)

	Scenario 3 ($B = 5000$)					Scenario 4 ($B = 5000$)			Scenario 5 ($B = 1000$)
(Prediction Error)	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(a)
Oracle	16.8	16.8	16.8	16.8	16.8	18.1	36.1	60.2	45.1
Full OLS	19.4	19.4	19.4	19.5	19.4	35.3	70.4	117.9	
Ridge	16.6	16.2	16.0	15.7	15.5	17.2	32.7	53.1	42.4
aRidge ($\gamma = 0.5$)	16.6	16.4	16.1	15.8	15.5	16.4	31.9	52.4	42.3
aRidge ($\gamma = 1.0$)	16.7	16.7	16.4	16.0	15.7	16.0	31.6	52.2	42.3
aRidge ($\gamma = 2.0$)	16.9	17.2	16.9	16.5	15.9	15.9	31.3	52.0	42.3
Lasso	17.0	16.9	16.6	16.4	16.0	16.8	33.4	54.7	42.3
aLasso ($\gamma = 0.5$)	17.2	17.4	17.2	16.8	16.3	16.1	31.9	52.9	42.3
aLasso ($\gamma = 1.0$)	17.3	17.7	17.5	17.1	16.5	15.9	31.6	52.5	42.3
aLasso ($\gamma = 2.0$)	17.4	18.1	17.9	17.5	16.8	15.9	31.6	52.5	42.4
ElasticNet	16.6	16.2	16.0	15.7	15.5	16.8	32.8	53.2	42.4
aElasticNet ($\gamma = 0.5$)	16.7	16.4	16.1	15.9	15.6	16.1	31.8	52.5	42.4
aElasticNet ($\gamma = 1.0$)	16.8	16.7	16.4	16.0	15.7	15.9	31.6	52.3	42.4
aElasticNet ($\gamma = 2.0$)	16.9	17.2	16.9	16.5	15.9	15.8	31.5	52.2	42.4
(Estimation Error)									
Oracle	1.20	1.30	1.50	1.83	2.56	13.85	27.69	46.00	1.96
Full OLS	1.94	2.15	2.48	3.05	4.28	27.87	55.52	93.20	
Ridge	1.12	0.97	0.98	1.02	1.08	1.05	1.52	2.01	1.71
aRidge ($\gamma = 0.5$)	1.14	1.05	1.06	1.10	1.17	1.02	1.37	1.83	1.71
aRidge ($\gamma = 1.0$)	1.17	1.15	1.19	1.24	1.32	1.30	1.57	2.00	1.71
aRidge ($\gamma = 2.0$)	1.23	1.34	1.44	1.54	1.65	1.84	2.17	2.75	1.71
Lasso	1.27	1.28	1.39	1.55	1.83	2.81	2.99	3.12	1.71
aLasso ($\gamma = 0.5$)	1.32	1.47	1.62	1.83	2.16	3.24	3.33	3.47	1.71
aLasso ($\gamma = 1.0$)	1.35	1.59	1.76	1.98	2.34	3.43	3.77	4.26	1.71
aLasso ($\gamma = 2.0$)	1.38	1.72	1.92	2.17	2.58	3.91	5.03	6.33	1.71
ElasticNet	1.13	0.99	1.01	1.05	1.13	1.94	1.91	2.23	1.72
aElasticNet ($\gamma = 0.5$)	1.15	1.05	1.07	1.11	1.20	2.41	2.46	2.61	1.72
aElasticNet ($\gamma = 1.0$)	1.18	1.15	1.19	1.25	1.33	2.60	2.89	3.31	1.72
aElasticNet ($\gamma = 2.0$)	1.24	1.35	1.44	1.54	1.66	3.20	4.15	5.38	1.72
(Model Selection Error)									
Oracle	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Full OLS	0.50	0.50	0.50	0.50	0.50	0.62	0.62	0.62	
Ridge	0.50	0.50	0.50	0.50	0.50	0.62	0.62	0.62	0.89
aRidge ($\gamma = 0.5$)	0.50	0.50	0.50	0.50	0.50	0.62	0.62	0.62	0.90
aRidge ($\gamma = 1.0$)	0.50	0.50	0.50	0.50	0.50	0.62	0.62	0.62	0.90
aRidge ($\gamma = 2.0$)	0.50	0.50	0.50	0.50	0.50	0.62	0.62	0.61	0.90
Lasso	0.39	0.33	0.34	0.37	0.40	0.43	0.45	0.44	0.89
aLasso ($\gamma = 0.5$)	0.39	0.36	0.37	0.39	0.43	0.34	0.35	0.36	0.90
aLasso ($\gamma = 1.0$)	0.40	0.37	0.38	0.41	0.44	0.31	0.32	0.33	0.90
aLasso ($\gamma = 2.0$)	0.42	0.38	0.40	0.42	0.45	0.28	0.28	0.29	0.90
ElasticNet	0.47	0.47	0.47	0.48	0.48	0.38	0.51	0.56	0.86
aElasticNet ($\gamma = 0.5$)	0.48	0.49	0.49	0.49	0.49	0.33	0.41	0.46	0.87
aElasticNet ($\gamma = 1.0$)	0.48	0.50	0.50	0.50	0.50	0.33	0.39	0.43	0.87
aElasticNet ($\gamma = 2.0$)	0.48	0.50	0.50	0.50	0.50	0.34	0.38	0.41	0.87

Chapter 2

A Flexible Dantzig Selector

2.1 Introduction

The previous chapter introduced penalties in the linear regression setting, with a primary goal of inducing coefficient sparsity to perform simultaneous estimation and variable selection. The reasoning behind this approach is simple: if we expect only a subset of the coefficient to be nonzero, then we should encourage this behavior. Essentially, we are imposing a penalty that reflects a prior expectation of a sparse coefficient vector. However, quite often we have additional information beyond a simple expectation of sparsity. For example, we may expect predictors that are highly correlated with each other to behave similarly with regard to a particular response variable, and predictors may also have spatial or temporal dependency structures.

In this chapter we consider more complex fusion penalties as a means of imposing known structural relationships between predictors on the resulting coefficient estimates. Further, we consider the Dantzig Selector as an alternative approach to achieving sparsity, and propose an extension to incorporate general coefficient penalties, including fusion, under this framework. In alignment with the literature, we also propose an adaptive version that weights each penalty proportional to an initial estimate. The utility of our method is illustrated using simple examples and a detailed simulation, and we conduct an analysis on neuroimaging data obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI), where the goal is to identify areas of the brain that are volumetrically associated with

a quantitative measure of disease severity derived from patient scores on the Alzheimer's Disease Assessment Scale (ADAS).

2.2 Background

2.2.1 Generalized Lasso

2.2.1.1 Definition

Tibshirani and Taylor [65] propose a general formulation of the Lasso that can capture a variety of existing Lasso-type problems. Specifically, they consider problems of the form:

$$\hat{\boldsymbol{\beta}}_\lambda = \underset{\beta}{\operatorname{argmin}} \left\{ ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda \, ||\mathbf{M}\boldsymbol{\beta}||_1 \right\} \ , \quad \lambda \geq 0, \tag{2.1}$$

where $\mathbf{M} \in \mathbb{R}^{m \times p}$ is a fixed penalty matrix. For example, if $\mathbf{M} = \mathbf{I}_p$ where I_p represents a p-dimensional square identity matrix, then Equation 2.1 reduces to a standard Lasso problem. In general, penalty matrix $\mathbf{M}$ allows for L_1 penalties to be applied to any linear combination of $\boldsymbol{\beta}$, where some familiar special cases include: Fused Lasso [63], polynomial trend filtering, and outlier detection [65]. Further, if $\text{rank}(\mathbf{M}) = m \leq p$ then Equation 2.1 can be expressed as a standard Lasso problem, and the entire solution path can be obtained using the efficient LARS algorithm [17]. This is not the case when $\text{rank}(\mathbf{M}) < m$, which occurs when there are more constraints than predictors; a path algorithm is developed for this case, the derivation of which is based on solving the dual problem [65].

2.2.1.2 Estimation

The proposed optimization algorithm for estimating the coefficients relies on solving the dual problem, which enjoys constraints that are more well-behaved than those of the primal problem. Although the dual problem is not strictly convex when $\text{rank}(\mathbf{M}) < m$, the primal problem is strictly convex regardless of $\mathbf{M}$; thus, strong duality can be employed [65]. Specifically, the entire dual solution path can be computed and subsequently transformed to achieve the primal solution path. Unfortunately, the path algorithm is not efficient in high-dimensional settings ($p \gg n$) or when there are a large number of linear constraints on the

coefficients; the path algorithm requires m steps at a minimum (i.e., one step per constraint), but usually more. These shortcomings can be overcome by stopping the algorithm after a pre-specified number of steps; the path concludes at the unregularized OLS estimate, while interest usually lies in the more regularized solutions that are computed early on in the coefficient path. Alternatively, an approximate version of the path algorithm is guaranteed to converge after m steps; however, performance of the approximate algorithm is largely untested, and can still be unwieldy if m is large [65].

The `genlasso` package for R can be used to fit the Generalized Lasso and is currently available for download from CRAN [4].

2.2.1.3 Incorporating a Ridge Penalty (L_2)

The benefits of including an additional penalty on the L_2-norm of the coefficients $\left(||\boldsymbol{\beta}||_2^2\right)$ are akin to those enjoyed by Ridge regression [31]; performance is improved in collinear situations, and the computational issues associated with a less-than-full-rank design matrix are alleviated. Further, this 'Generalized Elastic Net' can be formulated as a Generalized Lasso problem, and the solution path can be computed along a fixed grid of values for $\tau > 0$ and $\lambda > 0$:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \left|\left| \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{X} \\ \tau \mathbf{I}_p \end{bmatrix} \boldsymbol{\beta} \right|\right|_2^2 + \lambda \left|\left|\mathbf{M}\boldsymbol{\beta}\right|\right|_1 \right\}. \tag{2.2}$$

2.2.1.4 Penalty Matrix, M

The Generalized Lasso provides a flexible framework for introducing constraints, which is useful for tackling a number of unique applications. For example, we can impose a one-dimensional fusion penalty using:

$$\mathbf{M}_{1d} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

For any penalty matrix, we can impose pure sparsity on the solution by appending the identity matrix; for example, we can consider the one-dimensional Fused Lasso:

$$\mathbf{M}_{s1d} = \begin{bmatrix} \mathbf{M}_{1d} \\ \mathbf{I}_n \end{bmatrix}$$

This flexibility allows for specification of unique penalties that cannot be easily applied using existing methods. Note that $\text{rank}(\mathbf{M}_{1d}) = m$ while $\text{rank}(\mathbf{M}_{s1d}) < m$; only the former penalty matrix can be formulated as a Lasso problem and solved directly via LARS [65].

A further appeal, $\mathbf{M}$ can be constructed from any undirected graph $\mathbf{G}$ (Figure 2.1). Specifically, suppose there are p nodes, representing potential predictors, that may be connected by m_e edges ($m = m_e + p$) representing known variable dependencies. The penalty matrix is constructed such that variable pairs with known dependencies (i.e., edges) incur a penalty based on the absolute difference in their regression coefficients, while each node also incurs its own magnitude-based penalty. In this way, node-based penalties encourage overall sparsity while edge-based penalties promote similar coefficients among connected nodes.

Note that although we can construct $\mathbf{M}$ for any undirected graph, the reverse is not true; there are a multitude of $\mathbf{M}$ that do not have a corresponding undirected graph representation. This speaks to the flexibility of the penalty matrix, as it can easily adapt to a variety of applications.

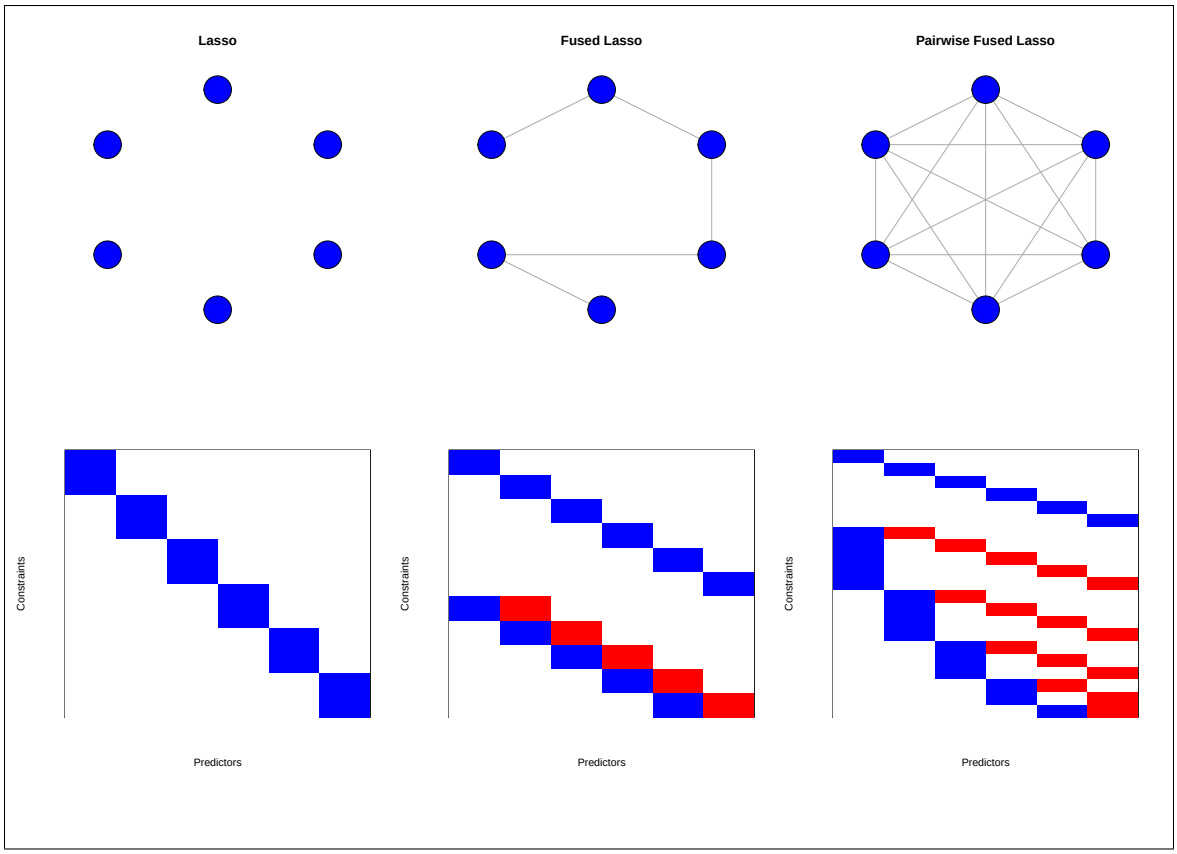

Figure 2.1: Graphical networks reflecting predictor structure (top row) and the corresponding penalty matrices (bottom row) for the Lasso [60], Fused Lasso [63], and Pairwise Fused Lasso [54]. As the number of predictor-dependencies increase, so does the number of rows in $\mathbf{M}$.

2.2.1.5 Shortcomings

Although flexible, the Generalized Lasso cannot accommodate certain interesting penalties; for example, it cannot accommodate the hybrid L_1/L_2-norm penalty of the Group Lasso, which encourages sparsity across groups and grouping within [74]. An additional concern is that the incorporation of a Ridge-type penalty (L_2-norm) is tailored specifically for the linear regression setting, and may prove more difficult to include when considering non-Normal outcomes or alternative loss functions. Further, the equal influence of all constraints may allow some constraints to dominate, especially when considering penalties of different types and magnitudes. This is also true of the number of constraints, which is directly proportional to how much influence each constraint is given (relative to λ). Ultimately, these problems

stem from having only a single tuning parameter and no scheme for adaptive weighting, although they can be overcome on a case-by-case basis with intelligent specification of the penalty matrix.

2.2.1.6 Discussion

The Generalized Lasso presents a unique framework in which to incorporate known variable dependencies in a linear regression model. The close tie between undirected graphs and penalty matrix $\mathbf{M}$ is especially appealing for applications with spatial and/or temporal predictor dependencies (e.g., neuroimaging, genetics). For example, graphical networks are a useful tool for representing and visualizing brain connectivity. One potentially interesting application of the Generalized Lasso would involve incorporating knowledge of regional connectivity in the resulting model estimates, such as structural connectivity information derived from diffusion tensor imaging (DTI).

However, there are still numerous problems regarding the Generalized Lasso that need to be addressed. First and foremost, computational hurdles remain. Proposed algorithms are only effective under a variety of limiting conditions, including: $n > p$, a relatively small number of predictors (p), and a reasonable number of constraints (m). Thus, current algorithms are not feasible in high-dimensional settings, and an optimization algorithm currently only exists for Normally-distributed outcomes. Extending the Generalized Lasso to generalized linear models (GLMs), in both theory and computation, will open up a wide array of unique and exciting applications.

2.2.2 Dantzig Selector

2.2.2.1 Definition

An alternative approach to the Lasso, the Dantzig Selector was originally proposed by Candes and Tao [14] as a method for estimating a sparse parameter vector when the number of observations is much smaller than the number of predictors. More formally, suppose we have a response vector $\mathbf{y} \in \mathbb{R}^n$ such that $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\mathbf{X} \in \mathbb{R}^{n \times p}$ is a matrix of predictors, $\boldsymbol{\beta} \in \mathbb{R}^p$ is a sparse parameter vector of interest, and $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is a noise vector

such that $\epsilon_i \sim^{i.i.d.} N(0, \sigma^2)$. The Dantzig Selector solves the problem:

$$\min_{\boldsymbol{\beta}} ||\boldsymbol{\beta}||_1 \text{ subject to } \left|\left|\mathbf{X}^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right|\right|_\infty \leq \lambda, \tag{2.3}$$

for $\lambda > 0$. As λ increases, so does the influence of the sparsity assumption. Technically, it should be written as $\lambda\left(\sigma^2\right)$ because the value of λ is related to the noise level; we suppress the notation for convenience.

2.2.2.2 Properties

As with other regularization techniques, theoretical results regarding the Dantzig Selector require a number of somewhat restrictive assumptions. First, we assume that $\boldsymbol{\beta}$ is sufficiently sparse. In general, we require a handful of observations for estimation of each nonzero coefficient; exactly how many observations are required depends on the properties of the design matrix [14]. This assumption is necessary to ensure that we have a unique solution - otherwise, we might have $\mathbf{X}\boldsymbol{\beta} \approx \mathbf{X}\boldsymbol{\beta}'$ where $\boldsymbol{\beta}$ and $\boldsymbol{\beta}'$ have different nonzero elements, similar to the behavior of OLS estimates when $\text{rank}(\mathbf{X}) < p$. Results can be extended to a near-sparse $\boldsymbol{\beta}$ if we assume an appropriate level of decay in the coefficients [14]. With regard to the design, we assume $\mathbf{X}$ follows the *uniform uncertainty principle*, which places restrictions on the degree of correlation between predictors [14]. Unfortunately, this condition can be difficult to assess in practice, as it requires "checking the extremal singular values for exponentially many sub-matrices". Candes and Tao argue that these assumptions are not overly restrictive, citing that problems violating the *uniform uncertainty principle* would prove difficult for any type of estimator [14]. Regardless, it is unrealistic to check in practice [66] and can be difficult to satisfy if the number of predictors is large relative to the sample size due to unavoidably high correlations between independent predictors [19].

Assumptions satisfied, the Dantzig Selector has some nice properties. Notably, a non-asymptotic bound on the L_2 estimation error is within a $\log p$ factor of the achievable error rate if the true nonzero coefficients were known in advance (i.e., an oracle estimator). And because $\log p$ grows slowly for increasing p, the Dantzig Selector pays a small price for adaptively selecting the important predictors [14]. Additionally, and in contrast to the

Lasso, the Dantzig Selector is invariant to transformations applied to the data [14].

In the special case when the design matrix is orthogonal, the Dantzig Selector performs soft-thresholding:

$$\hat{\beta}_j = \max\left(|(\mathbf{X}^{\mathrm{T}}\mathbf{y})|_j - \lambda, 0\right) \operatorname{sign}\left((\mathbf{X}^{\mathrm{T}}\mathbf{y})_j\right). \tag{2.4}$$

Specifically, coefficients with an absolute value smaller than λ are set to 0; otherwise, coefficients are 'shrunk' towards 0, where the strength of shrinkage is determined by the magnitude of λ. In general, Dantzig Selector estimates have a downward bias, similar to the Lasso [14]. In the spirit of OLS-LARS, the Adaptive Lasso, and the Relaxed Lasso [49], a two-stage approach can potentially correct this bias (e.g., Gauss-Dantzig selector [35]).

2.2.2.3 Computation

The Dantzig Selector can be expressed as a linear programming problem, and therefore can be solved using standard algorithms. Briefly, linear programming is a technique for optimizing a linear objective function that is subject to linear equality and inequality constraints. Candes and Tao utilize a primal-dual interior point method, and have made their MATLAB routines available via the L_1-`magic` package [14]. However, one drawback is that this must be done over a user-defined grid of values for the tuning parameter; in contrast, the Lasso uses the LARS algorithm to compute solutions along a path of λ's, which is based on the number of piecewise-linear knots [17]. More recently developed path algorithms for the Dantzig selector include: (1) 'Primal-Dual pursuit' and (2) DASSO; the former is a homotopy (i.e., path-following) algorithm that employs strong duality between the primal and dual problems [6], while the latter constructs the entire piecewise linear solution path utilizing a simplex-like algorithm that resembles, and is computationally competitive with, LARS [36].

2.2.2.4 Comparisons with Lasso

For the $n \geq p$ case, the Lasso and Dantzig Selector share the same solution path if $\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$ is diagonally dominant (i.e., if $\mathbf{D} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$ then $D_{jj} \geq \sum_{i \neq j} |D_{ij}|, \ \forall i = 1, \ldots, p$) [50]. James et al. [36] consider the high dimensional case, deriving conditions for their equivalence

at a given value of λ. Special cases where the equivalence holds for all values of the tuning parameter in the $p > n$ case include: (1) orthogonal $\mathbf{X}$, and (2) $corr(\mathbf{x}_i, \mathbf{x}_j) = \rho$, $\rho \in [0, 1)$ for all predictor pairs (i, j) [36]. In general, when Lasso and Dantzig Selector are not equivalent, the latter will produce more sparse solutions (i.e., smaller L_1-norm) [36].

Under sparsity and restricted eigenvalue assumptions on $\mathbf{X}^{\mathrm{T}}\mathbf{X}$, Bickel et al. [9] derive oracle inequalities for L_2 prediction loss in general nonparametric regression; the Lasso and Dantzig Selector exhibit similar behavior. Further, in the general linear regression settings this similarity extends to L_p loss in $\boldsymbol{\beta}$, for $p \in [1, 2]$ [9, 72].

2.2.2.5 Extensions

Akin to the Adaptive/Relaxed Lasso, multi-stage and adaptive approaches have been developed to improve variable selection consistency and overcome the downward bias of coefficient estimates [35, 45, 16]. Antoniadis et al. [3] extend the Dantzig Selector to the Cox proportional hazards regression model, where the L_∞-norm is imposed on the score process. Liu et al. [44] propose the Group Dantzig selector and derive conditions under which it is equivalent to the Group Lasso. Li and Dicker [43] propose an adaptive Dantzig Selector for a response variable subject to right-censoring. James and Radchenko [35] extend the Dantzig Selector to GLMs, and propose a path algorithm for efficient optimization.

2.3 Methods

2.3.1 Flexible Dantzig Selector

2.3.1.1 Motivation

The fusion penalty was first introduced in the form of the Fused Lasso [63], and extended to the Generalized Lasso [65]; by imposing additional L_1 penalties on $\beta_i - \beta_j$ for variables i and j, coefficient pairs are encouraged to be similar. Fusion penalties have received a lot of attention in the context of the Lasso (Pairwise Fused Lasso [54]; Split Bregman method for large scale Fused Lasso [73]; Group Fused Lasso [2]), but extensions to the Dantzig selector are limited; Chen and Dalalyan [15] proposed a scaled version of the Dantzig selector that

can impose fusion penalties, but they restrict the number of constraints to be less than the number of predictors ($m < p$). As a result, their approach cannot be applied to problems that impose coefficient fusion and sparsity simultaneously.

2.3.1.2 Definition

In the spirit of Tibshirani and Taylor [65], we define the *Flexible Dantzig Selector (FDS)* estimator as the solution to:

$$\min_{\boldsymbol{\beta}} ||\mathbf{M}\boldsymbol{\beta}||_1 \text{ subject to } \left|\left|\mathbf{X}^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right|\right|_\infty \leq \lambda, \tag{2.5}$$

where $\lambda \geq 0$ and $\mathbf{M}$ is a fixed $(m \times p)$ constraint matrix describing the penalty structure. Penalty matrix $\mathbf{M}$ is user-defined, and thus can be uniquely constructed for a given problem; a special case, when $\mathbf{M} = \mathbf{I}_p$ the FDS devolves into a regular Dantzig Selector problem.

2.3.1.3 Flexible Penalty Matrix ($\mathbf{M}$)

An appealing aspect of the FDS is the inclusion of a flexible user-defined penalty matrix, allowing for the method to be utilized across a variety of applications. Of particular interest, if a problem has a known variable structure in undirected graph form, we can define a corresponding $\mathbf{M}$ to impose structure on the solution (Figure 2.1). For more details regarding the versatility of $\mathbf{M}$, we refer the reader to the original Generalized Lasso paper ([65]).

2.3.1.4 Computation

Similar to the Dantzig Selector, the Flexible DS can be solved by reformulating the problem as a linear programming problem and relying on standard algorithms (e.g., interior-point, simplex). Specifically, let $\mathbf{u}$ be a m-dimensional vector representing the constraints in the rows of $\mathbf{M}$, and re-express Equation 2.5 as the minimization of a $p + m$ parameter vector

subject to $2m + 2p$ constraints,

$$\min_{\boldsymbol{\beta},\mathbf{u}} \left\{ \begin{bmatrix} \mathbf{0}'_p & \mathbf{1}'_m \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \mathbf{u} \end{bmatrix} \right\} \text{ subject to } \begin{bmatrix} \mathbf{M} & -\mathbf{I}_m \\ -\mathbf{M} & -\mathbf{I}_m \\ \mathbf{X}'\mathbf{X} & \mathbf{0}_{p\times m} \\ -\mathbf{X}'\mathbf{X} & \mathbf{0}_{p\times m} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \mathbf{u} \end{bmatrix} \leq \begin{bmatrix} \mathbf{0}_m \\ \mathbf{0}_m \\ \lambda + \mathbf{X}'\mathbf{y} \\ \lambda - \mathbf{X}'\mathbf{y} \end{bmatrix}, \tag{2.6}$$

where $\mathbf{I}_k$ represents a k-dimensional identity matrix and $\mathbf{0}_k$ and $\mathbf{1}_k$ represent k-dimensional vectors of zeros and ones, respectively. Once formulated in this way, standard linear programming problem solvers can be employed to obtain the solution for a fixed λ. For example, the `linprog` function in `MATLAB`'s Optimization Toolbox or the `lpSolve/lpSolveAPI` packages in `R`. Of course, this approach may not be feasible as p and q grow large. Alternatively, we can utilize `CVX`, a `MATLAB` package for specifying and solving convex problems [25, 26].

We do not provide a path algorithm for determining a sequence of values for λ. However, it can be shown that $\lambda_{max} = \left|\left|\mathbf{X}^{\mathrm{T}}\mathbf{y}\right|\right|_\infty$, which makes user-specification of a sequence straightforward (i.e., a sequence of n values between 0 and λ_{max}. Warm-starts on subsequent iterations can help to improve computation time, although some solvers do not accept initial values (e.g., `linprog`).

2.3.1.5 Speed Tests

Speed tests were performed using the `linprog` function in MATLAB R2014b (PC Specs: Intel Core i5-2500 CPU (3.3GHz), 8GB of RAM).

The goal of the first test is to determine the impact of the *Number of Variables (p)* on the required computation time. To this end, we fix the sample size ($n = 100$) and test Equation 2.6 for varying p by:

1. Drawing $\mathbf{X}$ from a standard Normal distribution (i.e., uncorrelated predictors).
2. Generating $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\epsilon_i \sim \text{Normal}\left(0, \sigma^2\right)$ and $\boldsymbol{\beta} = (1, 1, 1, 1, 1, 0, 0, \ldots, 0)^{\mathrm{T}}$.
3. Computing the solution, $\hat{\boldsymbol{\beta}}_\lambda$, for a sequence of 50 equally-spaced λ values ($\lambda_{min} = 0$, $\lambda_{max} = \left|\left|\mathbf{X}^{\mathrm{T}}\mathbf{y}\right|\right|_\infty$).

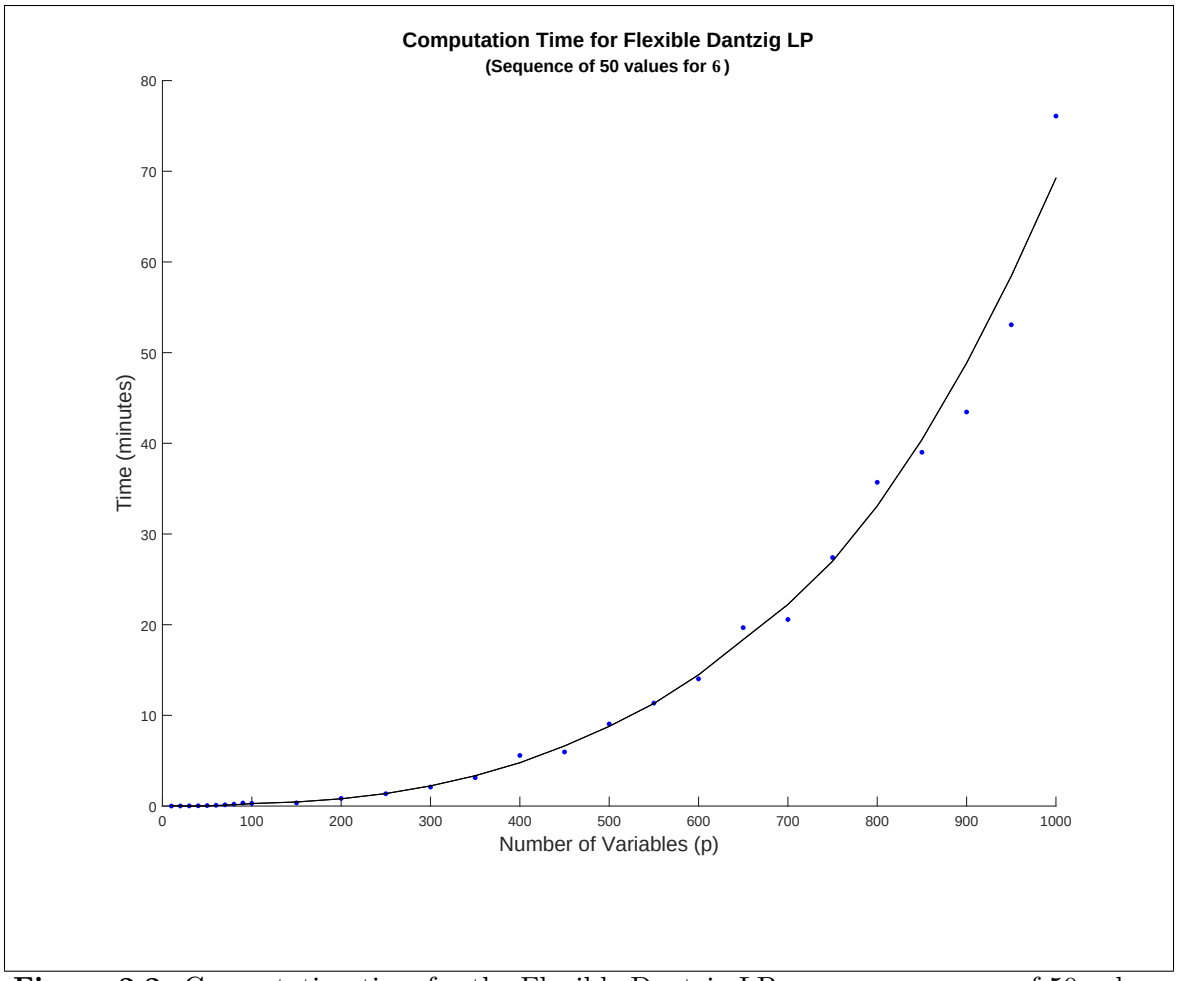

Figure 2.2: Computation time for the Flexible Dantzig LP across a sequence of 50 values for λ. Computation time increases exponentially with the number of predictors (p).

Overall, computation time increases exponentially as p increases (Figure 2.2), making this approach seemingly infeasible for large datasets. However, it should be noted that a significant portion of computation time is spent on smaller values of λ, corresponding to the least constrained solutions (Figure 2.3). Depending on the analysis goals, it may be reasonable to forgo computation of these less sparse coefficient vectors in the interest of time; when estimating coefficients using penalized approaches, we are not typically interested in the unconstrained solutions.

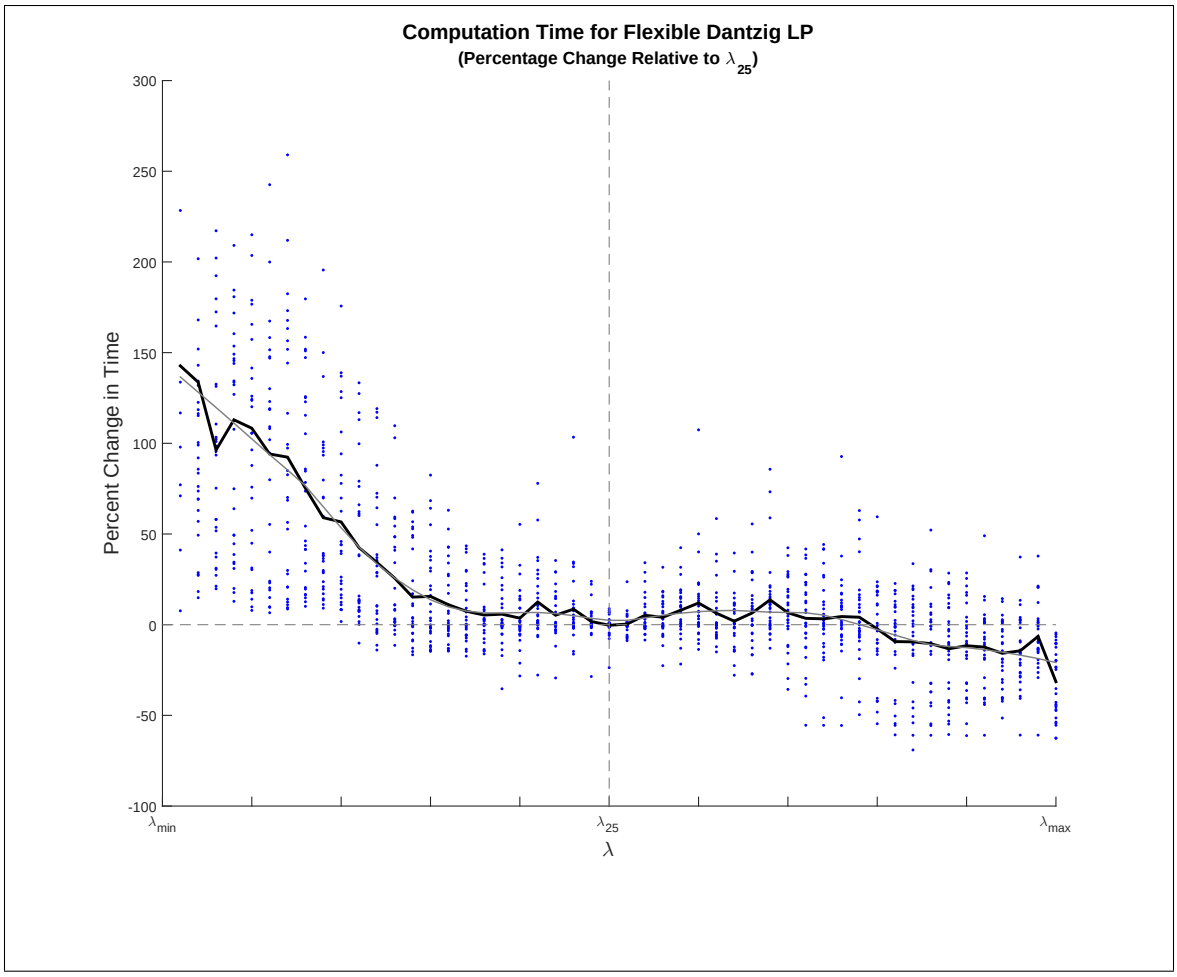

Figure 2.3: The percentage change in time (relative to the median value for λ) as a function of λ. Solutions that are less sparse, corresponding to smaller values of λ, take much longer for the algorithm to estimate. While computation time is much slower for less constrained solutions, there are limited gains in computation time for solutions corresponding to λ values beyond the median.

2.3.1.6 Variant 1: Proportional Weighting

As additional constraints are imposed, the quantity minimized by the Flexible DS ($||\mathbf{M}\boldsymbol{\beta}||_1$) can only increase. In other words, the contribution of each row of $\mathbf{M}$ is proportional to its magnitude relative to the L_1-norm of the entire vector, and thus will inevitably shrink as m grows. While this is not an issue when considering only sparsity constraints (i.e., $m = p$), as $m - p$ grows large relative to p the sparsity constraints may no longer have enough influence on the solution to force unimportant coefficients to zero. The issue stems from having a

flexible penalty structure that can handle many different types of penalties, but only utilizes a single tuning parameter. To overcome this issue, we propose including a unique tuning parameters for each type of penalty. For example, if $\mathbf{I}_p$ represents sparsity constraints and $\mathbf{M}_f$ represents fusion constraints on coefficient pairs, then our proportionally weighted penalty matrix takes the form $\mathbf{M}(\alpha) = \left(\mathbf{I}_p^{\mathrm{T}}, \alpha\mathbf{M}_f^{\mathrm{T}}\right)^{\mathrm{T}}$ for $\alpha > 0$. Of course, this reverts to the regular Flexible DS when $\alpha = 1$. In theory, if there were k different types of penalty structures (including sparsity) then we could incorporate additional tuning parameters $(\alpha_1, \ldots, \alpha_{k-1})$, at the obvious cost of increasing the dimensionality of the hyperplane over which these values must be tuned. Unfortunately, computation is already quite slow for a single tuning parameter, so this approach is not feasible for large p or m when relying on our current optimization algorithm.

In practice, a good strategy to select a sequence for α is to leverage an initial estimate and determine the value of α at which both types of penalties have equal overall contribution, or $\alpha^* = \left|\left|\mathbf{I}_p\hat{\boldsymbol{\beta}}^*\right|\right|_1 / \left|\left|\mathbf{M}_f\hat{\boldsymbol{\beta}}^*\right|\right|_1$. Then, consider α for a few different values for k, $\alpha = k\alpha^*$, where $k:1$ represents the ratio of the impact of $\mathbf{I}_p : \mathbf{M}_f$ (i.e., sparsity : fusion).

2.3.1.7 Variant 2: Adaptive Weighting

Concurrent with the literature, we propose an adaptive version of the Flexible Dantzig Selector that relies on an initial estimate, $\hat{\boldsymbol{\beta}}^*$. Adaptive weighting can improve estimation and model selection, especially when the imposed structure is not supported by the data. Particularly appealing because $\mathbf{M}$ is user-specified , the use of adaptive weighting can help to alleviate concerns associated with a misspecified penalty matrix. Given a diagonal weight matrix, $\boldsymbol{\Delta}$, where the j^{th} diagonal element, δ_j, corresponds to the constraint imposed in row j of $\mathbf{M}$ (for $j = 1, 2, \ldots, m$), we define the Adaptive Flexible Dantzig Selector (aFDS) as the solution to:

$$\min_{\boldsymbol{\beta}} ||\boldsymbol{\Delta}\mathbf{M}\boldsymbol{\beta}||_1 \text{ subject to } \left|\left|\mathbf{X}_j^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right|\right|_1 \leq \lambda\delta_j, \quad \forall j \in \{1, 2, \ldots, p\} \tag{2.7}$$

where the adaptive weights are defined as $\delta_j = \left|\left(\mathbf{M}\hat{\boldsymbol{\beta}}^*\right)_j\right|^{-\gamma}$ for $j = 1, 2, \ldots, m$ and $\gamma > 0$.

The addition of adaptive weights requires that the first p rows of $\mathbf{M}$ always correspond

to the sparsity constraints, because the values of δ_j for $j = 1, \ldots, p$ are used to construct predictor-specific constraints (λ_j) on the sup-norm of the current residuals. To emphasize this, the penalty matrix is represented as $\mathbf{M} = \left(\mathbf{I}_p^{\mathrm{T}}, \mathbf{M}_f^{\mathrm{T}}\right)^{\mathrm{T}}$, where $\mathbf{I}_p$ imposes the sparsity constraints and $\mathbf{M}_f$ is a $(m-p) \times p$ matrix that imposes the remaining constraints on the regression coefficients.

Suppose we let $\boldsymbol{\delta} = diag(\boldsymbol{\Delta})$ be a m-dimensional vector of data-adaptive constraint weights and let $\boldsymbol{\delta}_p$ represent a vector containing the first p weights corresponding to the sparsity constraints. We can obtain the solution for fixed λ using solvers for the following linear program:

$$\min_{\boldsymbol{\beta},\mathbf{u}} \left\{ \begin{bmatrix} \mathbf{0}_p^{\mathrm{T}} & \boldsymbol{\delta}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \mathbf{u} \end{bmatrix} \right\} \text{ subject to } \begin{bmatrix} \boldsymbol{\Delta}\mathbf{M} & -\mathbf{I}_m \\ -\boldsymbol{\Delta}\mathbf{M} & -\mathbf{I}_m \\ \mathbf{X}^{\mathrm{T}}\mathbf{X} & \mathbf{0}_{p\times m} \\ -\mathbf{X}^{\mathrm{T}}\mathbf{X} & \mathbf{0}_{p\times m} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \mathbf{u} \end{bmatrix} \leq \begin{bmatrix} \mathbf{0}_m \\ \mathbf{0}_m \\ \lambda\boldsymbol{\delta}_p + \mathbf{X}^{\mathrm{T}}\mathbf{y} \\ \lambda\boldsymbol{\delta}_p - \mathbf{X}^{\mathrm{T}}\mathbf{y} \end{bmatrix} \tag{2.8}$$

2.3.1.8 Variant 3: Weighting Observations

Data of varying quality warrants observation-specific weights. To account for heteroscedastic data, we introduce a diagonal weight matrix, $\mathbf{W}$, where w_{ii} represents the weight associated with observation i, for $i = 1, 2, \ldots, n$, and we define the Weighted Flexible Dantzig Selector as the solution to:

$$\min_{\boldsymbol{\beta}} ||\mathbf{M}\boldsymbol{\beta}||_1 \text{ subject to } \left|\left|\mathbf{X}^{\mathrm{T}}\mathbf{W}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right|\right|_\infty \leq \lambda \tag{2.9}$$

We can obtain coefficient estimates for a fixed λ by using standard solvers for the following linear program:

$$\min_{\boldsymbol{\beta},\mathbf{u}} \left\{ \begin{bmatrix} \mathbf{0}_p^{\mathrm{T}} & \mathbf{1}_m^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \mathbf{u} \end{bmatrix} \right\} \text{ subject to } \begin{bmatrix} \mathbf{M} & -\mathbf{I}_m \\ -\mathbf{M} & -\mathbf{I}_m \\ \mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} & \mathbf{0}_{p\times m} \\ -\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} & \mathbf{0}_{p\times m} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \mathbf{u} \end{bmatrix} \leq \begin{bmatrix} \mathbf{0}_m \\ \mathbf{0}_m \\ \lambda + \mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{y} \\ \lambda - \mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{y} \end{bmatrix} \tag{2.10}$$

2.3.1.9 Bootstrap-Enhanced Estimation

One drawback regarding the selection of λ is that it can vary considerably as a result of minor perturbations of the data. The 'jittery' behavior of Dantzig Selector solutions relative to the Lasso (as a function of λ) implies that using an information criteria approach yields an untrustworthy (i.e., highly variable) set of important predictors. Given that standard error estimates are not available, this potential lack of consistency is concerning. To overcome this issue in the Lasso setting, Bunea et al. [10] introduce a bootstrap-enhanced version that attaches a measure of uncertainty to each predictor. Derived from the proportion of nonzero coefficient estimates across bootstrap samples, they recommend choosing predictors that have an inclusion probability exceeding 0.5. However, in practice we found that the recommended 0.50 threshold is not always appropriate, because the fitting method and tuning procedure can influence the resulting probabilities; some approaches are more selective than others, and this is reflected in variable inclusion probabilities.

We propose a bootstrap-enhanced version of the Flexible DS. Specifically, we draw B samples with replacement from the original dataset and estimate the coefficients for each using the Flexible DS. Due to computational considerations, we recommend forgoing cross-validation and instead selecting the tuning parameter by minimizing prediction error in a separate tuning dataset that is composed of observations not included in the current bootstrap sample. Selection probabilities for each predictor are then derived from the proportion of samples with a nonzero coefficient estimate.

To further improve the bootstrap-enhanced approach, we consider imposing a weight on the selection probabilities that is proportional to the sign-consistency of the coefficients. This will have no impact on predictors that consistently have positive (or negative) coefficient estimates across bootstrap samples, but it will serve to attenuate selection probabilities for sign-inconsistent predictors. This makes sense intuitively, because if the direction of relationship between a predictor and the outcome changes across replicates of the data, then it is not likely to be a strong or useful predictor; certainly, we cannot comfortably interpret the resulting estimate. An additional benefit, selection probabilities for non-sparse fitting methods (e.g., OLS, Ridge) can now be less than one, ostensibly performing variable

selection with methods that typically cannot.

We also consider bootstrap-enhanced coefficient estimates, calculated as the average across all bootstrap samples. However, the average of many sparse estimates is no longer sparse; it may seem counterintuitive that our bootstrap-enhanced approach produces both selection probabilities and nonzero coefficients for each predictor. We construct a sparse estimate by defining a probability threshold and assigning a zero coefficient to any predictor that does not exceed it.

2.3.1.10 Randomized Estimation

One of the more substantial drawbacks regarding the Lasso and Dantzig selector is inaccuracy when facing multicollinearity. Ridge regression is capable of handling data with complex predictor correlation structures, and an application of the same idea in the Lasso framework evolved into the Elastic Net. Unfortunately, it is not clear how to directly incorporate a Ridge-type penalty in the Flexible DS framework.

Wang et al. [70] developed the Random Lasso as an approach for overcoming multicollinearity in the Lasso setting, and we can apply the same algorithm in the Flexible DS setting. An additional benefit, Random Lasso solutions can have more the p nonzero coefficients. Briefly, the Random Lasso involves two-dimensional bootstrapping. At each iteration we draw a sample, with replacement, from the original data, and we also select a random subset of the predictors (without replacement). This is done in two stages, with the predictor-selection completely random in the first stage and proportional to the magnitude of first-stage estimates during the second stage. At each iteration the estimates for excluded predictors (via subset-selection or the Lasso) are set to zero, and end-of-stage estimates are obtained by averaging across all iterations.

Like the bootstrap enhancement, the Random approach relies on the selection of λ at each iteration, and the results tend to be tuning-method-dependent. Further, there are two additional tuning parameters, q_1 and q_2, corresponding to the size of the predictor-subset at each iteration for stages one and two, respectively. Of course, the results tend to depend heavily on the values of q_1 and q_2 in practice. Also, this approach does not directly yield sparse solutions; when using this approach for model selection, a final thresholding step is

necessary.

While the Random Lasso relies on marginal selection probabilities, we propose a single-stage approach that considers conditional selection probabilities. For a given predictor, we only consider bootstrap samples that it is included in the randomly selected subset.

2.4 Results

2.4.1 Illustrations

Prior to conducting simulations and formal data analyses, we illustrate the application of the Flexible Dantzig Selector in a variety of settings.

Illustration 1 - An Arbitrary Graph

Our first illustration is designed to make evident the impact of incorporating variable structure on the resulting coefficient estimates. The data are generated according to Scenario 1(a) in the simulation below (Table 2.3). Briefly, we fit the Flexible DS to the data with three different variables structures: (1) none (i.e., Dantzig selector), (2) correct, and (3) random. Graphical versions of the variable structures and coefficient paths for the corresponding Flexible DS fits are shown in Figure 2.4, where the important predictors are represented in red.

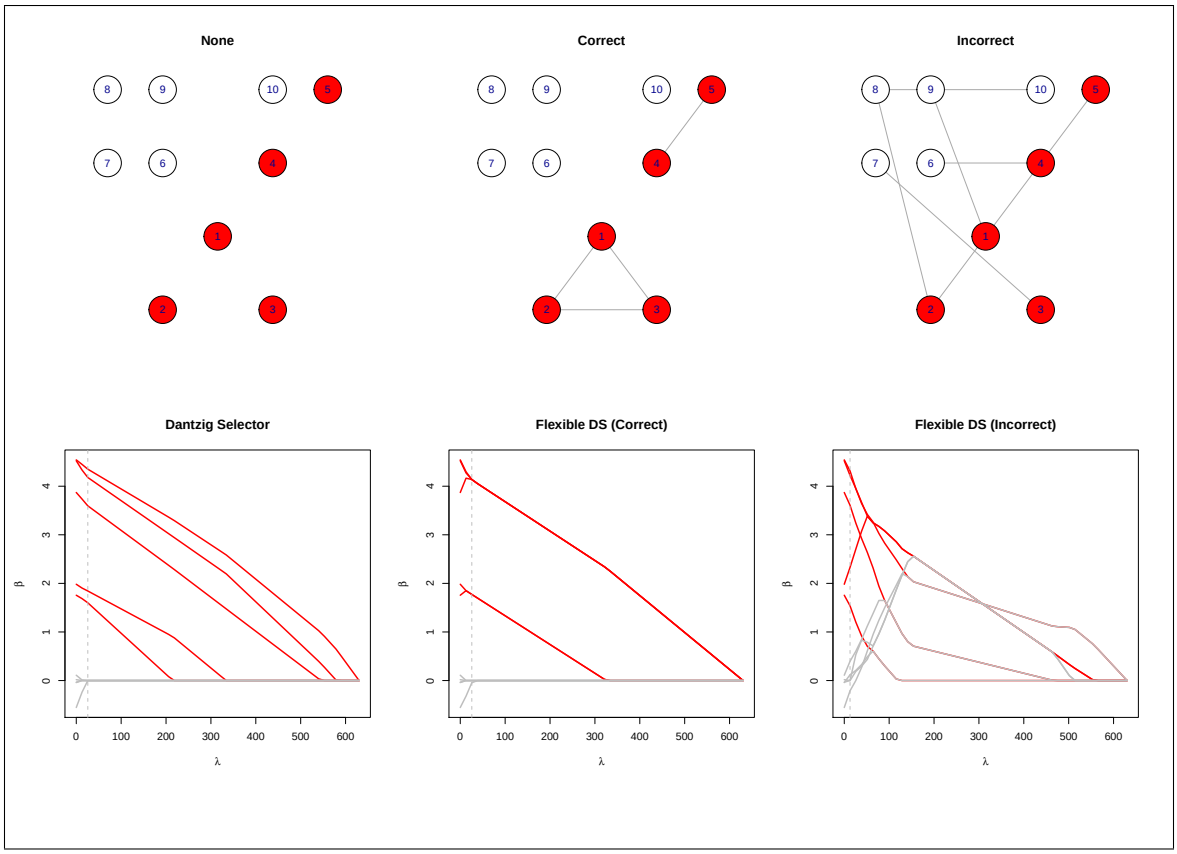

Figure 2.4: Each column corresponds to a version of the Flexible Dantzig Selector. The first row illustrates graphically the variable structure imposed in penalty matrix $\mathbf{M}$, and the second row shows the corresponding coefficient paths versus the tuning parameter λ. In particular, the correct structure is capable of correctly fusing coefficient groups, while the incorrect structure clearly has a negative impact as there no longer exists a value of λ for which the correct model is selected.

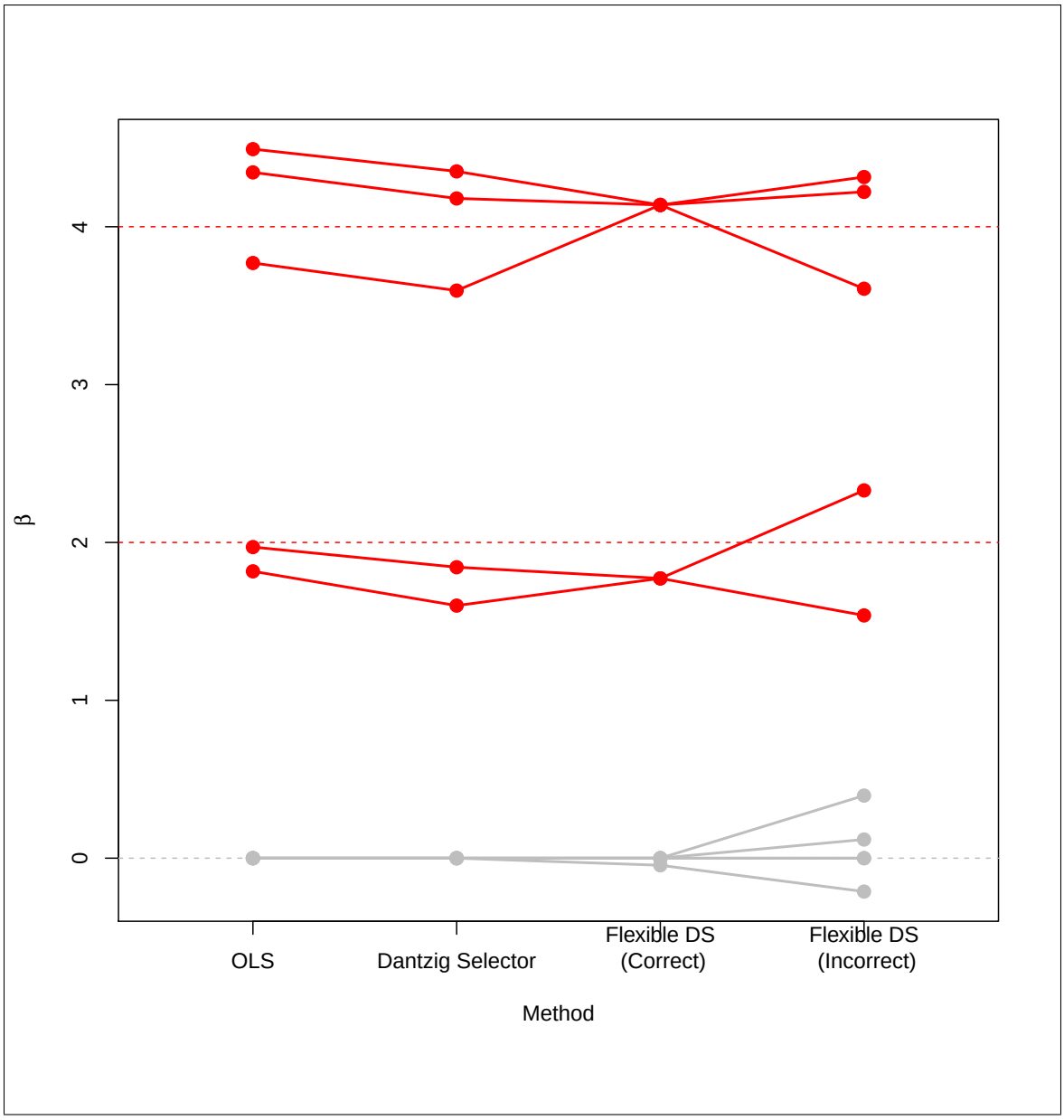

Figure 2.5: Coefficient estimates for each fitting method. From left-to-right: (1) Oracle OLS, (2) Dantzig Selector, (3) Flexible DS with the correct variable structure, and (4) Flexible DS with a random variable structure. We selected λ by minimizing the prediction error in the tuning set. When imposing the correct variable structure, we correctly fuse coefficients within predictor groups. However, the incorrect structure causes coefficients to behave more erratically and forces many insignificant predictors to be nonzero.

Illustration 2 - Spatial Graph

Our second illustration involves an application to spatial data. In particular, we treat each of the $p = 50$ states as predictors and fit the Flexible DS using three different variable structures (Figure 2.6). Data are generated according to Scenario 2(a) in the simulations below (Table 2.3); there are only 4 nonzero coefficients. With many more predictors than our previous example, the coefficient paths appear to be more unstable; however, imposing the adjacency constraint helps to stabilize estimates for the important predictors (Figure 2.7). We select the optimal λ for each method by minimizing the prediction error in a separately generated tuning dataset. The resulting coefficient estimates, superimposed on a map of the United States, show the benefits and drawbacks of imposing correct and incorrect penalty structures, respectively (Figure 2.8). Although the Dantzig Selector correctly identifies the important states among some additional irrelevant ones, the Flexible DS select only the important states and further assigns them identical coefficient estimates. However, a random structure causes some obvious problems, and many irrelevant states are assigned nonzero coefficients and we almost fail to detect the importance of Montana.

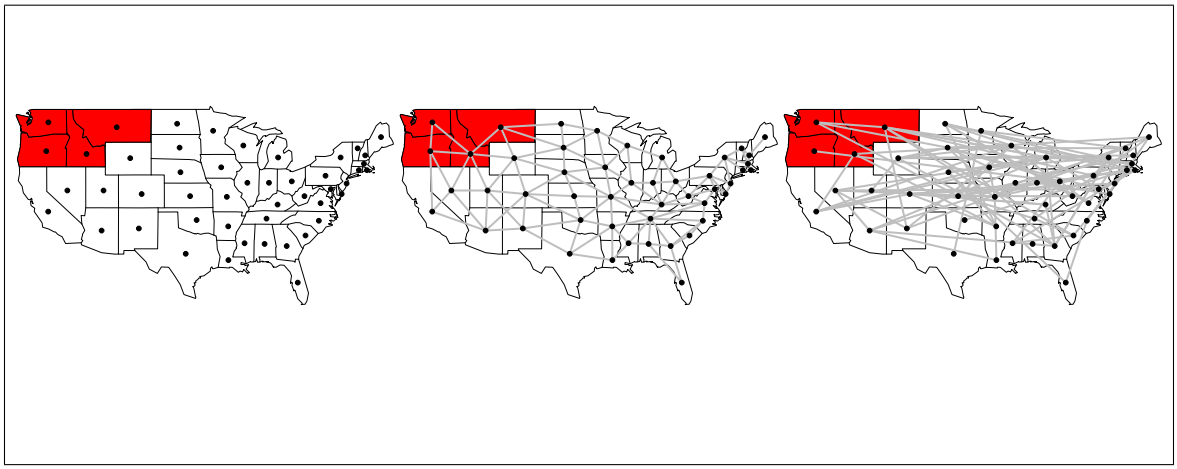

Figure 2.6: Undirected graphs are superimposed on the true coefficient vector (red indicates nonzero coefficients) for each of the three penalty structures used to fit the Flexible DS. Variable structures include: none (left), adjacent neighbors (middle), and random neighbors (right). Alaska and Hawaii (not shown) have coefficients equal to 0, and no neighbors.

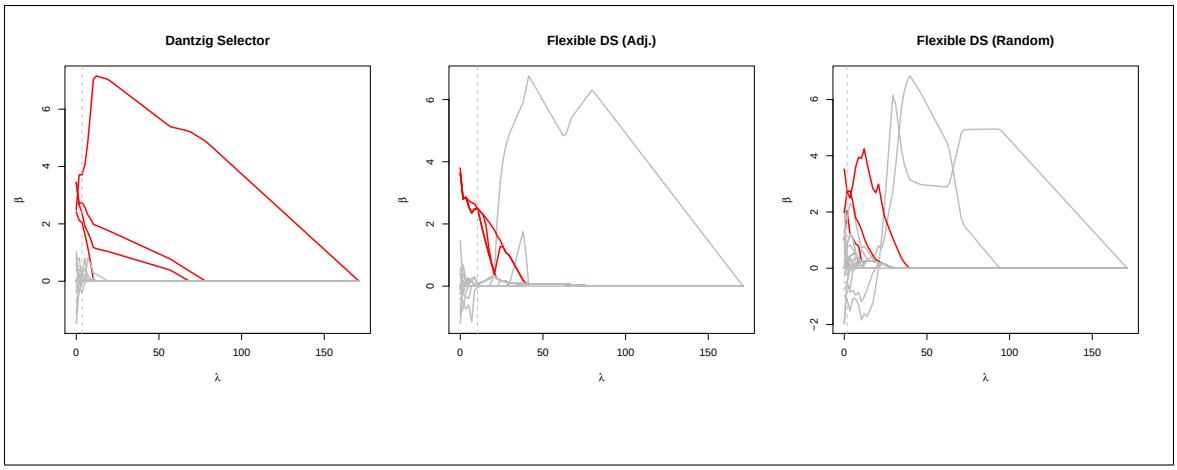

Figure 2.7: Coefficient estimates along a sequence of λ values for the Dantzig Selector (left), the Flexible DS with adjacent-neighbor fusion (middle), and the Flexible DS with random fusion (right). Red and gray lines represent the important and unimportant predictors, respectively, and the dotted gray line is the value of λ that minimizes prediction error in the tuning dataset.

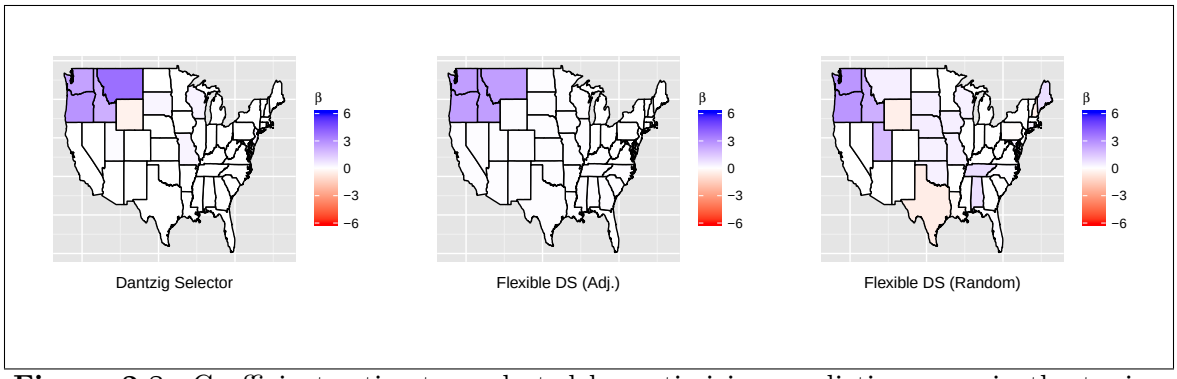

Figure 2.8: Coefficient estimates, selected by optimizing prediction error in the tuning dataset, corresponding to the Dantzig Selector (left), the Flexible DS with adjacent-neighbor fusion (middle), and the Flexible DS with random fusion (right). Notably, the Flexible DS with a correct penalty structure identifies the important predictors and correctly detects that their relationship with the outcome is equivalent (i.e., their coefficient estimates are identical). However, an incorrect penalty structure can cause serious problems with the quality of the resulting estimates.

2.4.2 Simulations

We conduct a large simulation to evaluate the performance of the Flexible Dantzig Selector. Our first goal is to illuminate the impact of constraining coefficient estimates based on the underlying variable structure. In particular, we anticipate improved model fit when imposing a correct variable structure. Our second goal is to emphasize the impact of adaptive weighting on coefficient estimates, especially when the variable structure is misspecified. Given its data-driven origins, we expect adaptive weighting to provide substantial improvements, assuming a good initial estimate ($\boldsymbol{\beta}^*$) is available; for our simulations, we use the Full OLS solution as the initial estimate for the Adaptive FDS. For our simulation, we consider various scenarios to assess the comparative performance of fitting methods based on their ability to: (1) predict future observations, (2) estimate the coefficients, and (3) select the important predictors (Table 2.1).

At each of the B iterations of our simulation, we generate three datasets: (1) training, (2) tuning, and (3) testing. The training set is used to estimate coefficients for each method. For methods that yield estimates across a range of tuning parameter values, we apply these estimates to the tuning set and select the one that minimizes the prediction error. After computing estimation and model selection error, the estimates are applied to the test dataset to measure prediction accuracy.

For all simulations in this chapter, we use training and tuning datasets of size n, and a test dataset of size $4n$.

Table 2.1: Simulations – Measures of Error

Type	**Formula**
Prediction Error	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
Estimation Error	$\sqrt{\frac{1}{p}\sum_{j=1}^{p}\left(\beta_j - \hat{\beta}_j\right)^2}$
Model Selection Error	$\frac{1}{p}\sum_{j=1}^{p}\left(I\left\{\beta_j \neq 0\right\} - I\left\{\hat{\beta}_j \neq 0\right\}\right)^2$

Table 2.2: Simulations – Fitting Methods

Method	**Estimation Details**	$\boldsymbol{\beta}^*$	**Path Length**	**Tuning**
Oracle	True coefficient vector		1	x
Oracle OLS	OLS fit to true predictors		1	x
Full OLS	OLS fit to all predictors		1	x
Ridge	`glmnet()`		50	✓
Lasso	`glmnet()`		50	✓
Dantzig Selector	Solve equation 2.6		50	✓
Flexible DS (correct)	Solve equation 2.6		50	✓
Flexible DS (random)	Solve equation 2.6		50	✓
Adaptive DS	Solve equation 2.8	Full OLS	50	✓
Adaptive FDS (correct)	Solve equation 2.8	Full OLS	50	✓
Adaptive FDS (random)	Solve equation 2.8	Full OLS	50	✓

2.4.2.1 Description of Scenarios

Table 2.3: Scenario Details

	Scenario 1			**Scenario 2**		
	(a)	(b)	(c)	(a)	(b)	(c)
$n_{train/tune/test}$	80/80/320			100/100/400		
p	10			50		
$\sum_j I(\beta_j \neq 0)$	5			4	10	25
σ	3	10	25	3	15	50
Σ_{ij}	ρ			ρ		
ρ	0.2	0.5	0.8	0.5		
$\approx R^2$	0.92	0.63	0.29	0.80	0.50	0.40

Scenario 1: The first scenario involves only $p = 10$ predictors, and half are associated with the outcome. Figure 2.9 illustrates the true underlying variable structure for this sce-

nario; there are two groups of significant predictors, and five ungrouped predictors that are irrelevant to the outcome. We construct three separate versions of this scenario along the difficulty spectrum, including: (a) Easy $\{\sigma = 3,\ \rho = 0.2\}$, (b) Medium $\{\sigma = 10,\ \rho = 0.5\}$, and (c) Hard $\{\sigma = 25,\ \rho = 0.8\}$. For all difficulty levels, we assume a compound-symmetric structure for the predictor correlations ($\Sigma_{ij} = \rho$).

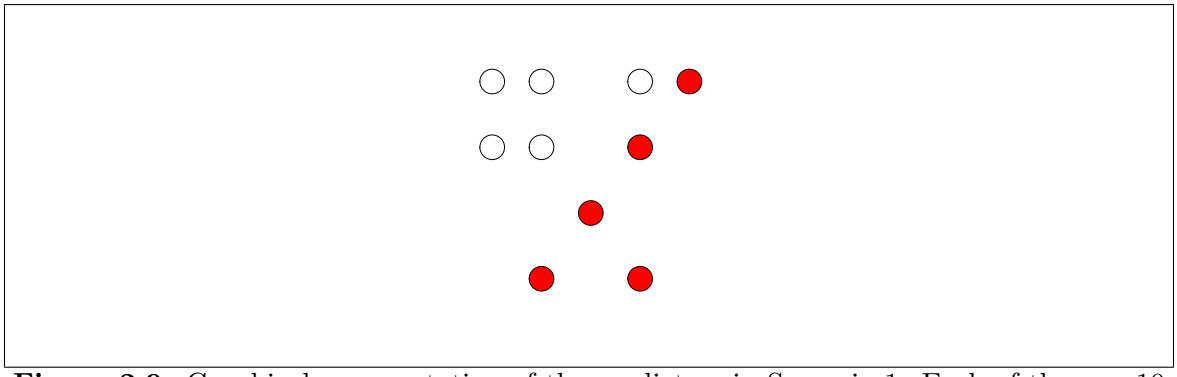

Figure 2.9: Graphical representation of the predictors in Scenario 1. Each of the $p = 10$ nodes represent a predictor, and important predictors are red.

Scenario 2: The second scenario is modeled after a state-specific analysis of the United States. There are $p = 50$ predictors, and between-predictor correlation is set to $\rho = 0.5$. We set $n = 100$ and consider three separate scenarios, with decreasing signal quality, that range in noise level ($\sigma = \{3, 15, 50\}$) and the number of important predictors (Figure 2.10).

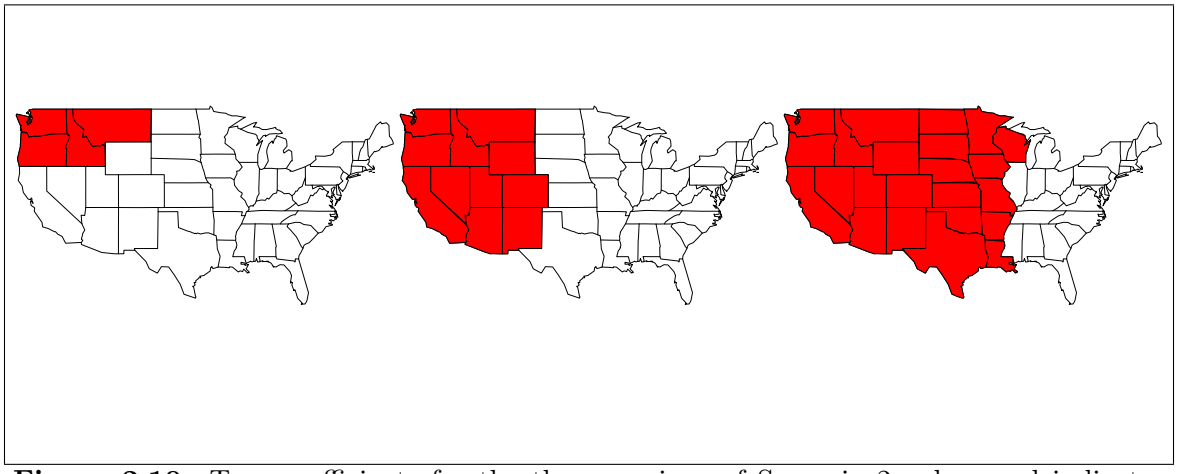

Figure 2.10: True coefficients for the three versions of Scenario 2, where red indicates nonzero.

2.4.2.2 Simulation – Goal 1

Our first simulation goal is to determine if the Flexible Dantzig Selector outperforms popular methods that cannot incorporate a known underlying variable structure. Further, we consider a randomly defined (i.e., incorrect) variable structure in addition to the correct structure, to determine the impact of incorrect specification on the results.

Scenario 1 is a relatively simple example that should highlight the benefits of imposing structure on the resulting coefficient estimates; we expect the Flexible Dantzig Selector with correct structure to outperform all other methods (expect the Oracles) across our three criteria for evaluation. In fact, our results indicate that the FDS (correct) provides the best predictions in versions (a) and (b) of Scenario 1, and is only slightly outperformed by Ridge in version (c); FDS (correct) has the smallest prediction error among all sparsity estimation methods in Scenario 1, and also outperforms Oracle OLS (Table 2.4). This trend continues regarding estimation, with FDS (correct) performing the best, with the exception of another loss to Ridge in version (c). While FDS (correct) cannot hope to defeat the Oracles with regard to model selection, the FDS (correct) outperforms all remaining contenders. Unfortunately, where we see success for the correct variable structure, there is a comparable lack of success when imposing an incorrect variable structure; the FDS (incorrect) performs worst across the board in versions (a) and (b) of Scenario 1, and only marginally recovers in version (c). In version (c), the strong predictor-correlations and increased noise likely benefit from structure, regardless of its accuracy, to help stabilize coefficient estimates, which may help to explain the performance of FDS (incorrect). But the general message is quite clear: incorporating structure can be beneficial when correctly specified, but harmful otherwise.

Scenario 2 is more difficult for the Flexible Dantzig Selector, computationally. We anticipate seeing improvements in prediction, estimation, and model selection when using the correct structure, but the increased computational demands suggest the benefits must be substantial to justify its use. Our simulation results indicated that the Lasso had the optimal performance in version (a) of Scenario 2 across all criteria, which makes sense given the sparsity of the true coefficient vector; although they were outperformed, Dantzig Selector

and FDS (correct) were close behind in estimation and prediction, and the former in model selection (Table 2.4). In general, FDS (correct) and Ridge provided the best predictions and estimates in versions (b) and (c) Scenarios 2, with the latter performing slightly better. Notably, there are large differences between the FDS (correct) and the Dantzig selector for versions (b) and (c) of Scenario 2, suggest that FDS (correct) is capable in both sparse and non-sparse settings. Regarding model selection, the Lasso actual performs best across all versions of Scenario 1, with the Dantzig Selector close behind; regardless of structure, the Flexible DS does not select the correct model. A closer look at the resulting coefficients indicates that in many iterations FDS (correct) yields coefficient estimates that are small, nonzero, and equivalent, the result of many more fusion than sparsity constraints. Proportional or adaptive weighting may useful for balancing the influence of constraint-types and pushing these negligible estimates to exactly zero.

As expected, the [Flexible] DS takes much longer than the Lasso to obtain coefficient estimates. As illustrated in Figure 2.2, computation time increases quadratically/exponentially with p when using our algorithm. Computational demands for Lasso grow at a much slower rate, evident in our simulation results; while the DS takes about 20 times longer than Lasso in Scenario 1, it takes over 100 times as long in Scenario 2. Computation time is further increased as we introduce additional constraints; in Scenario 2, the Flexible DS takes over 250 times as long as Lasso and over twice as long as the DS.

2.4.2.3 Simulation – Goal 2

We have illustrated that substantial errors can be incurred on our resulting estimates if the predictor structure is incorrectly specified. Our second simulation goal is to determine if we can incorporate adaptive weighting as a means of overcoming an incorrect structure, or further improving upon the results of a correctly-specified predictor structure.

In general, we expect adaptive weighting to lead to improvements in prediction, estimation, and model selection accuracy. When the correct structure is specified, the main purpose of adaptive weighting is to push small nonzero coefficients to exactly zero; likely, accuracy improvements will only be obvious in model selection for the Adaptive FDS (correct). However, adaptive weighting should lead to improvements across all facets when an

incorrect structure is specified. Of course, adaptive weighting is based on an initial estimate, the quality of which will likely have an impact on the results.

Our simulation results for Scenario 1 were only partially in agreement with our expectations (Table 2.5). In particular, version (a) highlights the benefits of adaptive weighting as the previous top-performer, FDS (correct), is outdone only by its adaptive counterpart. Although the gains are minimal regarding estimation and prediction, there is nearly a three-fold improvement in model selection (0.78 vs. 2.12) that suggests small negligible coefficients are being successfully pushed toward zero. Similarly, the benefits of adaptive weighting for an incorrect structure are evident across all criteria, but most obvious with model selection; an adaptively-weighted incorrect structure yields more accurate models than the unweighted DS. However, versions (b) and (c) of Scenario 1 tell a different tale, as the adaptive versions tend to provide worse predictions and coefficient estimates than their unweighted counterparts, and select more inaccurate sets of predictors. This may be a result of a poor initial estimate, as Full OLS is unequivocally the worst method in these settings (Table 2.4). A better initial estimate may lead to improved performance for the adaptive versions.

Our simulation results for Scenario 2 illustrate the performance of these methods as sparsity decreases and noise increases (Table 2.5). When there is a very sparse coefficient vector (i.e., $< 10\%$ nonzero) and a strong signal (i.e., version (a)), we see clear benefits to adaptive weighting as the Adaptive FDS (correct) has the best overall performance. Most notably, it is the only method to provide better predictions and estimates than Oracle OLS, and it only misses the correct model by an average of less than one predictor. Also interesting in version (a), the Adaptive FDS (incorrect) performs quite well, earning second place in prediction and estimation and a decisive third place in model selection; compared to the non-adaptive methods, the Adaptive FDS (incorrect) is best. Even when given an incorrect structure, utilizing adaptive weights leads to a model that, on average, contains one wrongly-identified predictor, which is 28 less than its non-adaptive counterpart and 9 less than regular sparsity-inducing methods (i.e., Lasso, DS).

However, adaptive versions do not perform as well in versions (b) and (c) of Scenario 2. In (b), the adaptive versions provide the least accurate predictions and coefficient estimates,

outperforming only the Full OLS (Table 2.4). The same is true for version (c), although here the Oracle OLS also performs poorly. Fortunately, the adaptive versions see a benefit with regard to model selection, with even the Adaptive FDS (incorrect) providing more accurate predictor-sets than the Dantzig Selector in (b). But the adaptive versions are universally unsubstantiated in version (c), leading to worse prediction, estimation, and model selection relative to their unweighted counterparts. Once again, this is very likely the result of a poor initial estimate, as Full OLS fails extravagantly in versions (b) and (c) of Scenario 2. Given its solid performance, ease of estimation, and non-sparsity, we instead consider using Ridge as an initial estimate for the Adaptive FDS in our data analysis.

Although not as evident in Scenario 1, computation times are slightly reduced for the Adaptive FDS relative to the FDS. However, the DS is faster than the Adaptive DS. This suggests that weighting the fusion penalties can lead to reduced computation time, which is appealing given that introducing approximately 100 pairwise fusion constraints to the DS problem increased computation 2.5 times. It may also be reasonable to completely remove any constraints with weights not exceeding a predefined threshold, as their influence will be minimal on the solution but maximal on computation time.

2.4.3 Data Analysis

2.4.3.1 Spatially-Informed Test Statistic Thresholding

We apply the Flexible Dantzig Selector to a two-dimensional slice of the brain, where voxel values represent univariate test statistics derived from a functional magnetic resonance imaging (fMRI) brain activation study. Generally, these test statistic maps are derived from fMRI data on a group of subjects that has completed specific tasks, with the test statistics reflecting estimated differences between task conditions (e.g., easy vs. difficult, happy faces vs. sad faces). However, these test statistics are usually obtained via massive univariate analysis; a separate linear model is fit at each voxel, and spatial information is ignored. Common approaches for subsequently determining regions with significant activation include: familywise error correction (Bonferroni), false discovery rate ([24]), and Gaussian random-field theory ([71]). By assuming all voxels are independent, the Bonfer-

roni correction tends to be too conservative in this setting. The latter methods are capable of accounting for the spatial relationship between voxels, but they require strong assumptions. In particular, the assumptions of Gaussian random-field theory require the data be reasonably smooth, which may not always be reasonable in the brain. For example, bilateral voxels that are in close physical proximity but lie in different hemispheres, and are therefore separated by sulci (i.e., significant divides between brain regions), often violate spatial smoothness assumptions.

For this example we are simply using the Flexible Dantzig Selector as a tool for thresholding, so $\mathbf{X} = \mathbf{I}_n$. We compare soft-thresholding solutions to the comparable solutions from the Flexible DS with a penalty matrix that imposes both sparsity and two-dimensional fusion (Figure 2.11). Our results indicate that the Flexible DS produces solutions that are more spatially coherent, reflected in the fewer number of clusters; the FDS produces a more clinically interpretable result.

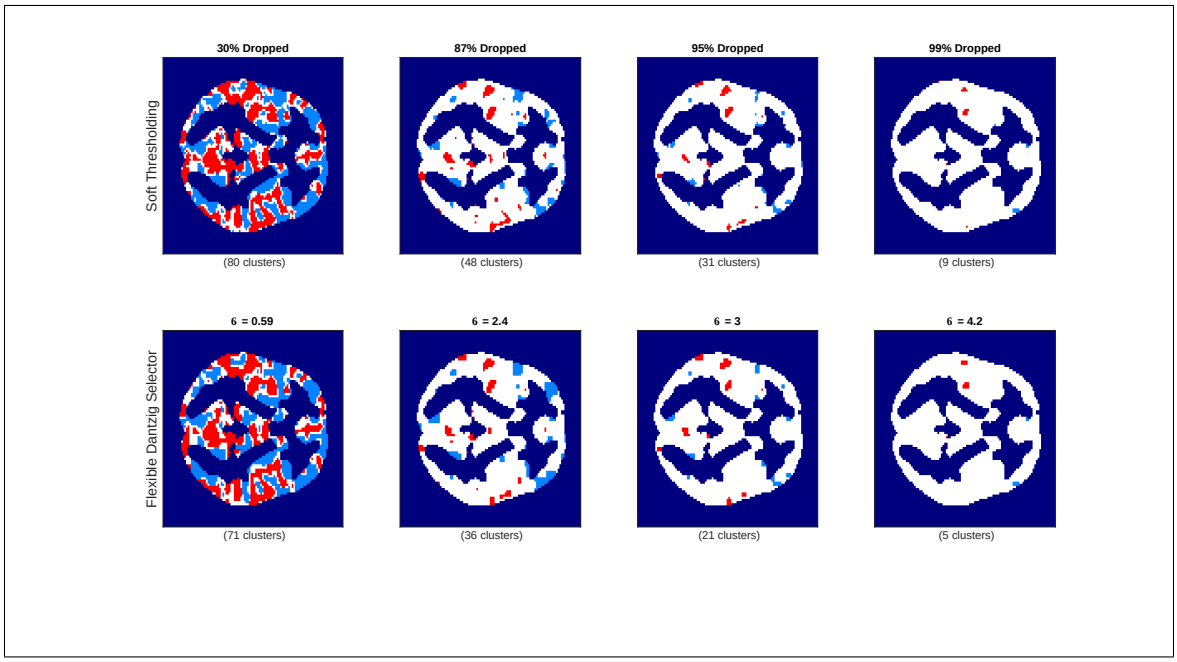

Figure 2.11: Thresholded coefficients that are estimated from a massive univariate linear model. Soft-thresholding results (top row) and corresponding Flexible DS solutions (bottom row) at comparable thresholds, where FDS includes penalties for sparsity and two-dimensional fusion. The Flexible DS has fewer clusters at any given threshold, reflecting increased spatial consistency relative to soft-thresholding.

Alternatively, we may be interested in grouping the voxels, perhaps for constructing regions of interest (ROIs) or classifying tissue-types. As an alternative to clustering techniques, many of which cannot account for spatial information, this goal can be achieved by applying the Flexible DS sans sparsity constraints.

2.4.3.2 The Alzheimer's Disease Neuroimaging Initiative (ADNI)

Background/Introduction

Alzheimer's disease (AD) is a serious mental illness that affects an estimated 5.3 million Americans; it is the most common cause of dementia among the elderly. Characterized by a progressive cognitive decline, AD has been notoriously difficult to diagnose due to symptom-overlap with other mental disorders; until recently, AD could only be confirmed posthumously. The Alzheimer's Disease Neuroimaging Initiative (ADNI) is a longitudinal study designed to track AD biomarkers, identify at-risk patients, and evaluate the efficacy of novel treatments. Already, researchers have identified several AD biomarkers, including: variations of the APOE gene, amyloid-beta build-up in the brain (detected via PET), decreases in brain volume (i.e., tissue degeneration), and cognitive decline as measured by various clinician-administered mental evaluations.

The Data

The data for this project are derived from the publicly available ADNI database (adni.loni.usc.edu). The ADNI was launched in 2003 by the National Institute on Aging (NIA), the National Institute of Biomedical Imaging and Bioengineering (NIBIB), the Food and Drug Administration (FDA), private pharmaceutical companies and non-profit organizations, as a 60 million dollar, 5 year public-private partnership. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). Determination of sensitive and specific markers of very early AD progression is intended to aid researchers and clinicians to develop new treatments and monitor their effectiveness, as well as lessen the time and cost of clinical trials.

The Principal Investigator of this initiative is Michael W. Weiner, MD, VA Medical

Center and University of California - San Francisco. ADNI is the result of efforts of many co-investigators from a broad range of academic institutions and private corporations, and subjects have been recruited from over 50 sites across the U.S. and Canada. The initial goal of ADNI was to recruit 800 subjects but ADNI has been followed by ADNI-GO and ADNI-2. To date these three protocols have recruited over 1500 adults, ages 55 to 90, to participate in the research, consisting of cognitively normal older individuals, people with early or late MCI, and people with early AD. The follow up duration of each group is specified in the protocols for ADNI-1, ADNI-2, and ADNI-GO. Subjects originally recruited for ADNI-1 and ADNI-GO had the option to be followed in ADNI-2. For up-to-date information, see www.adni-info.org.

In particular, we focus on the `UA - MRI SPM Voxel Based Morphometry (VBM) Analysis [ADNI1]` dataset, made available through the ADNI database. This data consists of VBM measures for each subject regarding 116 brain regions defined by the Automatic Anatomical Labeling (AAL) template [68], where VBM measures are derived from magnetic resonance images (MRI). VBM is a voxel-specific measure of brain volume, corrected for size and shape differences across subjects; preprocessing involves registering each subject's brain to a common template and spatially-smoothing the data to correct for minor registration inconsistencies [5]. Although measured at the voxel level, our data reflects the average across all voxels within each region of interest (ROI); although information is inevitably lost, this helps to overcome common criticisms related to image registration and smoothing. The data are longitudinal – subjects have data from MRI scans performed at baseline, in addition to 6 and 12 month follow-ups – but we only consider baseline observations to avoid violating the independence assumption.

VBM data demands consideration when determining potential predictors associated with Alzheimer's disease, because it is known to reduce brain size. Many studies have previously used VBM to discriminate between healthy controls and varying levels of disease progression [29, 42, 39]; Ferreira et al. [20] conducted a meta-analysis of VBM studies and identified atrophy in the left medial temporal lobe, specifically the left hippocampus and parahippocampal gyrus, as a consistent predictor of conversion from MCI to AD.

In addition to many other evaluations, at each visit ADNI participants complete the

Alzheimer's Disease Assessment Scale - Cognitive Subscale, [56], or ADAS-Cog. A quantitative score is assigned to reflect a patient's cognitive function, derived from subtests pertaining to memory, praxis, and language; in our sample, scores range from 1 to 50, where higher scores indicate greater levels of cognitive dysfunction. Because AD is quite difficult to clinically diagnose, but ADAS-Cog takes only 30 minutes to administer, we utilize scores from ADAS-Cog as a surrogate measure for the Alzheimer's disease severity. Further, we consider ADAS-Cog instead of the popular Mini Mental State Exam (MMSE) because results of previous studies suggest these tests measure different aspects of AD-induced changes, where ADAS-Cog is more reflective of structural changes in the AAL-defined ROIs [8].

Methods

Our analysis goal is to build an interpretable linear regression model that predicts patient ADAS-Cog scores, serving as quantitative measures of disease severity, using regional VBM data derived from the AAL template ($p = 116$ regions). In particular, we want our model to be useful in the sense that it provides good predictions, and informative in that it is parsimonious and clinically interpretable. We analyze the ADNI dataset using constrained estimation methods, including Ridge regression, Lasso, and the Flexible Dantzig Selector.

Initially, we conduct our analysis by first partitioning the data into a training set (50%), a tuning set (25%), and a testing set (25%). Each constrained fitting method is applied to the training set to obtain coefficient estimates for a sequence of tuning parameter values. These estimates are used to make predictions in the tuning set, and the coefficient estimate that minimizes prediction error is selected. Finally, we evaluate the performance of the coefficient estimates for predicting patient ADAS-Cog scores in the test dataset.

We found our initial results unsatisfying, but not because of the story they told. As illustrated in Table 2.6 for a small subset of ROIs, coefficient estimates and the set of selected predictors vary considerably across bootstrap replicates of our data. Neuroimaging data is known to be noisy, and we certainly do not expect regional VBM data to be incredibly predictive of Alzheimer's disease, so it is not incredibly surprising that our results depend so heavily on the particular partitioning of observations; the model we are trying to identify likely has a very small R^2. This fact, combined with having no standard error estimates, is enough to make any reader suspect of our results.

To enhance and substantiate our analysis, we consider a bootstrap-enhanced approach to estimation (adapted from Bunea et al. [10]. We set aside 30% of the data as a test dataset, which ensures that test cases do not contribute to coefficient estimates, and we draw $B = 250$ samples of size n (with replacement) from our original dataset (i.e., training dataset). For each bootstrap sample, a tuning dataset is constructed from the remaining observations that are not already included in the training or test datasets; for any given bootstrap sample, the training, tuning, and test datasets are disjoint.

We fit the following methods to each of the $B = 250$ samples: (1) Ridge regression, (2) Lasso, (3) Dantzig Selector, (4) Flexible Dantzig Selector, and (5) Adaptive Flexible Dantzig Selector. For the latter two methods, we define a flexible penalty that includes sparsity constraints and pairwise-fusion constraints on spatially-adjacent (i.e., boundary-sharing) ROIs. For each method, estimates are obtained along a sequence of 50 values for λ. As described previously (Methods: Bootstrap-Enhanced Estimation), final coefficient estimates are obtained by averaging across bootstrap samples for predictors that exceed the specified inclusion probability threshold, and each method's estimated model is evaluated based on its performance in the test dataset.

For the Adaptive FDS, bootstrap-enhanced estimation necessitates an alternate approach to adaptive weighting. OLS estimates are likely not accurate in this setting (see Simulation 1 in Chapter 1), so it would be unwise to use them for constructing weights. A popular alternative is to use Ridge regression estimates [35], but in our setting this will cause clear data-recursion issues – we will use the same tuning set for selecting both Ridge and Adaptive FDS estimates. As an alternative, we consider building our weights from the *entire* Ridge coefficient path; the initial estimate for each coefficient is obtained as an average across all 50 λ's. We found this to be a useful shortcut in practice, when the OLS estimate is not suitable; adaptive weights have a proportional influence, and the grouping effect of Ridge causes the penalty to influence coefficients equally, so weights are usually near-constant across values of λ.

Analysis

The results of our initial analysis suggest that there is a considerable amount of noise in the data, evidenced by the variability in coefficient estimates across four randomly-

generated data-partitions (Table 2.6). Most concerning is the variability in the selected variables; we cannot confidently interpret the resulting coefficient estimates. Although there is inconsistency across partitions, it is reassuring that the estimates are consistent across methods; when faced with same dataset, these methods tend to agree on the important predictors.

The results of our bootstrap-enhanced analysis illustrate that some predictors are consistently selected more often than others (Figure 2.12). Using a probability threshold of 0.50, we select 19, 17, 101, and 16 predictors for the Lasso, Dantzig Selector, Flexible DS, and Adaptive FDS, respectively; although our sign-consistency adjustment causes Ridge to assign selection probabilities less than one for some predictors, all 116 ROIs still exceed 0.50. A similar set of predictors appears to be selected across methods, although the Flexible DS has much larger probabilities (i.e., it selects larger models) and therefore many more exceed the threshold. Previously noted in our simulations, the Flexible DS seems to have difficulty setting the coefficients to exactly zero, but the Adaptive FDS does a good job of correcting this issue. Given that the Adaptive FDS tends to be more computationally expensive, a viable alternative to weighting may be to simply have method-specific thresholds that are a function of their average model size. And based on our results, there are a lot of predictors that fall just below the threshold – do we really feel comfortable saying a predictor is not important because it was only selected 45% of the time?

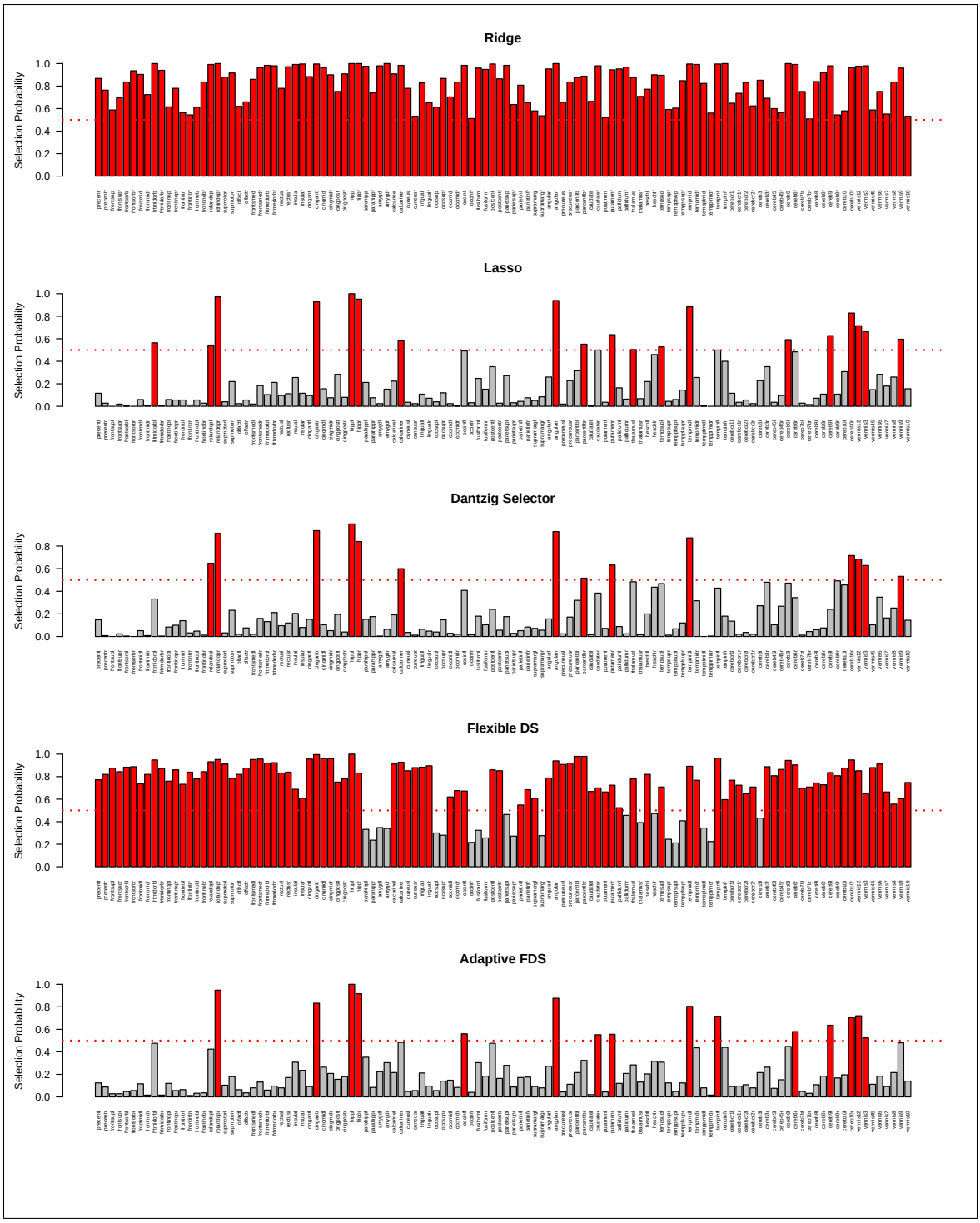

Figure 2.12: Probability of selection for each predictor ($p = 116$ AAL regions), calculated based on $B = 250$ bootstrap samples. Red bars correspond to predictors that exceed the threshold of 0.50, indicating that the predictor has been selected by the specified fitting method.

To further evaluate selection probabilities, we vary the significance threshold and determine the number of important predictors identified by each fitting method, for each threshold value (Figure 2.13). It is again clear that the Flexible Dantzig Selector selects many more predictors than all other methods (except Ridge), given a specific threshold. Lasso, Dantzig Selector, and Adaptive FDS have comparable model sizes across all threshold values, although the Adaptive FDS tends to choose slightly larger models for probability thresholds below 0.3. Notably, a small change in the selection probability threshold will cause a large change in the resulting predictors selected, regardless of fitting method; if we do not use 0.5, it is not clear what the threshold should be.

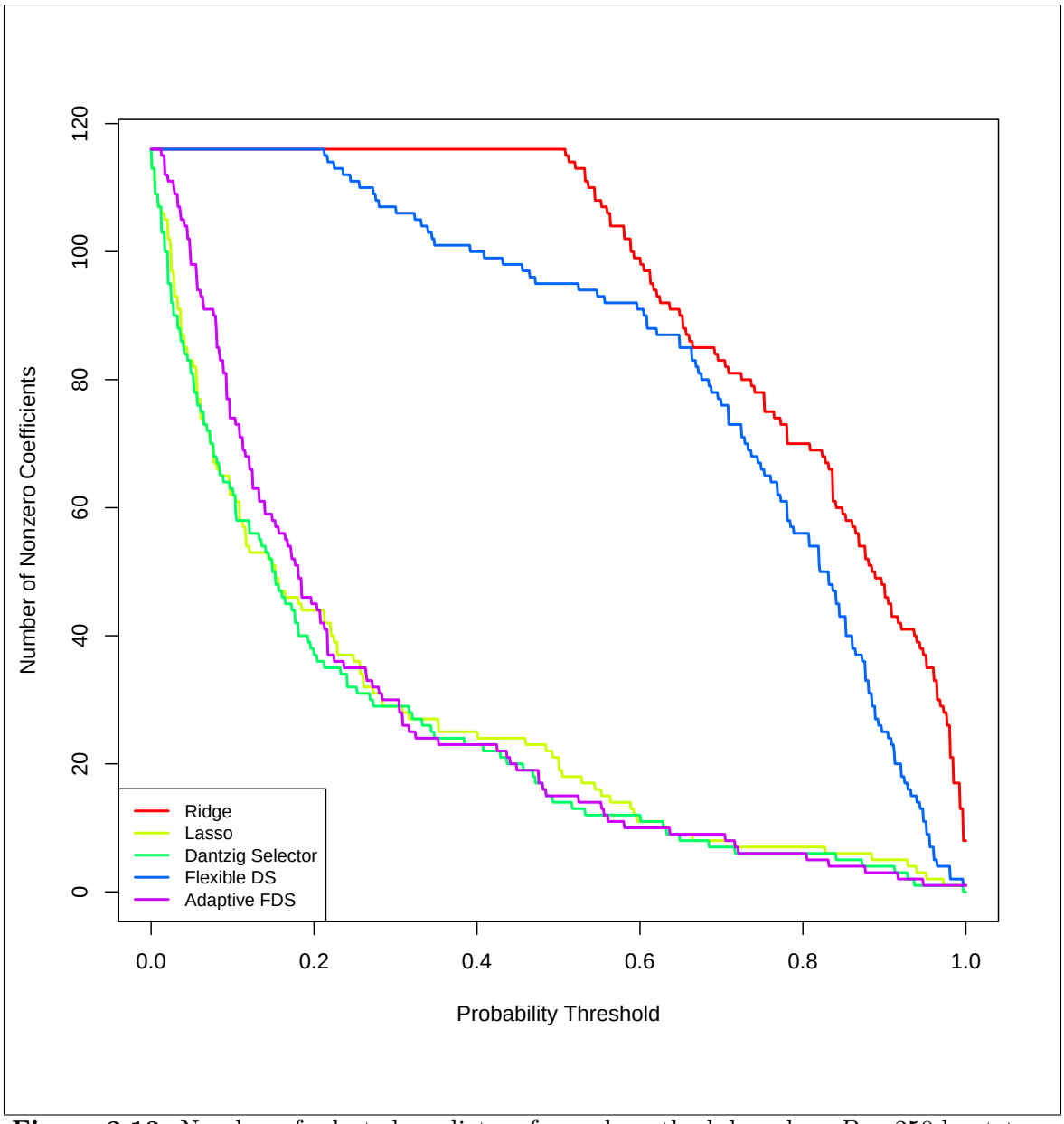

Figure 2.13: Number of selected predictors for each method, based on $B = 250$ bootstrap samples, across a range of values for the probability threshold for selection.

Shifting away from selection probabilities for the moment, we obtain bootstrap-enhanced estimates for each method by averaging coefficients across the $B = 250$ samples (Figure 2.14). We assume a probability threshold of zero, which yields non-sparse coefficient estimates for all methods. Among those considered, Ridge estimates tend to be the least attenuated. For coefficients that are distinctly nonzero, the Adaptive FDS tends to assign them a larger magnitude than other methods. Coefficients are generally in agreement for

Lasso, Dantzig Selector, and Flexible DS.

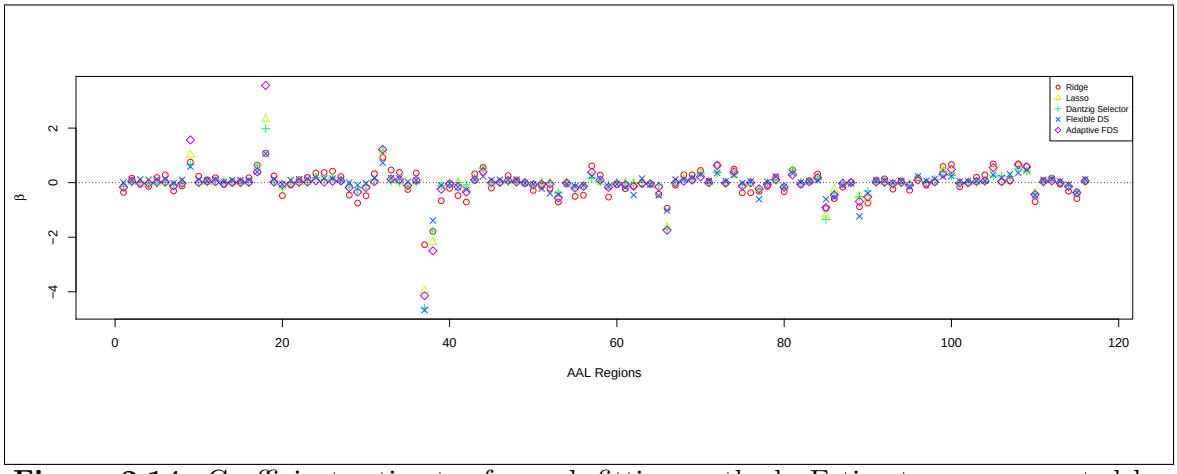

Figure 2.14: Coefficient estimates for each fitting method. Estimates are computed by averaging across $B = 250$ bootstrap samples. The dotted black line indicates $\beta = 0$, although by the nature of averaging (assuming a probability threshold of 0) we have mitigated sparsity in all fitting methods.

In terms of prediction, we found that the Adaptive FDS yields the most accurate predictions in the test set (Figure 2.15). This was true when considering models derived from each bootstrap sample individually (boxplots) and when considering the performance of the final bootstrap-enhanced estimates (red X's). Although the Adaptive FDS was consistently superior, all three version of the Dantzig Selector outperformed Ridge and Lasso.

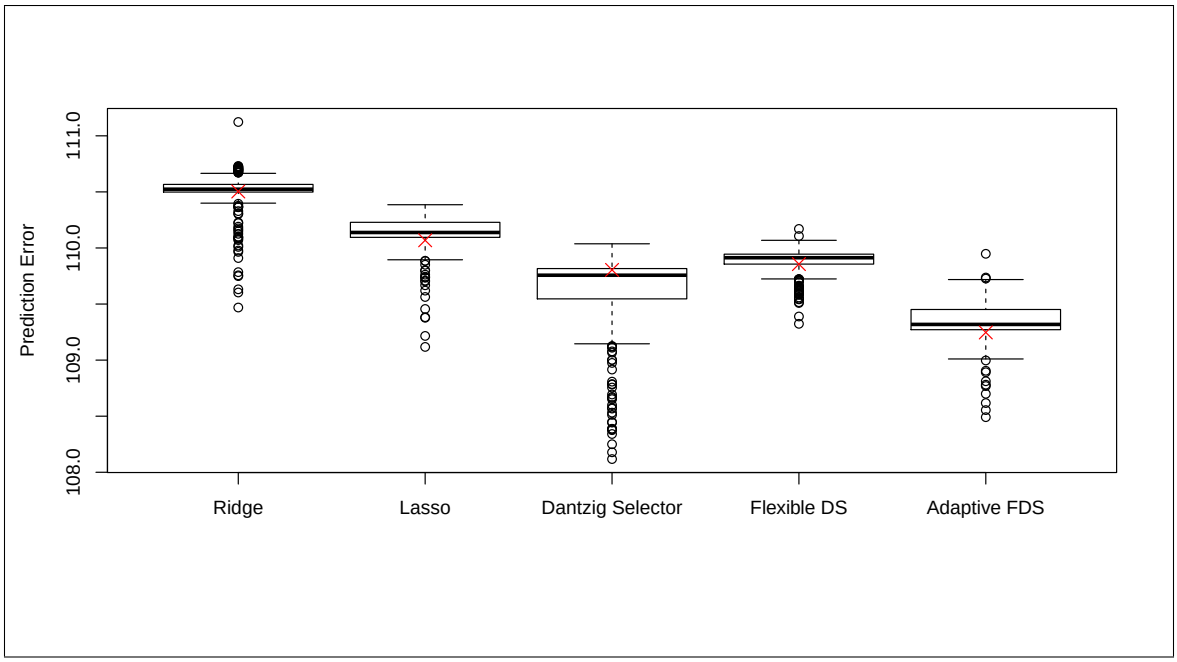

Figure 2.15: Prediction error in the test set for each fitting method. Boxplots represent the prediction errors when applying each of the $B = 250$ sets of coefficient estimates to the test set, and the red X indicates the prediction error for the bootstrap-enhanced estimates (i.e., average across samples).

We have established that the Adaptive FDS yields the most accurate model with regard to prediction in the test set. Does this have any impact on the size of the selected models? We already knew that the Flexible DS tends to include many more predictors than other methods, but the Adaptive FDS also appears to select marginally more predictors than Lasso and Dantzig Selector. This is not especially surprising, as we expect that incorporating spatial adjacency information in our resulting coefficient estimates will result in some degree of blurring (i.e., smoothing); a coefficient that was marginally significant previously may be magnified if its spatial neighbors are strongly associated with the outcome.

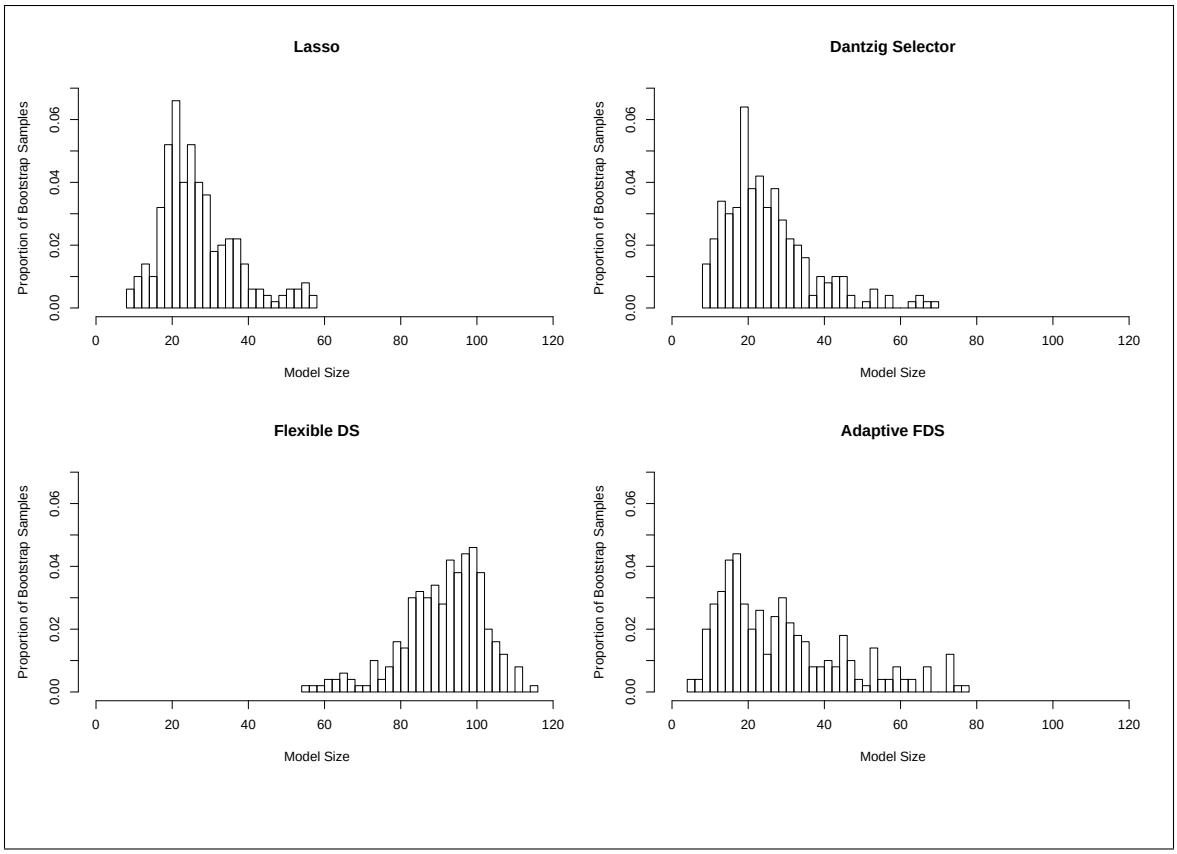

Figure 2.16: Number of nonzero coefficients (i.e., model size) across $B = 250$ bootstrap samples. Lasso and Dantzig selector are comparable, while Adaptive FDS selects slightly larger models a little more often. Flexible DS selects very large models relative to the other fitting methods, although many of the nonzero coefficients are very small. Ridge is excluded, because the model size is always $p = 116$ for any given bootstrap sample.

Before moving forward with selecting a final model and interpreting the results, we consider diagnostics to determine if we have drawn enough bootstrap samples to achieve convergence (Figure 2.17). Evaluated for both the coefficient estimates and the selection probabilities, it appears that convergence is achieved for the estimates after only about 50 bootstrap samples. Understandably more variable due to their discreteness, the selection probabilities seem to have mostly converged after $B = 250$ samples; although still a little jittery near the end, the variability is small and has a negligible impact on the selection of predictors. Another metric in support of convergence is prediction error in the test set (Figure 2.18). Thus, we have run enough iterations to confidently interpret the results.

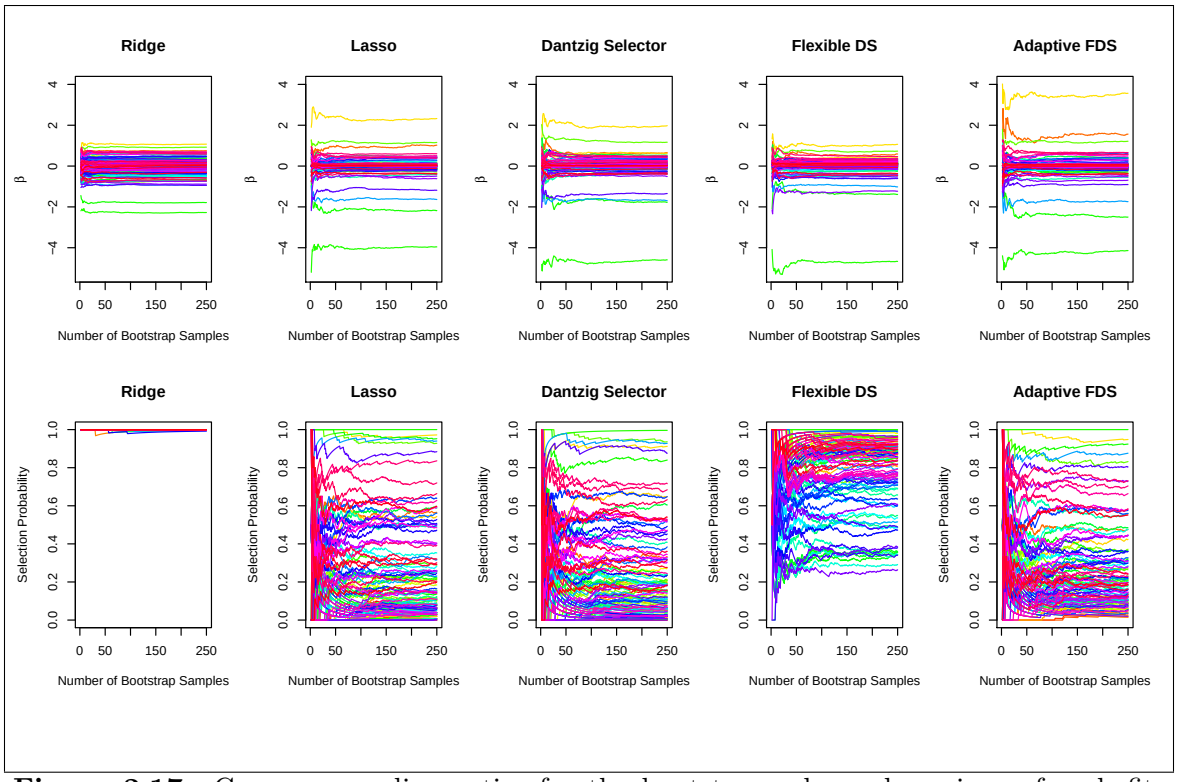

Figure 2.17: Convergence diagnostics for the bootstrap-enhanced versions of each fitting method. The top row shows the coefficient estimates versus the number of bootstrap samples, and the bottom row shows the selection probabilities. Although the selection probabilities are more variable, both seem to have converged after $B = 250$ iterations.

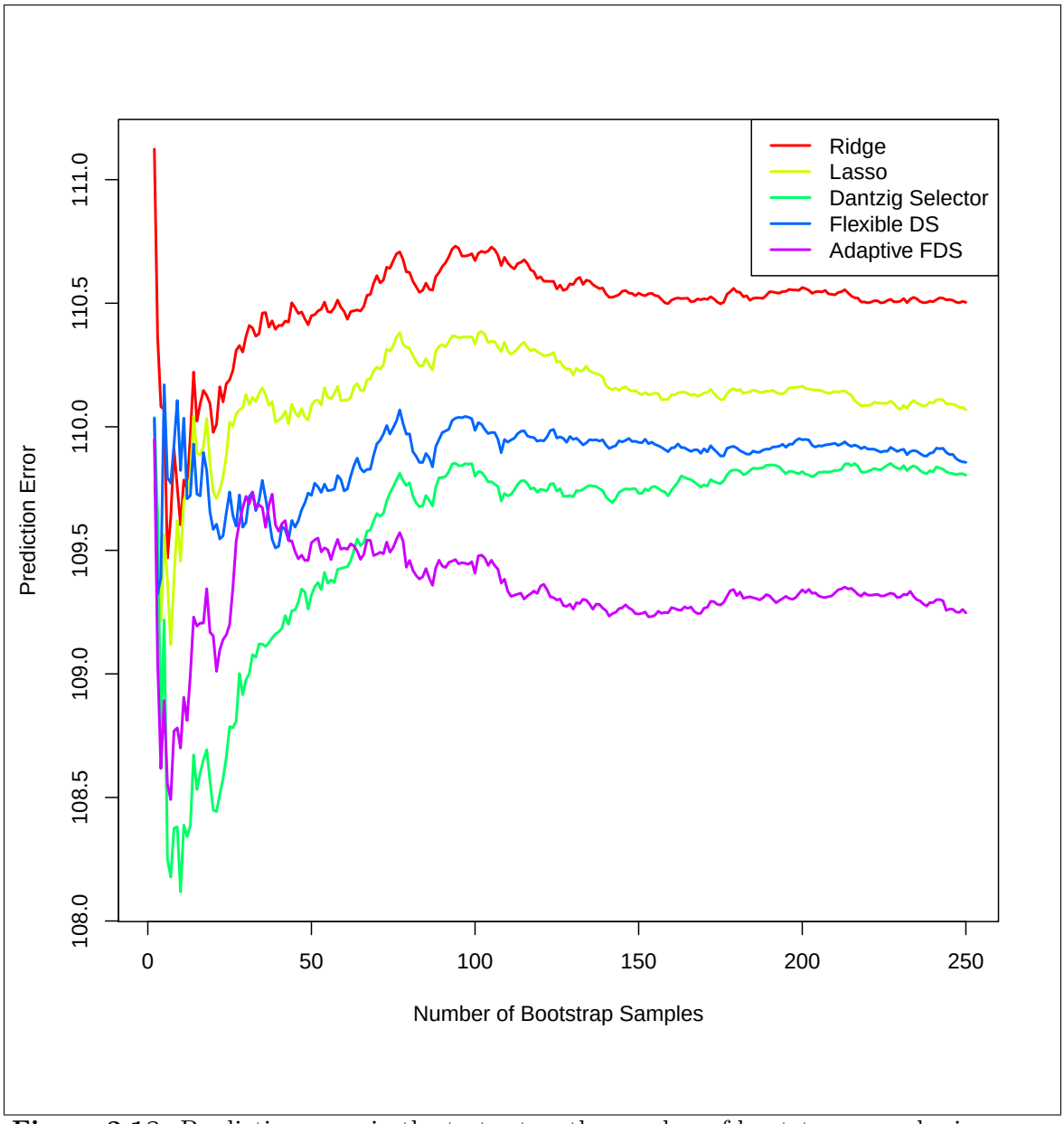

Figure 2.18: Prediction error in the test set as the number of bootstrap samples increase.

One unfortunate consequence of the bootstrap-enhanced approach is that the resulting coefficient estimates are not sparse; although individually sparse, averaging many sparse estimates yields a non-sparse solution (Figure 2.14). We impose sparsity by choosing a selection probability threshold and setting all coefficients corresponding to predictors not exceeding the threshold to exactly zero. Once again faced with the problem of selecting a probability threshold, we consider the performance of each bootstrap-enhanced estimate on the test dataset along a range of possible values for the selection probability threshold

(Figure 2.19, left). Because we are willing to consider a unique threshold for each method, it is more informative to consider prediction error as a function of the number of nonzero coefficient estimates (Figure 2.19, right). The Adaptive FDS has the lowest prediction among all methods, for almost all model sizes. In contrast, the Flexible DS performs poorly when it has a model with less than 80 predictors.

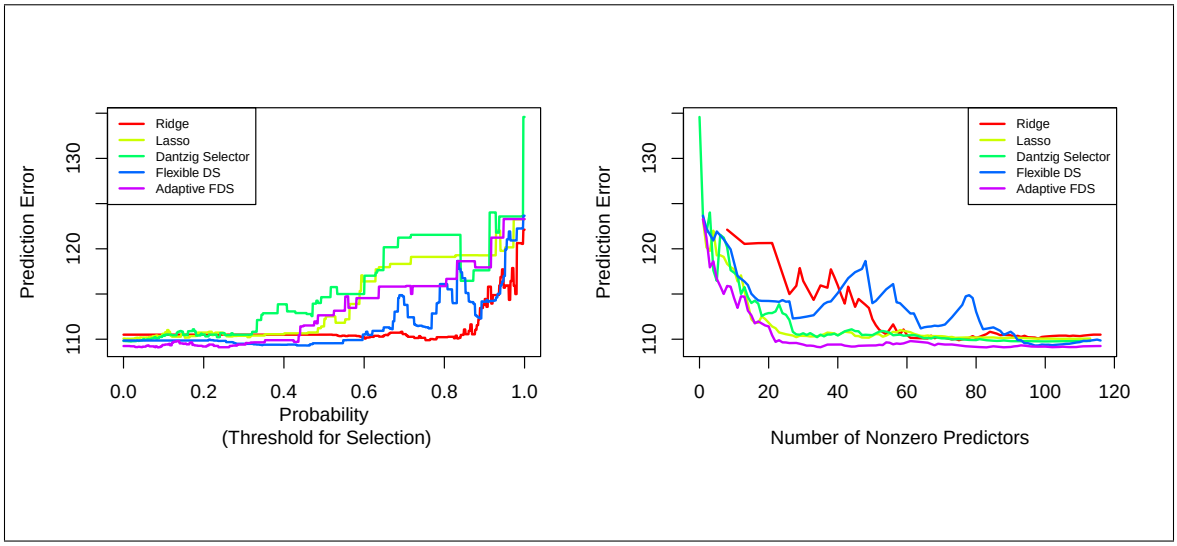

Figure 2.19: Prediction error in the test set as the level of sparsity in the coefficients changes, where sparsity is controlled by the selection probability threshold.

Although we can argue for a model between 20 and 40 predictors, the final decision is arbitrary. Selecting the model that minimizes prediction error is appealing, but will ensure our models perform optimally in the test set only; there is no guarantee that the chosen model-size will be optimal for future observations. Instead, we minimize a version of prediction error that includes a penalty for model-complexity. The complexity penalty is related to overfitting, where stronger penalties reduce the specificity of the model to our specific dataset; by imposing a penalty proportional to model size, we should ultimately improve the performance of our model for predicting future observations. Unfortunately, standard complexity penalties (e.g., AIC, BIC) are typically applied to the $-2\log L$, not the prediction error derived from an independent test dataset, so there is no theoretical basis for how strong the penalty should be for complexity. We consider a range of possible values for the complexity penalty multiplier, $\kappa \in [0, 2]$, where the complexity-penalized prediction

error is defined as

$$\mathrm{PE}(\kappa) = \mathrm{PE} + \kappa \left|\left|\hat{\boldsymbol{\beta}}\right|\right|_0$$

.

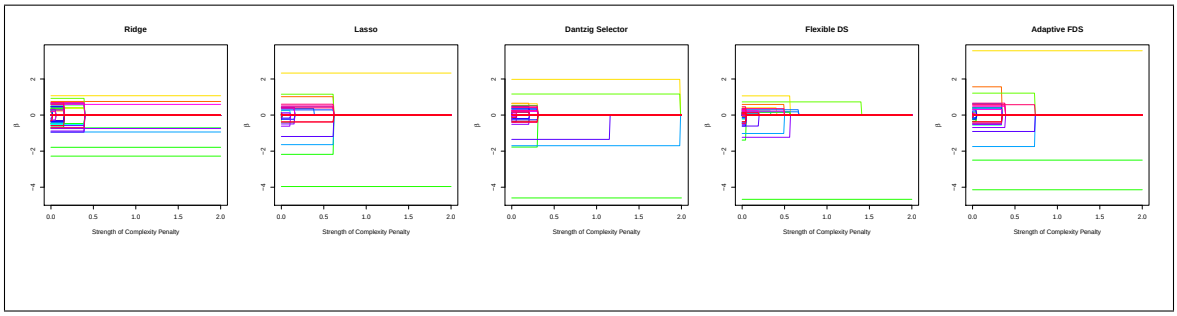

Figure 2.20: Coefficient estimates for each method, selected by minimizing $\mathrm{PE}(\kappa)$ in the independent test dataset, across a range of possible values for κ.

The final bootstrap-enhanced coefficient estimates for each fitting method are illustrated using a network (i.e., an undirected graph) that is superimposed on the brain, where each node represents VBM measures from an AAL region (Figure 2.21). Circular nodes denote AAL regions with VBM measures that are associated with ADAS score (i.e., selected predictors), while square nodes indicate ROIs not associated with ADAS score. Red nodes represent predictors that are positively associated with ADAS score, and blue nodes represent predictors with negative associations; color intensity reflects the magnitude of the coefficient estimate.

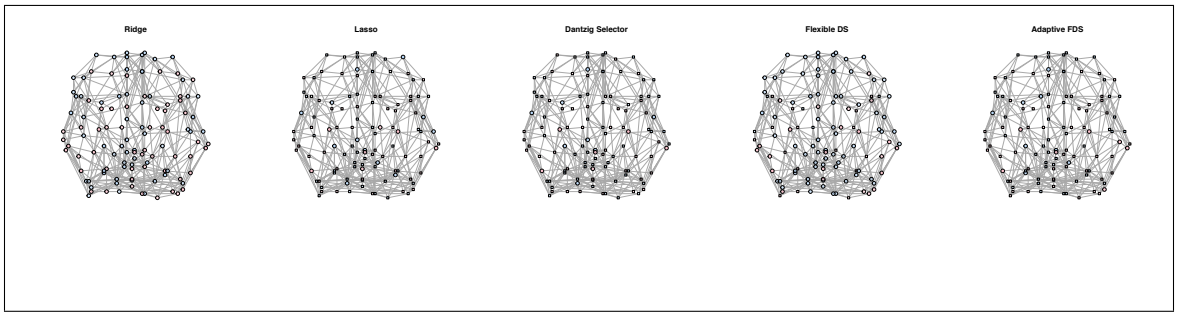

Figure 2.21: Network representation of the bootstrap-enhanced coefficient estimates for each fitting method. Circular and square nodes reflect predictors that exceed and do not exceed the selection threshold (0.5), respectively. Color reflects the sign of the coefficient estimates (red=positive, blue=negative), and transparency-level signifies estimate magnitude. Edges are constructed between all adjacent/neighbor ROIs.

2.5 Discussion

As we have illustrated, the Flexible Dantzig Selector can potentially produce regression models that outperform structureless competitors, assuming the true predictor structure is known. These improvements apply predominantly to prediction and estimation. As the number of structural connections (i.e., edges) increase, the Flexible DS encounters problems similar to the Lasso – its ability to select predictors deteriorates and model-size grows large, because many predictors are assigned very small, but nonzero, coefficient estimates. Further, caution should be used when applying the Flexible DS, because an incorrectly specified structure leads to poor performance.

Assuming we have an appropriate initial estimate, the Adaptive FDS provides significant improvements to the model-fit of the Flexible DS. Although we see improvements in prediction and estimation, the key feature of the adaptive weighting is the ability to push small coefficients to exactly zero; in general, the Adaptive FDS produces much smaller models than the Flexible DS. In addition, adaptive weighting is invaluable when we have incorrectly identified the predictor structure, as it serves to down-weight incorrectly-specified connections while emphasizing the sparsity/fusion constraints that are most consistent with the data. However, adaptive weighting does not construct *new* connections between variables; ignoring computational concerns, it may be wise to over-specify the underlying structure.

The Flexible DS relies on a known underlying variable structure, but this information may not always be available. One alternative is to assume the structure is defined by a complete graph (i.e., all predictors are connected), and then use adaptive weighting – essentially, a data-driven approach to defining the structure. Unfortunately, computation time for the Flexible DS increases with the number of constraints; it is not reasonable to fit the Adaptive FDS with all possible pairwise penalties. If we really want to go this route, one option is to define a 'connectivity threshold' and remove any constraints that, after applying the adaptive weights, do not exceed the threshold. However, the general criticism of this approach deals with double-dipping the data, as we are using the same data to both decide on the underlying structure and estimate the coefficients.

Another alternative is to use the correlations between predictors to determine structure

[54]. For example, we can deem predictors connected if their sample correlation exceeds a value of 0.5, and unconnected otherwise. As an alternative (or in addition) to adaptive weighting, we can use value the predictor correlations to directly to represent the strength of each structure connection. Although still technically double-dipping, the correlation-based approach at least ignores the response variable when constructing the variable structure.

One problem we encountered with the Adaptive FDS is that the OLS estimate, commonly used as the initial estimate for constructing weights, is incredibly unstable when p is large and predictors are correlated. It seems silly to think that weights derived from an unstable estimate will improve the model fit. Other initial estimates have been considered previously, including those derived from Ridge and Lasso fits. Lasso estimates are not appealing because some coefficients are exactly zero, leading to infinite weights; however, Ridge estimates are always nonzero, perform admirably in high-dimensional settings, and are reasonable when predictors are correlated. Unfortunately, both require an additional tuning step to obtain the estimate, which means we will be using the same tuning dataset to select the initial weights and to tune the Adaptive FDS. Instead, we propose using the average Ridge estimate across all values of the tuning parameter. We find this to be a reasonable approach because the impact of adaptive weights is related to the value of each weight relative to all other weights, as opposed to the absolute magnitude of the weight; although the Ridge coefficients shrink as the penalty increases, shrinkage applies to all coefficients and ultimately has a minimal impact on the relative magnitude of each coefficient at any given value of λ.

A drawback of estimates derived under constraints (sparsity or fusion) is the lack of an acceptable standard error estimate, and ultimately statistical inference. As any statistician knows, an estimate with a large magnitude may be insignificant if it has a comparably large standard error. We illustrated with the ADNI data that our solution was quite different across a few data perturbations, suggesting a reasonable amount of noise in the data. Instead of reporting these results, we considered a bootstrap-enhanced version to help evaluate variability, which yields bootstrap-enhanced coefficient estimates and selection probabilities for each predictor. Further, we propose an improvement to this approach in the form of a sign-consistency requirement. Specifically, instead of computing selection probabilities

from the proportion of times a predictor's coefficient is nonzero, we require the nonzero estimates to also be consistent in their relationship with the outcome. Essentially, selection probability is the maximum of two probabilities, one being the probability a predictor's coefficient is positive and the other negative. In addition to attenuating selection probabilities corresponding to direction-'flaky' predictors for any fitting method, the approach has the additional benefit of transforming the bootstrap-enhanced Ridge into a procedure for model selection; although all estimates are nonzero at every iteration, the selection probabilities will be less than one for predictors with sign-inconsistent coefficients.

Although the bootstrap-enhanced approach yields selection probabilities for each predictor, the resulting coefficient estimates will be non-sparse; averaging many individually sparse estimates will produce an estimate that is not sparse if a coefficient is assigned a nonzero value for at least one bootstrap sample. We impose sparsity by leveraging selection probabilities, assigning zero estimates to coefficients that do not exceed the threshold, and illustrate that this approach is effective for reducing prediction error in the test dataset. We use an ad-hoc complexity penalty to make a fair and comparable selection for the probability threshold; we want to select a model that does well in the test set, but a small complexity penalty helps reduce the risk of overfitting and should ultimately lead to improved external validity of our fitted model.

Presented using graphical networks, the results of the ADNI analysis illustrate the impact of incorporating a spatial-adjacency structure. When no structure is imposed (Lasso, Dantzig Selector) the resulting networks may have a negative coefficient that lies within a sea of positive coefficients, as no information from spatial-neighbors is utilized. Ridge solutions show more spatial structure, presumably a result of the grouping effect on correlated predictors, but are not sparse. Also not sparse, the Flexible DS solution seems to reflect the underlying graphical structure, most evidenced by the many negative coefficients near the top of the network. In fact, the solution for the Flexible DS may too strongly reflect the underlying structure, as many coefficients have changed sign relative to the solutions for other methods. Finally, the Adaptive FDS solution corresponds to the underlying network structure, and additionally imposes sparsity on the solutions.

In our analysis of the ADNI data, we consider imposing a structure on predictors that

are spatially adjacent. We believe this approach is reasonable because the data has been previously subject to spatial smoothing; an artifact of preprocessing, spatial neighbors are related. Further, it makes sense that ROIs in close proximity will have similar behavior. However, there is some evidence that geometric distance does not tell the entire story when it comes to the brain, as there are hard boundaries (i.e., sulci) that separate proximal regions and underlying structural connections facilitate communication between remote regions. If information regarding these connections is available (e.g., Diffusion Tensor Imaging (DTI), fMRI, etc), our model can easily incorporate this information multimodal information.

One direction for future work involves specification of the penalty matrix. Ideally, we would like to develop a procedure that automatically selects the structural connections to optimize model fit. One approach may be to parallel Stability Selection [48] and the Randomized Lasso [70], by considering a bootstrap-enhanced approach and randomly assigning penalty weights at each iteration. Alternatively, we can further investigate the results of the bootstrap-enhanced approach by evaluating the consistency of results for predictor-pairs, with and without a penalty. This would likely require many more iterations, and emphasizes the need for a more computationally efficient approach to optimization.

Another direction is to extend the Flexible DS to settings beyond the standard linear regression model, so we can apply similar penalty structures when faced with Binomial- or Poisson-distributed response variables. This will require an iterative approach to optimization that builds off the algorithm used for the Flexible DS, and may suffer from instability issues as the number of predictors and/or constraints grows large.

Table 2.4: Simulation Results: Goal 1

	Scenario 1 ($B = 1000$)			Scenario 2 ($B = 500$)		
	(a)	(b)	(c)	(a)	(b)	(c)
R^2	0.92	0.63	0.30	0.91	0.71	0.65
Prediction Error						
Oracle	**26.61**	**88.76**	**221.71**	**29.84**	**198.69**	**495.63**
Oracle OLS	**27.51**	**91.66**	**228.85**	**30.42**	**213.39**	**573.70**
Full OLS	28.44	94.88	236.55	41.92	281.66	704.60
Ridge	28.53	92.57	**225.04**	36.97	210.05	**508.30**
Lasso	28.04	92.88	226.50	**31.90**	215.35	523.41
Dantzig Selector	28.08	92.86	226.49	32.14	218.24	528.99
Flexible DS (correct)	**27.39**	**90.95**	225.48	32.32	**209.91**	510.51
Flexible DS (random)	28.35	94.10	226.03	33.20	214.81	515.63
Estimation Error						
Oracle	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Oracle OLS	**0.79**	**3.15**	**12.12**	**0.72**	**10.38**	**39.94**
Full OLS	1.20	5.03	19.30	4.19	28.02	70.71
Ridge	1.21	3.79	**6.68**	3.09	**8.85**	**12.54**
Lasso	1.02	3.90	9.04	**1.49**	11.02	21.78
Dantzig Selector	1.02	3.92	9.02	1.58	12.10	23.88
Flexible DS (correct)	**0.67**	**2.57**	7.11	1.67	**8.87**	14.82
Flexible DS (random)	1.16	4.59	8.64	2.01	10.93	18.40
Model Selection Error						
Oracle	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Oracle OLS	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Full OLS	5.00	5.00	5.00	46.00	37.00	25.00
Ridge	5.00	5.00	5.00	46.00	37.00	25.00
Lasso	3.11	3.25	4.31	**10.07**	**15.40**	**21.86**
Dantzig Selector	3.13	3.23	4.32	10.25	16.33	22.28
Flexible DS (correct)	**2.12**	**2.24**	**3.09**	27.25	28.95	22.15
Flexible DS (random)	4.71	4.74	4.92	29.21	33.50	24.27
Computation Time (minutes/path)						
Oracle	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Oracle OLS	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Full OLS	0.00	0.00	0.00	0.00	0.00	0.00
Ridge	0.01	0.01	0.01	0.01	0.01	0.01
Lasso	0.01	0.01	0.01	0.01	0.01	0.01
Dantzig Selector	0.20	0.17	0.16	1.10	1.12	1.13
Flexible DS (correct)	0.24	0.20	0.19	2.83	2.84	2.87
Flexible DS (random)	0.26	0.22	0.21	2.67	2.67	2.70

Table 2.5: Simulation Results: Goal 2

	Scenario 1 ($B = 1000$)			Scenario 2 ($B = 500$)		
	(a)	(b)	(c)	(a)	(b)	(c)
R^2	0.92	0.63	0.30	0.91	0.71	0.65
Prediction Error						
Oracle	**26.61**	**88.76**	**221.71**	**29.84**	**198.69**	**495.63**
Oracle OLS	**27.51**	**91.66**	**228.85**	**30.42**	**213.39**	**573.70**
Ridge	28.53	**92.57**	**225.04**	36.97	**210.05**	**508.30**
Dantzig Selector	28.08	92.86	226.49	32.14	218.24	528.99
Flexible DS (correct)	**27.39**	**90.95**	**225.48**	32.32	**209.91**	**510.51**
Flexible DS (random)	28.35	94.10	226.03	33.20	214.81	515.63
Adaptive Dantzig Selector	28.13	94.36	228.57	31.45	232.23	552.03
Adaptive Flexible DS (correct)	**27.30**	92.81	228.53	**30.30**	226.12	545.85
Adaptive Flexible DS (random)	28.23	94.49	228.16	**31.35**	230.37	550.80
Estimation Error						
Oracle	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Oracle OLS	**0.79**	**3.15**	**12.12**	**0.72**	**10.38**	**39.94**
Ridge	1.21	3.79	**6.68**	3.09	**8.85**	**12.54**
Dantzig Selector	1.02	3.92	9.02	1.58	12.10	23.88
Flexible DS (correct)	**0.67**	**2.57**	**7.11**	1.67	**8.87**	**14.82**
Flexible DS (random)	1.16	4.59	8.64	2.01	10.93	18.40
Adaptive Dantzig Selector	1.07	4.68	11.27	1.29	16.22	32.41
Adaptive Flexible DS (correct)	**0.62**	**3.68**	11.26	**0.38**	14.37	30.99
Adaptive Flexible DS (random)	1.11	4.77	10.94	**1.18**	15.90	32.99
Model Selection Error						
Oracle	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Oracle OLS	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Ridge	5.00	5.00	5.00	46.00	37.00	25.00
Dantzig Selector	3.13	3.23	**4.32**	10.25	16.33	**22.28**
Flexible DS (correct)	**2.12**	**2.24**	**3.09**	27.25	28.95	**22.15**
Flexible DS (random)	4.71	4.74	4.92	29.21	33.50	24.27
Adaptive Dantzig Selector	2.56	3.57	4.56	**0.68**	**15.49**	23.64
Adaptive Flexible DS (correct)	**0.78**	**2.66**	4.66	**0.57**	**14.94**	23.02
Adaptive Flexible DS (random)	2.73	3.64	4.61	1.05	16.19	23.95
Computation Time (minutes/path)						
Oracle	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Oracle OLS	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Ridge	0.01	0.01	0.01	0.01	0.01	0.01
Dantzig Selector	0.20	0.17	0.16	1.10	1.12	1.13
Flexible DS (correct)	0.24	0.20	0.19	2.83	2.84	2.87
Flexible DS (random)	0.26	0.22	0.21	2.67	2.67	2.70
Adaptive Dantzig Selector	0.20	0.17	0.16	1.14	1.18	1.19
Adaptive Flexible DS (correct)	0.24	0.20	0.19	2.65	2.68	2.70
Adaptive Flexible DS (random)	0.26	0.22	0.21	2.52	2.54	2.56

Table 2.6: ADNI Analysis – Subset of Coefficient Estimates

Method	Partition	cereb3l	heschlr	thalamusl	hippl	vermis45	parcentlbl
Lasso	1	-0.27	0.00	-0.39	-3.75	0.00	0.37
	2	0.00	0.00	0.00	-4.58	-0.02	0.00
	3	0.54	-0.48	0.00	-3.65	0.00	0.00
	4	0.00	-0.47	0.00	-4.75	0.00	1.44
	5	0.34	0.15	-0.01	-3.61	-0.11	0.49
Dantzig Selector	1	-0.30	0.00	-0.07	-4.58	0.00	0.36
	2	0.00	0.00	0.00	-4.16	0.00	0.00
	3	0.00	-0.15	0.00	-4.15	0.00	0.00
	4	0.00	-0.61	0.00	-5.57	0.00	1.38
	5	0.00	0.00	0.00	-4.52	0.00	0.00

Chapter 3

Conclusion

We presented a detailed review of traditional and modern approaches to estimation and model selection in the linear regression setting. Following a overview of the ordinary least squares approach, we shifted towards approaches that estimate the regression coefficients under various constraints. We provided a comprehensive review of these techniques, including motivation for their development, theoretical underpinnings, and computation details. To help obviate the drawbacks and benefits to employing these approaches in practice, we devised a multitude of simulation settings and provided detailed numerical results and a discussion. Next, we presented a flexible and intuitive approach for constraining regression coefficients based on an undirected graph representation of their underlying structure. To account for uncertainty in the underlying variable structure, we propose various strategies for weighting the constraints. We further provide resampling strategies to stabilize coefficient estimates and propose a sign-consistency requirement on the bootstrap-enhanced approach to improve model selection accuracy. Finally, we applied these methods to neuroimaging data from ADNI to identify brain regions associated with Alzheimers' disease progression under various structural constraints. Future work in this area will focus on obtaining a better understanding of how constraints influence estimates, studying when constraints can lead to estimation problems, developing a better strategy for adaptively weighting constraints, and extending the flexible penalty structure beyond the linear regression setting.

Bibliography

[1] Hirotugu Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.

[2] Carlos M Alaz, lvaro Barbero, and Jos R Dorronsoro. *Group fused lasso*, pages 66–73. Springer, 2013.

[3] Anestis Antoniadis, Piotr Fryzlewicz, and Frdrique Letu. The dantzig selector in cox's proportional hazards model. *Scandinavian Journal of Statistics*, 37(4):531–552, 2010.

[4] Taylor B Arnold and Ryan J Tibshirani. Efficient implementations of the generalized lasso dual path algorithm. *Journal of Computational and Graphical Statistics*, (just-accepted), 2015.

[5] John Ashburner and Karl J Friston. Voxel-based morphometrythe methods. *Neuroimage*, 11(6):805–821, 2000.

[6] Muhammad Salman Asif. *Primal dual pursuit a homotopy based algorithm for the dantzig selector*. Thesis, 2008.

[7] F.R. Bach. Consistency of the group lasso and multiple kernel learning. *The Journal of Machine Learning Research*, 9:1179–1225, 2008.

[8] Leslie C Baxter, D Larry Sparks, Sterling C Johnson, Brian Lenoski, Jean E Lopez, Donald J Connor, and Marwan N Sabbagh. Relationship of cognitive measures and gray and white matter in alzheimer's disease. *Journal of Alzheimer's disease: JAD*, 9(3):253–260, 2006.

[9] P.J. Bickel, Y. Ritov, and A.B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of statistics*, 37(4):1705–1732, 2009.

[10] Florentina Bunea, Yiyuan She, Hernando Ombao, Assawin Gongvatana, Kate Devlin, and Ronald Cohen. Penalized least squares regression methods and applications to neuroimaging. *NeuroImage*, 55(4):1519–1527, 2011.

[11] P. Bhlmann and L. Meier. Discussion: One-step sparse estimates in nonconcave penalized likelihood models. *The Annals of statistics*, 36(4):1534–1541, 2008.

[12] P. Bhlmann and S. Van De Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer-Verlag New York Inc, 2011.

[13] E.J. Candes, M.B. Wakin, and S.P. Boyd. Enhancing sparsity by reweighted 1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, 2008.

[14] Emmanuel Candes and Terence Tao. The dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, pages 2313–2351, 2007.

[15] Arnak Dalalyan and Yin Chen. Fused sparsity and robust estimation for linear models with unknown variance. In *Advances in Neural Information Processing Systems*, pages 1259–1267.

[16] Lee Herbrandson Dicker. *Regularized regression methods for variable selection and estimation*. Thesis, 2010.

[17] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[18] MA Efroymson. Multiple regression analysis. *Mathematical methods for digital computers*, 1:191–203, 1960.

[19] Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911, 2008.

[20] Luiz K Ferreira, Breno S Diniz, Orestes V Forlenza, Geraldo F Busatto, and Marcus V Zanetti. Neurostructural predictors of alzheimer's disease: a meta-analysis of vbm studies. *Neurobiology of aging*, 32(10):1733–1741, 2011.

[21] J. Friedman, T. Hastie, H. Hfling, and R. Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[22] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[23] J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and a sparse group lasso. *Arxiv preprint arXiv:1001.0736*, 2010.

[24] Christopher R Genovese, Nicole A Lazar, and Thomas Nichols. Thresholding of statistical maps in functional neuroimaging using the false discovery rate. *Neuroimage*, 15(4):870–878, 2002.

[25] Michael Grant, Stephen Boyd, and Yinyu Ye. Cvx: Matlab software for disciplined convex programming, 2008.

[26] Michael C Grant and Stephen P Boyd. *Graph implementations for nonsmooth convex programs*, pages 95–110. Springer, 2008.

[27] E. Greenshtein and Y.A. Ritov. Persistence in high-dimensional linear predictor selection and the virtue of overparametrization. *Bernoulli*, 10(6):971–988, 2004.

[28] Trevor J.. Hastie, Robert John Tibshirani, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009.

[29] Yoko Hirata, Hiroshi Matsuda, Kiyotaka Nemoto, Takashi Ohnishi, Kentaro Hirao, Fumio Yamashita, Takashi Asada, Satoshi Iwabuchi, and Hirotsugu Samejima. Voxel-based morphometry to discriminate early alzheimer's disease from controls. *Neuroscience letters*, 382(3):269–274, 2005.

[30] H. Hoefling. A path algorithm for the fused lasso signal approximator. *Journal of computational and graphical statistics*, 19(4):984–1006, 2010.

[31] A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, pages 55–67, 1970.

[32] J. Huang, S. Ma, and C.H. Zhang. Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, 18(4):1603, 2010.

[33] Clifford M Hurvich and ChihLing Tsai. A corrected akaike information criterion for vector autoregressive model selection. *Journal of time series analysis*, 14(3):271–279, 1993.

[34] L. Jacob, G. Obozinski, and J.P. Vert. Group lasso with overlap and graph lasso. pages 433–440. ACM, 2009.

[35] Gareth M James and Peter Radchenko. A generalized dantzig selector with shrinkage tuning. *Biometrika*, 96(2):323–337, 2009.

[36] Gareth M James, Peter Radchenko, and Jinchi Lv. Dasso: connections between the dantzig selector and lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(1):127–142, 2009.

[37] J. Jia and B. Yu. On model selection consistency of the elastic net when p¿¿ n. Report, DTIC Document, 2008.

[38] I.T. Jolliffe, N.T. Trendafilov, and M. Uddin. A modified principal component technique based on the lasso. *Journal of computational and graphical statistics*, 12(3):531–547, 2003.

[39] S Kinkingnehun, M Sarazin, S Lehericy, E Guichart-Gomez, T Hergueta, and B Dubois. Vbm anticipates the rate of progression of alzheimer disease a 3-year longitudinal study. *Neurology*, 70(23):2201–2211, 2008.

[40] K. Knight and W. Fu. Asymptotics for lasso-type estimators. *Annals of Statistics*, pages 1356–1378, 2000.

[41] C. Leng, Y. Lin, and G. Wahba. A note on the lasso and related procedures in model selection. *Statistica Sinica*, 16(4):1273, 2006.

[42] Dirk T Leube, Susanne Weis, Kathrin Freymann, Michael Erb, Frank Jessen, Reinhard Heun, Wolfgang Grodd, and Tilo T Kircher. Neural correlates of verbal episodic memory in patients with mci and alzheimer's diseasea vbm study. *International journal of geriatric psychiatry*, 23(11):1114–1118, 2008.

[43] Yi Li, Lee Dicker, and Sihai Dave Zhao. The dantzig selector for censored linear regression models. *Statistica Sinica*, 24(1):251, 2014.

[44] Han Liu, Jian Zhang, Xiaoye Jiang, and Jun Liu. The group dantzig selector. In *International Conference on Artificial Intelligence and Statistics*, pages 461–468.

[45] Ji Liu, Peter Wonka, and Jieping Ye. Multi-stage dantzig selector. In *Advances in Neural Information Processing Systems*, pages 1450–1458, 2010.

[46] Colin L Mallows. Some comments on c p. *Technometrics*, 15(4):661–675, 1973.

[47] N. Meinshausen and P. Bhlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of statistics*, 34(3):1436–1462, 2006.

[48] N. Meinshausen and P. Bhlmann. Stability selection. *Arxiv preprint arXiv:0809.2932*, 2008.

[49] Nicolai Meinshausen. Relaxed lasso. *Computational Statistics & Data Analysis*, 52(1):374–393, 2007.

[50] Nicolai Meinshausen, Guilherme Rocha, and Bin Yu. Discussion: A tale of three cousins: Lasso, l2boosting and dantzig. *The Annals of Statistics*, pages 2373–2384, 2007.

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

[52] G. Obozinski, M.J. Wainwright, and M.I. Jordan. Support union recovery in high-dimensional multivariate regression. *The Annals of statistics*, 39(1):1–47, 2011.

[53] M.Y. Park and T. Hastie. L1regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):659–677, 2007.

[54] S. Petry, C. Flexeder, and G. Tutz. Pairwise fused lasso. 2011.

[55] A. Rinaldo. Properties and refinements of the fused lasso. *The Annals of statistics*, 37(5B):2922–2952, 2009.

[56] Wilma G Rosen, Richard C Mohs, and Kenneth L Davis. A new rating scale for alzheimer's disease. *The American journal of psychiatry*, 1984.

[57] V. Roth and B. Fischer. The group-lasso for generalized linear models: uniqueness of solutions and efficient algorithms. volume 104. Citeseer, 2008.

[58] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[59] Charles M Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981.

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

[61] R. Tibshirani. The lasso method for variable selection in the cox model. *Statistics in medicine*, 16(4):385–395, 1997.

[62] R. Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):273–282, 2011.

[63] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[64] R.J. Tibshirani. The lasso problem and uniqueness. *Arxiv preprint arXiv:1206.0313*, 2012.

[65] R.J. Tibshirani and J. Taylor. The solution path of the generalized lasso. *The Annals of statistics*, 39(3):1335–1371, 2011.

[66] T Tony and Jinchi Lv Cai. Discussion: The dantzig selector: Statistical estimation when p is much larger than n. In *Ann. Statist.* Citeseer, 2007.

[67] N.T. Trendafilov and I.T. Jolliffe. Dalass: Variable selection in discriminant analysis via the lasso. *Computational Statistics & Data Analysis*, 51(8):3718–3736, 2007.

[68] Nathalie Tzourio-Mazoyer, Brigitte Landeau, Dimitri Papathanassiou, Fabrice Crivello, Olivier Etard, Nicolas Delcroix, Bernard Mazoyer, and Marc Joliot. Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain. *Neuroimage*, 15(1):273–289, 2002.

[69] H. Wang and C. Leng. A note on adaptive group lasso. *Computational Statistics & Data Analysis*, 52(12):5277–5286, 2008.

[70] Sijian Wang, Bin Nan, Saharon Rosset, and Ji Zhu. Random lasso. *The annals of applied statistics*, 5(1):468, 2011.

[71] Keith J Worsley, Jonathan E Taylor, Francesco Tomaiuolo, and Jason Lerch. Unified univariate and multivariate random field theory. *Neuroimage*, 23:S189–S195, 2004.

[72] Fei Ye and Cun-Hui Zhang. Rate minimaxity of the lasso and dantzig selector for the lq loss in lr balls. *The Journal of Machine Learning Research*, 11:3519–3540, 2010.

[73] Gui-Bo Ye and Xiaohui Xie. Split bregman method for large scale fused lasso. *Computational Statistics & Data Analysis*, 55(4):1552–1569, 2011.

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

[75] Jian Zhang, Xinge Jessie Jeng, and Han Liu. Some two-step procedures for variable selection in high-dimensional linear regression. *arXiv preprint arXiv:0810.1644*, 2008.

[76] P. Zhao and B. Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7(2):2541, 2007.

[77] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

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

[79] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.

[80] H. Zou and H.H. Zhang. On the adaptive elastic-net with a diverging number of parameters. *Annals of Statistics*, 37(4):1733, 2009.