Reweighting Methods in High Dimensional Regression

Zhou Fang

Corpus Christi College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2012

Acknowledgements

I would like to thankfully acknowledge the support of my parents,

the comments of many friends and colleagues,

the criticism of several journal reviewers,

the assistance of my funding authority, the EPSRC,

and most importantly, my supervisor, Professor Nicolai Meinshausen, without whose many hours of patient advice and suggestions this thesis could not have been written.

Abstract

In this thesis, we focus on the application of covariate reweighting with Lasso-style methods for regression in high dimensions, particularly where $p \geq n$. We apply a particular focus to the case of sparse regression under a-priori grouping structures.

In such problems, even in the linear case, accurate estimation is difficult. Various authors have suggested ideas such as the Group Lasso and the Sparse Group Lasso, based on convex penalties, or alternatively methods like the Group Bridge, which rely on convergence under repetition to some local minimum of a concave penalised likelihood. We propose in this thesis a methodology that uses concave penalties to inspire a procedure whereupon we compute weights from an initial estimate, and then do a single second reweighted Lasso. This procedure – the Co-adaptive Lasso – obtains excellent results in empirical experiments, and we present some theoretical prediction and estimation error bounds.

Further, several extensions and variants of the procedure are discussed and studied. In particular, we propose a Lasso style method of doing additive isotonic regression in high dimensions, the Liso algorithm, and enhance it using the Co-adaptive methodology. We also propose a method of producing rules based regression estimates for high dimensional non-parametric regression, that often outperforms the current leading method, the RuleFit algorithm. We also discuss extensions involving robust statistics applied to weight computation, repeating the algorithm, and online computation.

Contents

List of Figures

Chapter 1

Introduction

In a wide range of problems in statistics, traditional methods have obtained reasonable, and indeed in many cases provably optimal solutions. In particular, in the case of regression, a broad range of algorithms have emerged over the years, to handle a variety of situations. But many modern applications create new difficulties that such methods are ill adapted to handle.

Firstly, when the amount of data to be analysed is too large, many algorithms fail because they cannot scale - in both memory requirements and computational speed - to accommodate it. Even with increases in computational power, it may be many years before computations requiring weeks to perform currently, can be dealt with in more reasonable time scales. Simultaneously, the size of datasets used as input is in many cases itself increasing, with the advancement of data collection technology. In astronomical databases, for example, the size of databases have exploded from less than a terabyte accumulated over several decades, to multiple terabytes to store and process generated each night. The problem is compounded by the common desire to repeat computations, either to test for robustness, to perform bootstrap style analysis, or to choose tuning parameters.

Secondly, the solution space can become too large for traditional algorithms to determine the correct answer. In the case of linear regression, standard approaches such as ordinary least squares requires that the number of parameters, p, not exceed the sample size n. Otherwise, the sample covariance matrix would be necessarily non-positive definite, and computation impossible. In asymptotic scenarios, with p increasing even proportionally to n, estimates will fail to be consistent. More generally, concurvity is a threat in all most attempts to fit complex models – in

such situations, multiple candidate estimates provide similar results in sample space, making it difficult or impossible to distinguish between them.

Thirdly, prior assumptions can be too complex to encode and incorporate into methodology. A key case is variable sparsity - the idea that a fitted model should be parsimonious in that only a small number of variables have an effect on the response variable. One other example is the presence of group sparsity, which suggest combinations of the variables that might simultaneously be irrelevant, or might simultaneously influence the response variable.

The Lasso [Tibshirani 1996] has been recently proposed to address these problems in the specific case of linear regression under a sparsity assumption. The Lasso consists of optimisation of a cost function that is the sum of a residual sum of squares type likelihood component, and a L1 penalty term. The L1 penalty term produces sparse coefficient estimates, and more significantly, helps leverage undiscovered sparsity or near-sparsity to produce excellent results. Implementations of the Lasso have also been devised that allow it to be computed highly efficiently. Variants of the Lasso have been produced to tackle group sparsity [Yuan and Lin 2006], and also provide extensions to additive modelling [Ravikumar et al. 2007] and other situations. In addition, an important alteration to the Lasso method has been devised - the Adaptive Lasso [Zou 2006], which uses an initial estimation step to reweight computations to produce what are in many cases substantially improved results.

The main contribution of this thesis is to produce a generalisation of the Adaptive Lasso. We propose a broader set of methods that conduct an initial estimate - which we propose to do so with the Lasso - and then compute a second estimate in a certain way by integrating the first estimate in as covariate weights. Unlike existing methods, our novelty is that we allow more complex and flexible formulations of these weights - the chief of which being that we allow weights placed on each covariate to relate to estimates of other, different covariates. This therefore enables us to take into account more complicated prior assumptions. We also dispense with many authors' focus on repeating methods to convergence, instead working with simple two-step algorithms. In particular, we use such a methodology to define the Co-adaptive Lasso, a procedure that enables the computation of estimates that both respect group sparsity, and also retains the capability of providing and taking advantage of additional sparsity within the groups.

To support this methodology, we prove theoretically some positive properties of this procedure, and provide comparisons to similar procedures. We differ from many other authors in our approach, by considering the performance of the method itself,

instead of as a local optimum of some underlying criteria. We also detail the practical implementation of the method, and provide empirical evidence for the effectiveness of this approach in both artificial datasets and real ones. As part of this, we also identify a means of computing an online estimate of the Co-adaptive Lasso, which also enables us to implement a novel online version of the Adaptive Lasso.

In addition, we contribute applications of the 'Co-adaptive principle' to problems in non-linear regression. One example is isotonic regression in many dimensions, which provides our initial motivation. For this, we define the Liso estimator, and augment it with our reweighting process to produce the Adaptive Liso. We also examine tree based methods, and produce the Co-adaptive RuleFit. For these, we provide algorithms for efficient computation and some empirical experiments.

The structure of this thesis will be as the following:

We begin by reviewing the Lasso and its variants, in Chapter 2. We present notational conventions for this thesis, and then review the literature around the estimator itself, the Group Lasso, and the Adaptive Lasso, and the main methods of its implementation.

In Chapter 3, inspired by the Lasso, we introduce the Lasso Isotone (Liso) for additive isotone regression under a sparsity constraint. We construct an algorithm for the computation of this estimate, presenting a theorem on its convergence. We then present an empirical example of this method in practice.

In Chapter 4, we consider concave penalties and reweighting methods, using the Liso as a motivating example. We discuss the advantages of concave penalties over other methods, and the Lasso in particular, using some simple examples. We then discuss the implementation of such methods, and argue for approaching the multiple repeated Lasso methodology of the LLA as an estimator itself, rather than as a local optimum for the implied concave penalised criterion. We apply this reweighting insight to the Liso to produce the Adaptive Lasso, for which we present empirical comparisons with other methods.

In Chapter 5, we make this more rigorous and general, proposing the Co-adaptive Lasso as a method of using reweighted Lasso computations to make estimations in the presence of grouping structures and sparsity. We present a variety of theorems on this method's prediction error performance, with comparisons to similar methods. We also suggest an extension involving the use of winzorised weights, with a theorem suggesting its superiority, albeit under a fairly restrictive condition. We discuss group structures featuring overlapping groups, and weighting methodologies that might handle these well.

In Chapter 6, we further extend the Co-adaptive Lasso, covering some other problems and situations, while providing numerical experiments. We discuss computation and tuning parameter selection, and the possible usefulness of repeating the Co-adaptive Lasso calculation to obtain a better result. We also construct an online version of the method. We present empirical examples using both real and artificial datasets. Finally, we describe and experiment with the Co-adaptive RuleFit.

Finally in Chapter 7 we summarise the thesis and suggest some future work. Lengthy proofs to theorems in this thesis are left for the Appendix.

Chapter 2

The Lasso and its variants

Starting with Tibshirani [1996], a variety of authors have contributed to the design and analysis of the Lasso and its variants. In this chapter, we define some notation for use in the rest of this thesis, and review some of their work.

2.1 General notation

Let $Y \in \mathbb{R}^n$ be a response vector, and X be the matrix of covariates. We adopt a subscript notation, wherein $X_{i,\cdot}$ denotes the i-th row and $X_{\cdot,k}$ denotes the k-th column. Each observation i, for $i = 1 \dots n$ then is a pair Y_i, $X_{i,\cdot}$, with $X_{i,\cdot}$ a $1 \times p$ row vector.

$\|\cdot\|_n$ denotes the empirical L2 norm, so that for instance,

$$\|Y\|_n = \sqrt{\sum_{i=1}^{n} \frac{Y_i^2}{n}}.$$

Meanwhile, $\|\cdot\|_q$ denotes the Lq norm, so that

$$\|Y\|_q = \left(\sum |Y_i|^q\right)^{1/q}.$$

As a shorthand, we write $\|\cdot\| = \|\cdot\|_2$. Note that the L1 norm is $\|\cdot\|_1$ in this notation.

Note that throughout, for clarity, we use small caps Roman letters s to denote

scalar or vector quantities, capitals S to denote sets or matrices, and script letters $\mathcal{S}$ to denote sets of sets.

For any subset $G \subset \{1, \ldots, p\}$, we then denote by $X_{\cdot,G}$ – or for simplicity, if this is unambiguous, X_G – the columns of X corresponding to the indices in G. Similarly, for a vector v, say, we denote by v_G the terms of v corresponding to the indices in G.

Finally, we use bracketed values in subscripts to express order statistics, so that $X_{(i),k}$ would denote the i-th smallest value of $X_{\cdot,k}$.

2.2 The Lasso

Consider the linear model framework, under this notation, with an independent Normal error term ε:

$$Y = X\beta + \varepsilon$$

We are then interested in recovering $\beta \in \mathbb{R}^p$. Now, maximising likelihood corresponds to minimising the negative loglikelihood, which in this case is given by the residual sum of squares $\|Y - X\beta\|^2$. Adding a penalty then produces the general form of a penalised likelihood estimator:

$$\arg\min_{\beta} \frac{1}{2} \|Y - X\beta\|_n^2 + \sum_{k=1}^{p} P_\lambda(\beta_k). \tag{2.2.1}$$

The penalty modifies the results we obtain, in often beneficial ways. The question then is what form P_λ should take to give a good estimator.

The Lasso was first conceived of as a least squares calculation constrained in the L1 norm of the estimate [Tibshirani 1996]. That is, for some $\tau > 0$, the Lasso estimate is

$$\widehat{\beta}^{(l)} = \arg\min_{\beta} \|Y - X\beta\|_n^2, \quad \|\beta\|_1 \leq \tau.$$

More convenient for computation, however, is to incorporate the L1 constraint as a *penalty term*, governed by a Lagrange multiplier $\lambda \geq 0$ that acts as a tuning parameter. In other words, we calculate

$$\widehat{\beta}_\lambda^{(l)} = \arg\min_{\beta} \ell_\lambda^{\text{lasso}}(\beta) = \arg\min_{\beta} \frac{1}{2} \|Y - X\beta\|_n^2 + \lambda \|\beta\|_1. \tag{2.2.2}$$

This methodology is equivalent to the constraint formulation, since every τ in the first definition produces a solution that equals that the solution from a corresponding choice of λ in the second. Note that the estimate is a function of the choice of tuning parameter, though for simplicity we will usually suppress the subscript λ in $\widehat{\beta}_{\lambda}^{(l)}$ when we are fixing a particular value of λ.

Clearly, then, the Lasso is an example of a penalised likelihood estimator, setting $P_{\lambda}(\beta_k) = \lambda|\beta_k|$. Tibshirani [1996] also suggested viewing the Lasso as a maximum a-posteri estimate using a Laplacian prior. However, the Lasso does not fit well into this Bayesian framework, because we find that generally, contrary to the usual Bayesian practice, use of the MAP estimate performs better than use of the more standard posterior mean, and the Laplacian prior itself certainly does not mean we have a non-zero prior probability of coefficients having exactly zero values.

It is illustrative to compare the Lasso to the traditional approach of Ridge regression, which uses for $P_{\lambda}(\beta_k)$ instead the squared L2 norm $\lambda\beta_k^2$. This simple change produces unexpectedly dramatic differences in results. While the Ridge Regression penalty provides often desirable shrinkage properties, the Lasso does not shrink but instead biases estimates, often undesirably. While Ridge Regression will almost always produce a β estimate that is non-zero in every component, the Lasso will always select a maximum of n variables, and usually much fewer.

Beyond these differences, however, are several desirable properties, which have been described by many authors. Suppose that the β we are trying to estimate is sparse, in the sense that $\text{Supp}(\beta) \subset S$ for some small S.

Under this assumption, Zhao and Yu [2006] and Meinshausen and Bühlmann [2006] wrote of the importance of the Irrepresentable Condition,

$$\max |X' X_S (X_S' X_S)^{-1} \text{sign}\,(\beta_S)\,| < 1$$

in controlling whether the Lasso selects the correct variables. When this is satisfied, the Lasso can with high probability set the remaining variables $\beta_{S^c} = 0$, while estimating β_S with the correct sign.

In addition, Bickel et al. [2009] and later van de Geer and Bühlmann [2009], amongst others, have shown that a a variety of weaker conditions - the Restricted Eigenvalue condition, the Compatibility condition and so on, assure success of the Lasso in achieving good bounds on estimation and prediction error. In this case, the prediction error of the Lasso may be bounded to being within a $\log(p)$ factor of the 'oracle' rate, the rate achievable if S was assumed known.

Together, these establish the usefulness of the Lasso when the sparsity assumption is met. The Lasso has been applied to a range of problems, including knot selection for spline fitting [Osborne et al. 1998], fitting rule ensembles [Friedman and Popescu 2008] and classification via the Logistic Lasso [Wu et al. 2009]. The Fused Lasso [Tibshirani et al. 2005] and Elastic Net [Zou and Hastie 2005] are closely related variants. One penalises differences between coefficient estimates, while the other introduces an additional L2 penalty to produce shrinkage of large β estimates.

2.3 Group Lasso

In other situations, a group sparsity constraint might be more appropriate. Specifically, suppose we are endowed with an a-priori grouping structure $\mathcal{G}$, being a set of subsets of $\{1, \ldots, p\}$, which determines the patterns of variables we wish to simultaneously select, or set equal to zero. Yuan and Lin [2006] proposed a natural way to modify the Lasso penalty to incorporate this grouping structure, with a grouped L2 norm penalty, producing the Group Lasso:

$$\widehat{\beta}^{(l)} = \arg\min_{\beta} \frac{1}{2} \|Y - X\beta\|_n^2 + \lambda \sum_{G \in \mathcal{G}} \|\beta_G\| .$$

Variants of the Group Lasso can be arrived at by using other group norm penalties. For example, the broad set of penalties of the form

$$\sum_{G \in \mathbb{G}} \|\beta_G\|_q ,$$

where $\|\cdot\|_q$, for $q > 1$ denotes the Lq norm, are termed the Composite Absolute Penalties family. Such penalties are considered in Zhao et al. [2009], which also provides algorithms and proves that similar solutions may be obtained.

The composite L2 type penalty changes the Lasso penalty so that discontinuities in the differential of the penalty exist only for values of β where entire groups of coefficients β_G are zero. Thus, in the absence of overlapping groups[1], the Group Lasso gives solutions where either an entire group is zero, or the entire group is selected in that each component β_k, $k \in G$, are all non-zero.

The Group Lasso is known to have better theoretical results than the Lasso when the grouping structure is sufficiently informative of the coefficient structure.

[1] The behaviour in the case of overlapping groups is more complex, and is discussed in Section 5.5.

Huang and Zhang [2010], for example, formulated the notion of 'Strong Group Sparsity', wherein for a given sparsity pattern, there exists a set of groups such that $S = \text{Supp}(\beta)$ is contained within their union, the number of groups is small, and the covering is efficient such that if $\mathcal{G}_S$ covers S, $|\bigcup \mathcal{G}_S|/|S|$ is small. Under such an assumption, the authors show that the Group Lasso can obtain better performance bounds and require weaker conditions than the Lasso. The authors argue that if the condition fails, the Group Lasso fails because a bias term on the order of $|\bigcup \mathcal{G}_S|$ may be introduced. Heuristically, this can be understood in terms of the Group Lasso being effectively a weighted ridge regression estimate, applied to the set of selected variables. If the groups are an inefficient cover, a large number of variables must be selected, and this ensures poor performance. Bach [2008] also proved consistent estimation results for the Group Lasso, using an analogue of the Irrepresentable Condition.

The Group Lasso has seen also wide application, including in classification of splice sites [Meier et al. 2008], and additive modelling [Ravikumar et al. 2007].

2.4 Adaptive Lasso

In many situations, the bias problem of the Lasso can be handled by a procedure termed the Adaptive Lasso [Zou 2006]. The Adaptive Lasso functions by adjusting the Lasso fitting criteria, so as to penalise different covariates by different amounts. In contrast to the definition used in equation (2.2.2), we define, for $\gamma > 0$:

$$\widehat{\beta}^{(a)} = \arg\min_{\beta} \frac{1}{2} \|Y - X\beta\|_n^2 + \lambda \sum_{k=1}^{p} \frac{|\beta_k|}{|\widehat{\beta}_k^0|^\gamma}. \tag{2.4.1}$$

Here, $\widehat{\beta}^0$ is an initial estimate. In the original Adaptive Lasso paper [Zou 2006], it is suggested that this be a root-n-consistent estimator, and in particular the ordinary least squares estimate. Later authors proposed the Iterated Lasso [Huang et al. 2008] using the Lasso itself as the initial estimator, and proved results for this. The use of the Lasso for the initial estimator is important, because it means the process can be applied in the case where $p > n$, where traditional estimates like the ordinary least squares solution cannot be calculated. Generally, $\gamma = 1$ suffices, and for simplicity in this thesis we will use the 'Adaptive Lasso' to refer to this formulation of conducting an initial Lasso, and then computing a second Lasso as per (2.4.1) with γ fixed as 1.

The Adaptive Lasso is related to a framework defined in Zou and Li [2008]. In this paper, the authors begin by proposing a potentially non-convex penalty function

$\sum P_\lambda(\beta_k)$, in place of the L1 penalty $\sum |\beta_k|$ used in the Lasso. Then, this may be approximated by a sequence of Lasso computations, each adjusted with weights. Under this framework, consider the following penalised maximum likelihood estimate:

$$\arg\min_{\beta} \frac{1}{2} \|Y - X\beta\|_n^2 + \sum_{k=1}^{p} \lambda \log(|\beta_k| + \delta),$$

then, taking δ down to zero, the LLA procedure produces the Adaptive Lasso. We will discuss further this method in Section 4.2.

Several authors have analysed the properties of the Adaptive Lasso. In particular, van de Geer et al. [2010] described performance bounds in terms of prediction and estimation error, and the number of false positives, and found that the Adaptive Lasso often exhibits superior performance to the Lasso. An adaptive version of the Group Lasso exists – the Adaptive Group Lasso [Wang and Leng 2008], though analyses of the type in van de Geer et al. [2010] have not been conducted.

2.5 Implementations

The major advantage of Lasso type methods is their computational simplicity. Initially, as proposed in Tibshirani [1996], generic interior point methods were used to compute the Lasso. Such methods were permissible because of the inherent convexity of Lasso penalised loss functions. But they were generally slow, and hence the method did not appear very useful.

This changed with the development of efficient 'homotopy' algorithms, by Osborne et al. [2000] and Efron et al. [2004]. These authors observed the piecewise linearity property of the Lasso fit - that is, it can be shown that between breakpoints where a variable is either added to or removed from the active set, the Lasso solution is a linear function of the tuning parameter λ. Hence, it suffices to conduct the procedure in Algorithm 1, which generate the entire path of Lasso solutions for all values of λ after a finite number of steps.

The Homotopy method dramatically reduces the computational burden of the Lasso, often to on the order of that required for computing a least squares calculation with sample size n and $\min(n, p)$ covariates. However, in very high dimensional problems with large sample sizes, this can still be quite slow. Processing a full solution pass involves, after all, calculations on each breakpoint, of which there can be many. In many cases it is not necessary for the full solution path to be derived for good

Algorithm 1 Least Angle Regression/Homotopy Method

Require: $Y \in \mathbb{R}^n, X \in \mathbb{R}^{n \times p}$.

1: Initialise $\widehat{\beta}_{\lambda_0} = 0$, $\mathcal{A} = \{k\}$, where k maximises $|X'_{\cdot,k}Y|$. Set $\lambda_0 = |X'_{\cdot,k}Y|$, $m = 0$.
2: **repeat**
3: $m \leftarrow m + 1$.
4: Compute $\partial\widehat{\beta}_\lambda/\partial\lambda$ using $\widehat{\beta}_{\lambda_{m-1}}, \mathcal{A}$.
5: Compute δ_λ, the distance until the next break point.
6: Set $\lambda_m = \lambda_{m-1} - \delta_\lambda$, $\widehat{\beta}_{\lambda_m} = \widehat{\beta}_{\lambda_{m-1}} + \delta_\lambda(\partial\widehat{\beta}_\lambda/\partial\lambda)$.
7: Update $\mathcal{A}$.
8: **until** $\lambda_m = 0$
9: Interpolate linearly the $\widehat{\beta}_\lambda$ estimates.
10: **return** $\widehat{\beta}_\lambda$, for $\lambda \leq \lambda_0$.

results, especially if the optimal value of λ cannot be exactly determined anyway. In addition, Homotopy methods are not available for variants of the Lasso where the paths are not linear - for example the Group Lasso. For computation on a fixed grid of results, a separate 'Shooting', or Coordinate Descent [Friedman et al. 2007] algorithm was proposed which can be much faster.

Coordinate descent for the Lasso was popularised by Friedman et al. [2007], though similar ideas were considered in earlier work such as Fu [1998]. The idea is simple – we cycle through each covariate in turn, computing the one dimensional version of the Lasso optimisation, assuming the other coefficients are fixed at the current estimate. More rigorously, we conduct Algorithm 2. The algorithm can be very fast, because the single step update in Step 7 can be very quick if that particular β_k^m is to be set to zero. Tseng [2001] proved the success of this method.

Algorithm 2 Coordinate Descent

Require: $Y \in \mathbb{R}^n, X \in \mathbb{R}^{n \times p}$, and tuning parameter $\lambda > 0$.

1: Initialise $\beta^0 \in \mathbb{R}^p$, $m = 0$.
2: **repeat**
3: $m \leftarrow m + 1$.
4: $\beta^m \leftarrow \beta^{m-1}$.
5: **for** $k = 1, \ldots, p$ **do**
6: Compute conditional correlation $c_k \leftarrow X'_{\cdot,k}\left(Y - X\beta^m + X_{\cdot,k}\beta_k^m\right)/n$.
7: Compute univariate optimisation $\beta_k^m \leftarrow \text{sign}(c_k)(c_k - \lambda)_+ / \left(\sum_{i=1}^n X_{i,k}^2/n\right)$.
8: **end for**
9: **until** β^m has converged.
10: **return** $\widehat{\beta} = \beta^m$.

Unlike Least Angle Regression, Coordinate Descent only computes the solution for a single choice of λ. However, the algorithm converges to the correct solution from

any starting point β^0, with the Lasso loss $\ell_\lambda^{\text{lasso}}(\beta)$ decreasing monotonically, and in particular converges quickly if given a 'warm start' that is close to the solution. A good choice of this in practice is the solution for a different value of λ, which is very useful for computing solutions on a grid of tuning parameters. Because more sparse solutions require less time to compute, the grid is usually sorted in descending order of λ, and chosen as a log-grid – that is, so that $\log(\lambda)$ is evenly spaced.

Some further work have proposed extensions to coordinate descent that improve speed. For example, some improvement is possible by introducing randomness in the selection of features to update, as in Shalev-Shwartz and Tewari [2009]. A wealth of literature exists on these, and further computational refinements, and they may be applied to our work later in this thesis, but are out of the scope of this research.

One final methodological variant are online algorithms. While the batch algorithms previously described work well, for massive datasets, they scale poorly in terms of storage. In particular, by definition all batch algorithms require the storage of the entire data set of X and Y, but when n is very large, this is not possible. While one possibility is to work with small subsamples of the data, such procedures may not perform very well.

Online algorithms solve this problem by storing a current estimate, that is updated as additional samples arrive, one by one. Hence, processing requires only the current sample, of $O(p)$ storage space, instead of the full $O(np)$ dataset. One of the leading methods of online computation for the Lasso is the Regularised Dual Averaging procedure of Xiao [2010]. This method works by calculating the gradient of the loss function at each new sample, averaging with previous gradients, and computing a regularised maximisation. After sufficiently many samples, the dual average becomes a linear approximation to the population loss function, and so Xiao [2010] proved regret bounds and other bounds on the estimates. The procedure has been extended to the Group Lasso in Yang et al. [2010].

The selection of tuning parameters is important for all methodologies. Typically, authors have used cross validation for finding the correct tuning parameter, with 5 or 10-fold cross validation being the most common variant – a survey is available in Arlot and Celisse [2010]. Other authors [Zou et al. 2007] have proved Stein Unbiased Risk Estimation [Stein 1981] based methodology, through calculation of the degrees of freedom of the Lasso. While that can be quite effective, the degrees of freedom of more complex procedures like the Adaptive Lasso remain difficult to calculate, and so we generally use the cross validation based methods in this thesis.

Chapter 3

Lasso Isotone

In this chapter, we define the Lasso Isotone, or Liso algorithm. This algorithm provides a procedure for calculating additive isotonic regression estimates in high dimensions, and poses a motivating example for our later work.

3.1 Motivation

In many cases, in uncovering or describing the dependence of a response on a large number of covariates, parametric and in particular linear models may prove overly restrictive. Additive modelling, as described for instance in Hastie and Tibshirani [1990], is well known to be an useful generalisation.

In additive modelling, we typically assume that the data is well approximated by a model of the form

$$Y_i = f_0 + \sum_{k=1}^{p} f_k(X_{i,k}) + \varepsilon_i,$$

where f_0 is a constant intercept term, and $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)$ is a random error term, assumed independent of the covariates and identically distributed with mean zero. For every covariate $k = 1, \ldots, p$, each component fit f_k is chosen from a space of univariate functions $\mathcal{F}_k$. Usually, these spaces are constrained to be smooth in some suitable sense, and in fitting, we minimise the L2 norm of the residual error,

$$\frac{1}{2}\left\| Y - f_0 - \sum_{k=1}^{p} f_k(X_{\cdot,k}) \right\|^2 := \frac{1}{2}\sum_{i=1}^{n}\left(Y_i - f_0 - \sum_{k=1}^{p} f_k(X_{i,k}) \right)^2,$$

under the constraint that $f_k \in \mathcal{F}_k$, for each $k = 1, \ldots, p$. In the case that ε is assumed to be normal, this can be directly justified as maximising the likelihood. Work on such methods of additive modelling have produced a profuse array of techniques and generalisations. Our particular interest is the work of Bacchetti [1989], which suggested the additive isotonic model.

With the Additive Isotonic Model, we are interested in tackling the problem of conducting regression under the restriction that the regression function is of a prespecified monotonicity with respect to each covariate. Such restrictions may be sensible whenever there is subject knowledge about the possible influence or relationship between predictor and response variables.

In the univariate case, a broad survey of the subject may be found in Barlow et al. [1972]. It turns out in this case, the Pool Adjacent Violators Algorithm, as first suggested in Ayer et al. [1955], allows rapid calculation of a solution to the least squares problem using this restriction alone. By doing so, we retain only the ordinal information in the covariates, and hence obtain a result that is invariant under strictly monotone transformations of the data. In addition, the form of the regression, being simply a maximisation of the likelihood, means that apart from the monotonicity constraint, we do not put on any regularisation or smoothing.

Bacchetti [1989] built on this, by generalising to multiple covariates. Here, the regression function is considered to be a sum of univariate functions of specified monotonicity. Fitting is conducted via the cyclic pool adjacent violators (CPAV) algorithm, in the style of a backfitting procedure built around PAVA — that is, cycling over the covariates, the partial residuals using the remaining covariates are repeatedly fitted to the current one, until convergence. Later theoretical discussion from Mammen and Yu [2007] outlined some positive properties of this procedure.

Nevertheless, CPAV, like many types of additive modelling, can fail in the high dimensional case — for instance, once $p > n$. The particular problem is that the least squares criterion loses strictness of convexity when the number of covariates is large, since it becomes easy for allowed component fits in some covariates to combine in the training data so as to be identical to possible component fits in unrelated covariates. It is hence impossible for the CPAV to distinguish between two radically different regression functions using different covariates since they give the same fitted values on the training dataset.

Some success might be achieved, though, if the solution sought is sparse, in the sense that most of the covariates have little or no effect on the response. As a trivial case, if the identity of the significant variables could be found, the CPAV could be

conducted on this much smaller set of covariates. However, exhaustive search to identify this sparsity pattern would be rapidly prohibitive in terms of computational cost, scaling exponentially in the number of covariates.

In this chapter, we shall attempt to use the Lasso penalised likelihood methodology to deal with these issues, by proposing the Lasso-Isotone (Liso) estimator. By modifying the additive isotonic model to include a Lasso-style penalty on the total variation of component fits, we hope to conduct isotonic regression in the sparse additive setting. The Liso is similar to the degree 0 case of the Lasso knot selection of Osborne et al. [1998], which is also identical to the fused Lasso of Tibshirani et al. [2005], if we replace the covariate matrix with ordered Haar wavelet bases, and do not consider coefficient differences for coefficients corresponding to different covariates. We also omit the additional L1 penalty many authors include in their definition of the Fused Lasso. Our procedure is also similar to the univariate problem considered by Mammen and van de Geer [1997]. In contrast to each of these procedures, however, we allow the additional imposition of a monotonicity constraint, producing an algorithm similar in complexity to the CPAV.

3.2 The Lasso-Isotone Optimisation

In this work, the term 'isotonic' means the functions are assumed to be increasing. However, an assumption of decreasing functions can be accommodated easily for any estimator by applying the algorithm using reversed sign observed covariates. The monotonically increasing function $\tilde{f}$ thus found can then be transformed to be a decreasing function estimate in the original covariates by

$$\widehat{f}(x) = \widetilde{f}(-x).$$

Hence, let us assume without loss of generality that we are conducting regression constrained to monotonically increasing regression functions. Let us first define some terms.

Recall that $X \in \mathbb{R}^{n \times p}$ is the matrix of covariates. For a specified X, for $k = 1, \ldots, p$, let $\mathcal{F}_{\text{iso},k}$ be the space of bounded, univariate, and monotonically increasing functions, that have expectation zero on the k-th covariate. $-\mathcal{F}_{\text{iso},k}$ is then the same for monotonically decreasing functions.

$$\mathcal{F}_{\text{iso},k} := \Bigg\{ f : \mathbb{R} \to \mathbb{R} \quad s.t. \quad \sum_{i=1}^{n} f\left(X_{i,k}\right) = 0,$$
$$\text{and } \exists U, V \text{ s.t. } \forall a < b, U \leq f(a) \leq f(b) \leq V \}$$

Additive isotonic models involve sums of functions from these spaces. It is simple to observe that each $\mathcal{F}_{\text{iso},k}$ is a convex half-space that is closed except at infinity, and so as a result the space of sums of these functions must also be convex and closed except at infinity.

Definition 3.2.1. We define the Lasso-Isotone (Liso) estimator for a particular value of tuning parameter $\lambda \geq 0$ as $\widehat{f}_\lambda(x) = \widehat{f}_0 + \sum_{k=1}^{p} \widehat{f}_{k,\lambda}(x)$, where $\widehat{f}_0$, a constant, and $\widehat{f}_{k,\lambda} \in \mathcal{F}_{\text{iso},k}\, \forall k$ together minimise the Liso loss

$$\ell_\lambda^{\text{liso}}\left(f_0, f_1, \ldots, f_p\right) := \frac{1}{2} \left\| Y - f_0 - \sum_{k=1}^{p} f_k\left(X_{\cdot,k}\right) \right\|_n^2 + \lambda \sum_{k=1}^{p} \Delta(f_k). \tag{3.2.1}$$

here $\Delta(f_k)$ denotes the total variation of f_k, which for $f_k \in \mathcal{F}_{\text{iso},k}$ can be calculated, as illustrated in Figure 3.2.1, as

$$\Delta(f_k) = \sup_{x \in \mathbb{R}} f_k(x) - \inf_{x \in \mathbb{R}} f_k(x).$$

We have introduced a mean zero constraint on the fitted components for identifiability, since we can easily add a constant term to any component fit f_k, and deduct it from another component, and still arrive at the same final regression function. Under this constraint, it follows trivially that, independently of λ, $\widehat{f}_0$ must equal the sample mean of the response, and hence (3.2.1) reduces to the case of optimising over $f_1, \ldots, f_p$ with $\overline{Y} = 0$.

As with the Lasso, the Liso objective function is the sum of a log-likelihood term and a penalty term. It is clear that the domain is convex and, considered in the space of allowed solutions, the objective itself is convex and bounded below. Indeed, outside a neighbourhood of the origin, both terms in the objective are increasing, so a bounded solution exists for all values of λ. However, the objective may not be strictly convex, so this solution may not be unique.

The log-likelihood term does not consider the values of f_k except at observed values of each covariate, while the total variation penalty term, assuming monotonicity, only

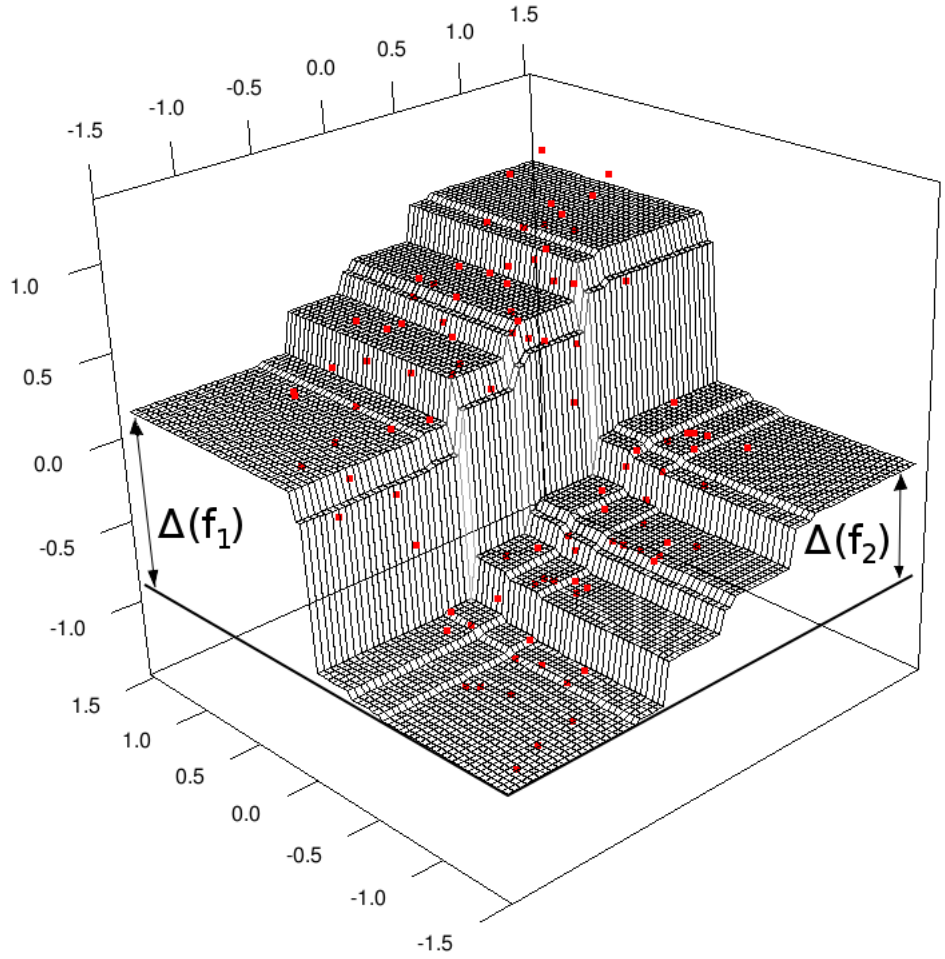

Figure 3.2.1: An example of a Liso solution for $p = 2$. The solution (the gridded surface) minimises the sum of two components - the squared distance between the data points and the surface, and the sum of the two labelled total variations, $\Delta(f_1)+\Delta(f_2)$.

takes account of the upper and lower bounds of the covariate-wise regression function — indeed, for optimality, these bounds must be attained at the extremal observed values of the appropriate covariate, with the solution flat beyond this region. Thus, given any one minimiser to $\ell_\lambda^{\text{liso}}$, another fit with the same function values at observed covariate points, interpolating monotonically between them, will have the same value of $\ell_\lambda^{\text{liso}}$, and so also be a Liso solution. This means that the we can equivalently consider optimisation in the finite dimensional space of fitted values $\widehat{f}_k(X_{\cdot,k})$.

For simplicity, we will represent found Liso solution components by the corresponding right continuous step function with knots only at each observation. For the remainder of this paper, we shall consider uniqueness and equivalence in terms of having equal values at the observed $X_{\cdot,k}$.

The total variation penalty shown here has been previously suggested for regression in Mammen and van de Geer [1997], though in that case, the focus was on

smoothing of univariate functions, without a monotonicity constraint.

3.3 Liso Backfitting

Considering the representation of the Liso in terms of step functions, the Liso optimisation for a given dataset can be viewed as ordinary Lasso optimisation for a linear model, constrained to positive coefficients, using an expanded design matrix $\widetilde{X} \in \mathbb{R}^{n \times p(n-1)}$. Each $\widetilde{X}_{\cdot,J_k} \in \mathbb{R}^{n \times (n-1)}, \quad k = 1, \ldots, p$ contains $n-1$ step functions in the k-th covariate, which form a basis for the vector $f_k(X_{\cdot,k})$, and so isotonic functions in that covariate. The coefficients β optimised over then represent step sizes. In other words, for each $j = 1, \ldots, p(n-1)$, writing $k = \lceil j/(n-1) \rceil$ and $l = j - (k-1)(n-1)$,

$$\widetilde{X}_{i,j} = I_{\{X_{i,k} > X_{(l),k}\}} - (n-l)/(n-1) \tag{3.3.1}$$

$$\beta_j = f_k\left(X_{(l+1),k}\right) - f_k\left(X_{(l),k}\right) \tag{3.3.2}$$

Such a construction is suggested in Osborne et al. [1998], amongst others. Under this re-parametrisation of the problem, existing Lasso algorithms for linear regression may be applied, with a modification to restrict solutions to non-negative values. In particular, the Homotopy/Least Angle Regression algorithm of Osborne et al. [2000] and Efron et al. [2004] is effective, since short cuts exist for calculating the necessary correlations.

On the other hand, the high dimensionality of $\widetilde{X}$ means that standard methods become very costly in higher dimensions, both in terms of required computation, but especially in terms of the storage requirements associated with very large matrices. Hence, we must consider more specialised algorithms for such cases. One such approach involves backfitting, and is workable due to the simple form of the solution when restricted to a single covariate.

3.3.1 Thresholded PAVA

In the $p = 1$ case, it turns out that we have an exceptionally simple way to calculate the Liso estimate, which we will later use to establish a more general multivariate procedure.

With no Liso penalty (i.e. $\lambda = 0$) and a single covariate, the Liso optimisation is equivalent to the standard univariate isotonic regression problem. In this case, the loglikelihood residual sum of squares term is strictly convex, and so, as a strictly convex optimisation on a convex set, an unique solution exists. Trivially, the solution must also be bounded. In fact, there exists, as described in Barlow et al. [1972] and attributed to Ayer et al. [1955], a fast algorithm for calculating the solution – the Pool Adjacent Violators Algorithm (PAVA).

Hence, defining $\widehat{f}_\lambda$ as the minimiser of ℓ^{liso} for λ, we have $\widehat{f}_0 \equiv \widehat{f}_{PAVA}$. The following theorem describes the solutions for other values of λ:

Theorem 3.3.1. *Given $\widehat{f}_{PAVA}$, the Liso solution for $\lambda \geq 0$, $p = 1$ is the Winsorised PAVA fit,*

$$\widehat{f}_{>A_\lambda,<B_\lambda} := \max(\min(\widehat{f}_{PAVA}, B_\lambda), A_\lambda),$$

with A_λ, B_λ thresholding constants that are piecewise linear, continuous and monotone (increasing for A_λ, decreasing for B_λ) functions of λ.

Specifically, if

$$2n\lambda \geq \sum_{i=1}^{n} |\widehat{f}_{PAVA}(X_i) - \overline{Y}|, \tag{3.3.3}$$

then $A_\lambda = B_\lambda = \overline{Y}$.

Otherwise, A_λ, B_λ are the solutions to

$$\sum_{i=1}^{n} (A_\lambda - \widehat{f}_{PAVA}(X_i))_+ = n\lambda \tag{3.3.4}$$

$$\sum_{i=1}^{n} (\widehat{f}_{PAVA}(X_i) - B_\lambda)_+ = n\lambda. \tag{3.3.5}$$

The proof of this theorem, and all other theorems in this thesis, are left for the Appendix.

Corollary 3.3.2. *Let π be a permutation taking $1, \ldots, n$ to indices that put the covariate in ascending order. Then if*

$$n\lambda \geq \max_m \left| \sum_{i=1}^{m} \left(Y_{\pi(i)} - \overline{Y} \right) \right|, \tag{3.3.6}$$

we have that $\widehat{f}_\lambda \equiv \overline{Y}$.

Remark 3.3.1. The PAVA algorithm itself can accommodate observation weights, as well as tied values in the covariates. In terms of the Liso, working with unequal observation weights demands that we work with weighted residual sums of squares. For equations (3.3.4) and (3.3.5), then, weights should be introduced in the summation. Tied values should be also dealt with by merging the relevant steps, and weighting them according to the number of data points at that covariate observation.

3.3.2 Backfitting algorithm

In general, however, simple thresholding fails to solve the Liso optimisation in higher dimensions, due to correlations between steps in different covariates. We can, however, extend the 1D algorithm to higher dimensions by applying it iteratively as a backfitting algorithm.

In other words, we define Liso-backfitting by the following steps:

Algorithm 3 Liso-Backfitting

Require: $Y \in \mathbb{R}^n, X \in \mathbb{R}^{n\times p}, \lambda > 0$.

1: Take $\widehat{f}_0 = \overline{Y}$, $Y \Leftarrow Y - \widehat{f}_0$.
2: Set $m = 0$.
3: Initialise component fits $(f_1, \ldots, f_p)$ as identically 0, or as the estimate for a different value of λ, storing these as the $n \times p$ marginal fitted values.
4: **repeat**
5: $f^m \Leftarrow (f_1, \ldots, f_p)$.
6: $m \Leftarrow m + 1$.
7: **for** $k = 1$ to p (or a random permutation) **do**
8: Recalculate residuals $r_i \Leftarrow Y_i - \sum_{k=1}^{p} f_k(X_{i,k})$, $i = 1, \ldots, n$.
9: Refit conditional residual $\{r_i + f_k(X_{i,k})\}_{i=1}^{n}$ using $X_{\cdot,k}$ by PAVA, producing $\widetilde{f}_k(X_{i,k})$, for $i = 1, \ldots, n$.
10: Calculate thresholds A_λ, B_λ from λ and $\widetilde{f}_k$ by Theorem 3.3.1.
11: Adjust component fit $f_k(X_{i,k}) \Leftarrow \widetilde{f}_{k,>A_\lambda<B_\lambda}(X_{i,k})$.
12: **end for**
13: **until** sufficient convergence is achieved, through considering f^m and f^{m-1}.
14: Interpolate f_k between the samples $X_{i,k}$ and construct $\widehat{f}_\lambda$.
15: **return** $\widehat{f}_\lambda, \widehat{f}_0$.

Step 8 is performed to avoid accumulation of calculation errors, but may be skipped to reduce computation time.

Theorem 3.3.3. *For $f^m = \left(f_0, f_1^m, \ldots, f_p^m\right)$, the sequence of states resulting from the Liso-backfitting algorithm, $\ell_\lambda^{\text{liso}}(f^m)$ converges to its global minimum with probability 1. Specifically, if there exists an unique solution to* (3.2.1)*, f^m converges to it.*

Remark 3.3.2. A proof of this is available via a theorem in Tseng [2001]. However, a simpler alternative proof is available in the case of random permutations, and for completeness is given in the appendix.

Remark 3.3.3. If there is no unique solution, the backfitting algorithm may not necessarily converge, though the Liso loss of each estimate will converge monotonically to the minimum. In addition, because the objective function is locally quadratic, as the change in the Liso loss converges to zero, the change in the estimate after each individual refitting cycle converges also to zero.

Remark 3.3.4. Moreover, defining $X_{(i),k}$ as the i-th smallest value of $X_{\cdot,k}$, if a certain individual step in the final functional fit

$$f_k\left(X_{(i),k}\right) - f_k\left(X_{(i-1),k}\right)$$

has a value of zero in all solutions to the Liso minimisation, then, after a finite number of steps, all results from the algorithm must take that step exactly to zero.

This is because steps being estimated as zero in a Liso solution implies that the partial derivative of the Liso objective function $\ell_\lambda^{\text{liso}}$ in the above individual step direction is greater than zero when evaluated at this solution. The partial derivatives are continuous, and so, as the algorithm converges, the partial derivatives associated with steps that are zero in the optimum of ℓ^{liso} must eventually be above 0 and remain so. But then, this can only be the case following a thresholded PAVA calculation involving the covariate associated with that step if that single covariate optimisation takes the step exactly to zero.

Convergence of the algorithm can be checked for by a variety of methods. One of the simplest is to note that due to the nature of the repeated optimisation, the Liso loss will always decrease in each step, and we will converge towards the minimum. Hence, one viable stopping rule would be to cease calculating when the Liso loss of the current solution drops by too small an amount. Alternatively, we can exploit Remark 3.3.3, and monitor the change in the results in each cycle, stopping when this becomes small.

3.3.3 Choice of regularisation parameter

It will be always necessary to choose a tuning parameter λ to facilitate appropriate fitting. As with the Lasso, too high a tuning parameter will shrink the fits towards

zero. Indeed, consideration of Corollary 3.3.2 shows that, with $\overline{Y} = 0$, and $\pi^{(k)}$ defined as a permutation that puts the k-th covariate into ascending order, a choice of λ greater than

$$\max_{\substack{k=1,\ldots,p,\\ m=1,\ldots,n}} \left| \sum_{i=1}^{m} Y_{\pi^{(k)}(i)}/n \right|$$

will result in a zero fit in every thresholded PAVA step starting from zero, and hence a zero fit overall for the Liso.

Conversely, too small a value of λ will also lead to improper fitting. This arises from two sources. Firstly, as with the Lasso, the noise term may flood the fit, as the level of thresholding is not sufficient to suppress correlations of the noise with the covariate step functions – the columns of $\widetilde{X}$ from Equation (3.3.1). Secondly, λ has a role in terms of fit complexity, with a small value of λ implying that the Liso, when restricted to the true covariates, would select more steps. This means a less sparse signal in the implied Lasso problem, so it becomes in turn more likely for selected columns of $\widetilde{X}$ to be correlated with columns belonging to irrelevant covariates, hence producing spurious fits in the other covariates.

More precisely, in the noiseless case, if the true model function can be written exactly as the sum of step functions with, in the expanded design matrix $\widetilde{X}$, corresponding column indices S, then correct recovery, given that Liso has fit non-zero fits to the true step functions, requires

$$\lambda \geq \widetilde{X}'_{S^c} \left(Y - \widetilde{X}_S \left(\widetilde{X}'_S \widetilde{X}_S \right)^{-1} (\widetilde{X}'_S Y - n\lambda) \right) \tag{3.3.7}$$

$$= \widetilde{X}'_{S^c} \left(\widetilde{X}_S \left(\widetilde{X}'_S \widetilde{X}_S \right)^{-1} n\lambda \right). \tag{3.3.8}$$

This is the Irrepresentable Condition of the Lasso, as detailed in Zhao and Yu [2006], Meinshausen and Bühlmann [2006], and it may fail if S is too large. With the Liso, then, the particular choice of λ itself influences the form the true covariates can take and so alters the criterion for Irrepresentability.

These effects are illustrated in Figure 3.3.1, in which we have generated X, with $n = 100$, p=200, according to an uniform distribution, and produced Y as the sum of $k = 5$ of the covariates. In other words, f is the sparse sum of linear functions. We give the full paths of fits in terms of, firstly, the total variation of fitted components

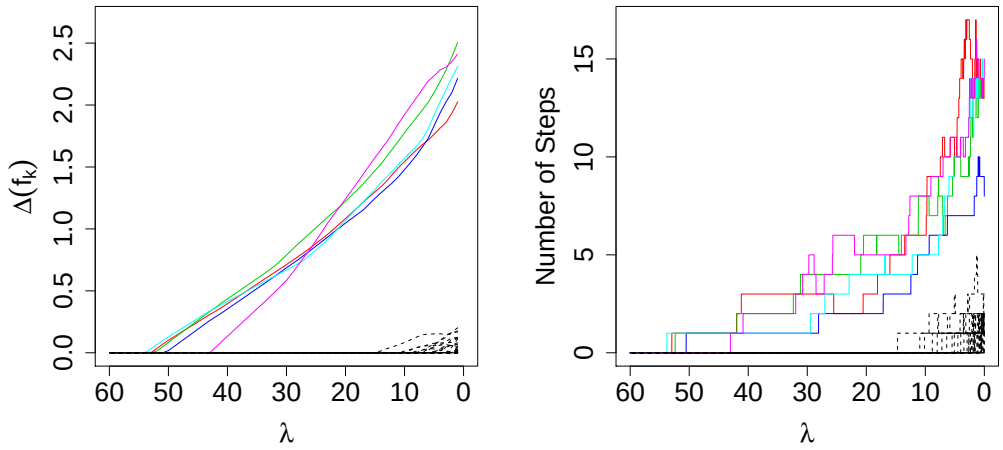

Figure 3.3.1: Effects of changing the regularisation parameter in the noiseless case. $n = 100, p = 200$. Each line represents how an individual covariate's estimate changes as λ varies, with the solid lines for the true covariates, while the dashed lines denote spurious fits on irrelevant variables.

$\Delta(f_k)$, and secondly the number of component steps in each covariate,

$$\left|\left\{i : f_k\left(X_{(i),k}\right) \neq f_k\left(X_{(i-1),k}\right)\right\}\right|.$$

Of particular note is that, unlike the Lasso, even without noise, the size of the basis of step functions and the non-sparsity of the true signal means that as $\lambda \to 0$, we do not converge to the true sparsity pattern. However, with higher λ, the number of steps we choose diminishes rapidly, and as a result we can remove the spurious fits and simultaneously not mistakenly estimate the relevant covariates as zero.

In Figure 3.3.2, we add an independent normal noise component to Y, with variance chosen so that the signal to noise ratio, $SNR = 5$. In the new Total Variation plot, we see that the noise component has added additional noise fits in some of the irrelevant variables, and as in the Lasso these vanish for higher λ. Since the spurious fits vanish before the true covariate components do, we see that recovery of the true sparsity pattern is still possible in this case.

Now, in the above examples, we worked with the true sparsity pattern being assumed known. In real problems, we need to estimate the correct value of λ directly from the data. To do this, with the goal of recovering the correct sparsity pattern, is generally understood to be very difficult. (See e.g. Meinshausen and Bühlmann [2006] for some attempts.)

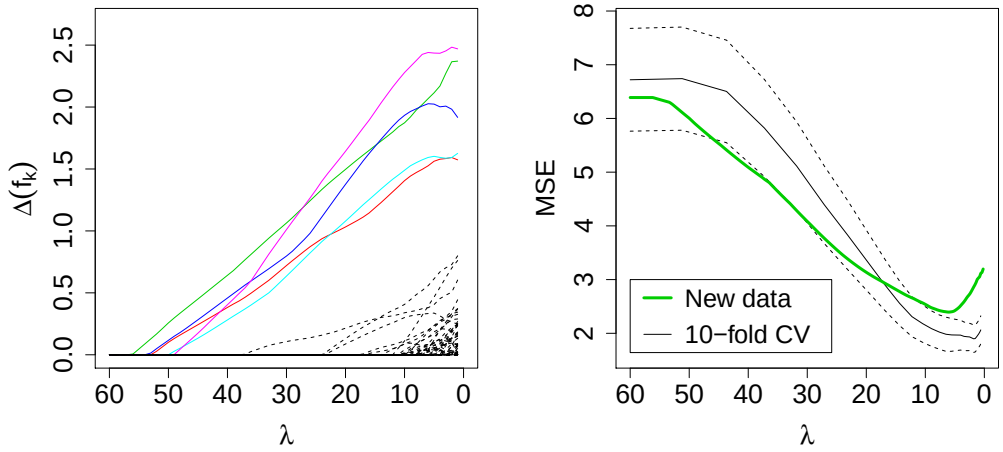

Figure 3.3.2: Effects of changing the regularisation parameter in the noisy case. $n = 100, p = 200, SNR = 5$. We show again in the first graph the total variation of each covariate estimate as λ alters, with solid lines for the truly important covariates, while the dashed lines denote spurious fits on irrelevant variables. The second graph shows the MSE from a 10-fold cross validation procedure with ± 1 s.d. in dashes, as well as the true MSE on a new set of data as the thick line.

However, as suggested in literature from Tibshirani [1996] onwards, cross validation is effective for minimising predictive error, and is illustrated by second graph of Figure 3.3.2. Here, we calculate CV error from a 10-fold cross validation. We may then take the λ that minimises the average mean squared error across the folds. If we desire a simpler model, we can, as is often suggested, take the largest λ that achieves a CV value within 1 s.d. of the minimum. Examining the thick line for the true predictive MSE shows that such a procedure, while not perfect, can give good results. In minimising predictive error, however, we do still fit some irrelevant covariates as non-zero, a phenomenon previously observed with the Lasso in Leng et al. [2006].

Now, unlike a LARS-like approach, Liso Backfitting will only give us the solution for an individual choice of λ. However, CV can still be practical, because coordinate-wise minimisation can be very fast for sparse problems, something already observed for the normal Lasso [Friedman et al. 2007]. We can further reduce the computational cost by noting that Liso solutions for similar values of λ are likely to be similar, and hence use the result for one value of λ as a start point for the calculation for a nearby value of tuning parameter. This is especially effective if we order the λ values we need to calculate in decreasing order, since large λ solutions are more sparse and so faster to calculate.

3.4 Empirical example

We are interested in the success of Liso in correctly selecting variables for varying levels of n and p. We adopt the following setup – we generate pairs $X \in \mathbb{R}^{n \times p}, Y \in \mathbb{R}^n$ by

$$X_{ij} \sim \mathit{Uniform}\,(-1, 1)$$
$$Y_i = 2\,(X_{i,1})_+^2 + X_{i,2} + \operatorname{sign}(X_{\cdot,3})\,|X_{i,3}|^{1/5} + 2I_{\{X_{i,4}>0\}} + \varepsilon_i$$

with $n = 1024$, $p = 1024$, independent $\varepsilon_i \sim N(0, 1)$. The covariates are then centred and standardised to have mean zero and variance 1, and Y is centred to have mean zero.

For $p' = 32, 64, 128, 256, 512, 1024$, $n' = 5, 10, 15, \ldots$, we then take as X', Y' subsets of X, Y corresponding to the first p' columns of X, and random samples without replacement of n' rows of X, Y. Hence we consider the problem of correctly finding 4 true variables, from amongst p' potential ones, based on n' observations. We quantify the success of Liso by looking at the proportion of 50 replications where the algorithm, for at least one value of λ, produces an estimate where the true covariates have at least one step while the other covariates are taken to zero. (We adopt this framework so as to reduce the additional noise from generating a complete new random dataset with each attempt.)

Figure 3.4.1 gives these results. As we can see, as in a variety of Lasso-type algorithms [Wainwright 2009], there is a sharp threshold between success and failure in recovery of sparsity patterns as a function of n. Moreover, as we increase p exponentially, the required number of observations n increases much more slowly, thus implying that $p \gg n$ recovery is possible.

Figure 3.4.2 gives an example of Liso fits arising from this simulation. As can be seen from the marginal plot of the data points, with such a small amount of data, it can be very difficult to find the true model function. Focusing on the dashed lines, which show the results of the Liso under the minimum regularisation required for correct sparsity recovery, we note the high level of shrinkage required to shrink the other variables to zero. This shrinkage exhibits itself as not only a thresholding on the ends of the component fits, which we have seen in the univariate case, but also an additional loss of complexity in the middle parts of each component fit.

However, notice the solid black line, which avoids this shrinkage and thus greatly improves the fit while still keeping the correct sparsity pattern recovery. This solid

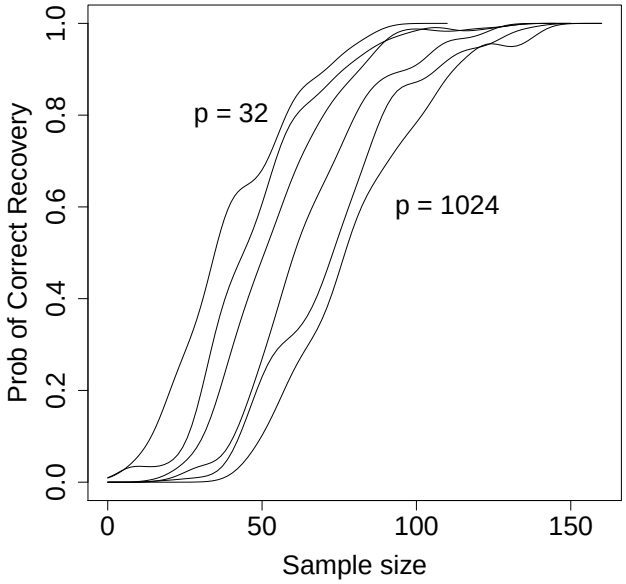

Figure 3.4.1: Probabilities of correct sparsity recovery with 4 true nonlinear but monotonic covariates, $SNR = 4$. Each line shows how the recovery probability changes as the sample size n changes for a single value of p, with p taking values $2^5, \ldots, 2^{10}$.

black line is achieved by a process we call the *Adaptive Liso*, which exemplifies the benefits of the reweighting based recalculation in this particular Liso framework. Note that as an added bonus, we get good sparsity recovery results here with the Adaptive Liso using merely the cross validated sample-determined tuning parameter values, instead of the fine tuned population-based values earlier. We shall discuss the this process in the next chapter.

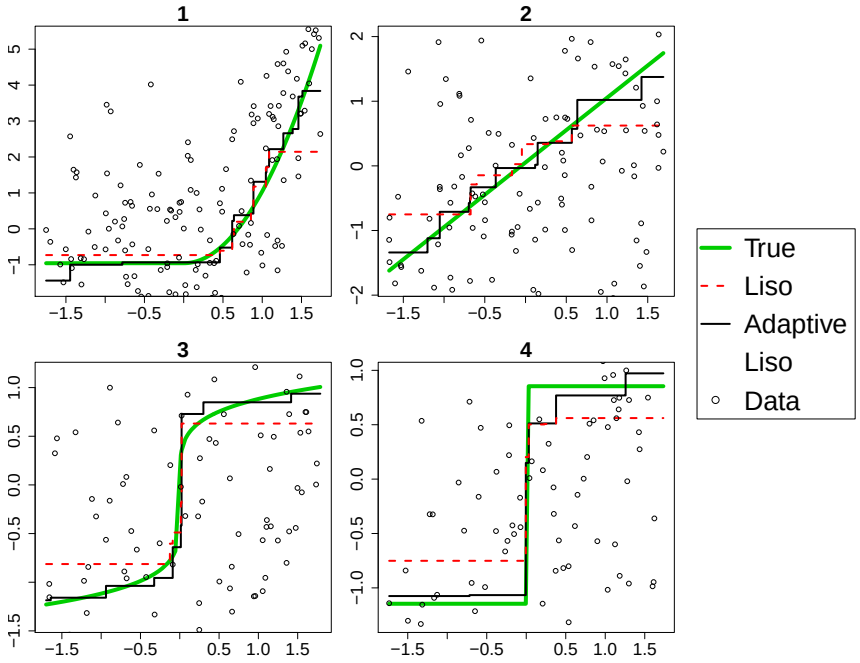

Figure 3.4.2: Example Liso covariate fits, for $n = 180$, $p = 1024$. The true component functions are given by the thick line, while the dashed line gives the raw Liso fit for the smallest amount of regularisation required to bring spurious fits in irrelevant covariates to zero. The solid black line shows a fit made by the Adaptive Liso, using tuning parameters found by cross validation. The fitted and true model functions for all 1020 remaining covariates are all constant zero. The circles are marginal plots of the data points.

Chapter 4

Concave penalties and Reweighting

While the Lasso has proven successful in many situations, an important extension is to change the penalty from the standard Lasso L1 penalty to penalties with concavity, or more generally non-convexity. In this chapter we discuss the general idea of concave penalties and the benefits they bring, how it leads to reweighting based algorithms, and then apply this idea to the Liso.

4.1 Benefits of concave penalties

Recall the general form of a penalised likelihood estimate, which produces $\widehat{\beta}$ minimising

$$\ell_\lambda(\beta) = \frac{1}{2} \|Y - X\beta\|_n^2 + P_\lambda(\beta). \tag{4.1.1}$$

The Lasso penalty $P_\lambda(\beta) = \lambda \|\beta\|_1$, was initially motivated as being the convex relaxation of the sparsity criteria inherent in formulations like the AIC, with convexity here signifying the convexity of the function, and also of the resultant level sets $\{\beta : P_\lambda(\beta) = \tau\}$. Indeed, the Lasso is minimally convex in the sense that, being linear except at the hyperplanes where each $\beta_j = 0$, the addition of any concave function that is not piecewise linear would produce a penalty function that is no longer convex.

Now, convexity produces a number of advantages - in particular, for any choice of $\lambda > 0$, this means the Lasso criterion $\ell_\lambda^{\text{lasso}}$, as the sum of two convex functions, is itself convex. In this case, local minima are global ones, thus allowing a range of convex

optimisation techniques, and so providing considerable advantages in computational terms.

However, it can be frequently desirable to use concave[1], or even generally non-convex penalties instead. There are several reasons for this.

4.1.1 Bias/selection trade-off

The first benefit is that concave penalties can allow a better trade off between bias and selection. Consider as a toy example the case where $p = 2$, and X is orthonormal, so that $X'X = I_2$, the 2×2 identity matrix.

Let

$$Y = X_{\cdot,1} + \varepsilon,$$

where $\varepsilon \sim N(0, 1)$. Hence the true model coefficients are $\beta = (1, 0)$.

For a general penalised maximum likelihood estimate $\widehat{\beta}$, as in Equation (4.1.1), suppose that the method recovers the correct sparsity pattern, so that $\widehat{\beta}_2 = 0$. Then if $P_\lambda(\beta)$ is differentiable in β_1 at $\beta = \widehat{\beta}$, the estimate of the non-zero coefficient, $\widehat{\beta}_1$, would satisfy

$$\begin{aligned}
0 &= X'_{\cdot,1}(X\widehat{\beta} - Y)/n + \frac{\partial P_\lambda}{\partial \beta_1}(\widehat{\beta}) \\
0 &= \widehat{\beta}_1 - \beta_1 - X'_{\cdot,1}\varepsilon + n\frac{\partial P_\lambda}{\partial \beta_1}(\widehat{\beta}) \\
\widehat{\beta}_1 &= \beta_1 - n\frac{\partial P_\lambda}{\partial \beta_1}(\widehat{\beta}) + X'_{\cdot,1}\varepsilon \\
E(\widehat{\beta}_1) &= \beta_1 - nE\left(\frac{\partial P_\lambda}{\partial \beta_1}(\widehat{\beta})\right).
\end{aligned}$$

Correspondingly, selection results also relate to the penalty function $P_\lambda(\cdot)$. Specifically, not selecting $X_{\cdot,2}$ requires that

$$\begin{aligned}
\lim_{\substack{\beta_1 = \widehat{\beta}_1, \\ \beta_2 \to 0+}} n\frac{\partial P_\lambda}{\partial \beta_2}(\beta) &\geq X'_{\cdot,2}(Y - X\widehat{\beta}) = X'_{\cdot,2}\varepsilon \\
\lim_{\substack{\beta_1 = \widehat{\beta}_1, \\ \beta_2 \to 0-}} n\frac{\partial P_\lambda}{\partial \beta_2}(\beta) &\leq X'_{\cdot,2}\varepsilon.
\end{aligned}$$

[1]Note that substantial confusion exists in the literature over the terminology used. We follow here the convention of Zhang [2010]. In contrast Zou and Li [2008] denotes such penalties as *nonconcave* penalties.

For the Lasso with tuning parameter λ, $\frac{\partial P_\lambda}{\partial \beta_k}(\widehat{\beta}) = \lambda \text{sign}(\widehat{\beta}_k)$ for $\widehat{\beta}_k \neq 0$. Therefore, the Lasso penalisation incurs a bias of $n\lambda$ on the estimates it finds to be non-zero. As $|X'_{\cdot,2}\varepsilon|$ may be large, a large λ may be required for $\widehat{\beta}_2 = 0$, and therefore the Lasso can be forced to give inaccurate estimates. Any other convex penalty function that produces sparsity would also cause similar problems.

In contrast, concave penalty estimates gives us a way of avoiding this. Consider that the selection results above require a condition on the partial differential of $P_\lambda(\beta)$ near $\widehat{\beta}_2 = 0$, while the bias is based on the partial differential of $P_\lambda(\beta)$ near $\widehat{\beta}_1 \neq 0$. Thus, for the same value of λ used for the Lasso, a penalty function with an one-sided differential $|(\partial P_\lambda/\partial \beta_2)(\widehat{\beta})| = \lambda$ near $\widehat{\beta}_2 = 0$ and $|(\partial P_\lambda/\partial \beta_1)(\widehat{\beta})| < \lambda$ near $\widehat{\beta}_1 \approx \beta_1$ would similarly succeed in setting $\widehat{\beta}_2 = 0$, but would obtain a smaller bias in its estimate. Thus, such a penalty would often be able to obtain a better trade-off between selection criteria and estimation accuracy. This is illustrated in Figure 4.1.1.

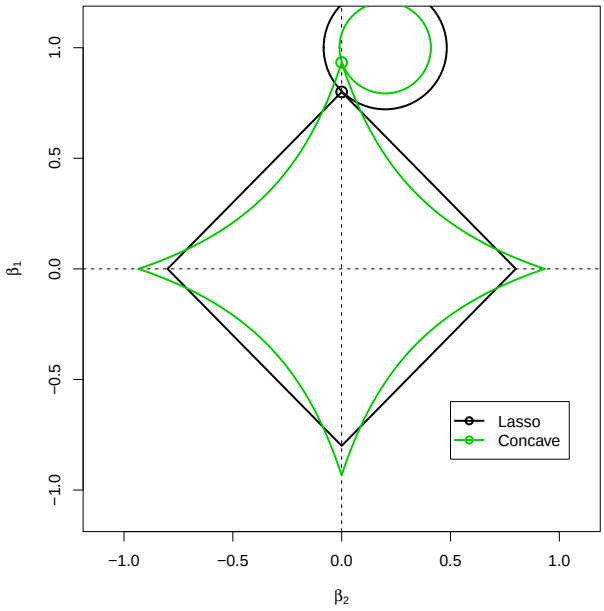

Figure 4.1.1: An example of bias/selection trade-off. The circles, centred around the least squares estimate, represent the residual sum of squares level sets, while the other shapes represent the constraint level sets, for a Lasso, and a concave penalty estimate (in this case, a log penalty). The estimate in this representation is then the points where the two make contact. The particular sets we depict are those associated with the minimum level of penalisation, that still attains the correct sparsity pattern under each penalisation scheme. The concave penalty attains a smaller level of bias, as seen by the smaller circle.

4.1.2 Selection properties

Secondly, concave penalties can allow better selection behaviour under correlated covariates.

Consider now the example in the previous section, but with $p = 3$, and $\beta = (1, 1, 0)$. Let $X_{\cdot,1}, X_{\cdot,2}$ be orthonormal, and $\|X_{\cdot,3}\|_n = 1$, but with $X_{\cdot,3}$ potentially correlated to the two true covariates. Suppose that the correct sign pattern is recovered so that $\widehat{\beta}_3 = 0, \widehat{\beta}_1, \widehat{\beta}_2 > 0$, and $P_\lambda(\beta)$ is differentiable in β_1 and β_2 at $\beta = \widehat{\beta}$. We have, using a similar argument to before,

$$\begin{aligned}
0 &= X'_{\cdot,j}(X\widehat{\beta} - Y) + n\frac{\partial P_\lambda}{\partial \beta_j}(\widehat{\beta}) \quad \text{for } j = 1, 2, \\
0 &= \widehat{\beta}_j - \beta_j - X'_{\cdot,j}\varepsilon + n\frac{\partial P_\lambda}{\partial \beta_j}(\widehat{\beta}) \quad \text{for } j = 1, 2, \\
\widehat{\beta}_j &= \beta_j - n\frac{\partial P_\lambda}{\partial \beta_j}(\widehat{\beta}) + X'_{\cdot,j}\varepsilon \quad \text{for } j = 1, 2,
\end{aligned}$$

and hence

$$\begin{aligned}
\lim_{\substack{\beta_1,\beta_2=\widehat{\beta}_1,\widehat{\beta}_2, \\ \beta_3 \to 0+}} n\frac{\partial P_\lambda}{\partial \beta_3}(\beta) &\geq X'_{\cdot,3}(Y - X\widehat{\beta}) \\
&= X'_{\cdot,3}\varepsilon + X'_{\cdot,3}X_{\cdot,1}\left(\beta_1 - \widehat{\beta}_1\right) + X'_{\cdot,3}X_{\cdot,2}\left(\beta_2 - \widehat{\beta}_2\right) \\
&= X'_{\cdot,3}\varepsilon + \sum_{j=1}^{2}(X'_{\cdot,3}X_{\cdot,j})\left(n\frac{\partial P_\lambda}{\partial \beta_j}(\widehat{\beta}) - X'_{\cdot,j}\varepsilon\right) \\
&= W + \sum_{j=1}^{2}(X'_{\cdot,3}X_{\cdot,j})\left(n\frac{\partial P_\lambda}{\partial \beta_j}(\widehat{\beta})\right) \\
\lim_{\substack{\beta_1,\beta_2=\widehat{\beta}_1,\widehat{\beta}_2, \\ \beta_3 \to 0-}} n\frac{\partial P_\lambda}{\partial \beta_3}(\beta) &\leq W + \sum_{j=1}^{2}(X'_{\cdot,3}X_{\cdot,j})\left(n\frac{\partial P_\lambda}{\partial \beta_j}(\widehat{\beta})\right),
\end{aligned}$$

for W a normal random variable with mean 0. Thus, for the Lasso, a necessary condition for correct selection is:

$$-n\lambda \leq W + n\lambda\sum_{j=1}^{2}(X'_{\cdot,3}X_{\cdot,j}) \leq n\lambda.$$

Hence, if $|\sum_{j=1}^{2}(X'_{\cdot,3}X_{\cdot,j})| > 1$, the Lasso must fail to recover the correct sign

pattern with probability at least 1/2, for all choices of tuning parameter λ. Note the distinction - while in the previous subsection we were concerned about a poor trade-off between bias and selection as a function of the tuning parameter, here, we see that the convexity of the penalty function implies that in problems with such high correlation, the Lasso necessarily fails to select the correct variables completely.

In contrast, a concave penalty can deal better with the presence of such correlations. Under such a penalty, we are free to set $\lim_{\beta_3 \to 0} |\frac{\partial P_\lambda}{\partial \beta_3}(\beta)|$ large, and yet $\frac{\partial P_\lambda}{\partial \beta_j}(\beta)$ small for $j = 1, 2$. Indeed, if

$$\sum_{j=1}^{2} \left| \frac{\partial P_\lambda}{\partial \beta_j}(\widehat{\beta}) \right| < \lim_{\beta_3 \to 0} \left| \frac{\partial P_\lambda}{\partial \beta_3}(\beta) \right|,$$

so long as $|W|$ is small enough, there will always be a local minimum with the correct sign pattern.

4.1.3 Complex prior information

Much of the literature [Zou 2006][Zou and Li 2008][Zhang 2010] has noted the previously mentioned advantages. In this thesis, however, we observe one additional benefit: Concave penalties allows us to integrate a range of complicated prior information in a natural, and effective, way.

Considering the formulation of the Lasso as the maximum a-posteriori probability estimate of a Bayesian model with a Laplacian prior, in general convex penalties corresponds to priors with log-concave density functions. Clearly, relaxing this log-concavity increases the types of potential priors that can be specified. From an alternative perspective, relaxing the convexity of the penalty changes the constraint set in the implied constrained optimisation to a non-convex one. In either case, we allow that more complicated assumptions be incorporated.

One particular case is especially interesting to us. Recall the example of the Group Lasso [Yuan and Lin 2006] in Section 2.3. In this scenario, we are endowed with a certain grouping structure $\mathcal{G}$ upon the variables. For a given $G \in \mathcal{G}$, with $G \subset \{1, \ldots, p\}$, we wish that the selection of some variable $j \in G$ encourages the selection of other variables $k \in G$.

In the case of the Group Lasso, we have a solution in this scenario where all variables in a group are selected simultaneously. However, it may be the case that we have sparsity within groups, instead of just sparsity amongst groups.

The combination of sparsity within groups and sparsity of groups is a relatively new idea, considered only recently by authors including Breheny and Huang [2009] and Friedman et al. [2010]. Conventionally, it may be intuitive to consider a trade-off between the importance of the grouping structure, and the importance of the sparsity assumption. However, there are cases where these assumptions can arise in combination, and be in fact complementary. For example, in the case of dummy variables when dealing with categorical covariates, such a situation might arise whereupon we desire both parsimony in terms of the categories used, but also expect that only some of the categories of each covariate that does matter, have a real impact on the response. Alternatively, we might generate, from each of our original covariates $X_{\cdot,k}$, additional covariates as its powers: $X^2_{\cdot,k}, X^3_{\cdot,k}$ and so on. In this case, sparsity in the original covariates translates to group sparsity, while we may have variable sparsity in terms of controlling the order of the final fitted polynomial. As a final example, one might consider the case of the Liso – in our expanded basis $\widetilde{X}$ from Equation (3.3.1), we expect group sparsity arising from the individual covariates, but also desire within-group sparsity so as to produce simple fits.

By considering both types of sparsity, we can produce estimates that reflect more accurately our prior assumptions, and if this structure is inherently present in the data, take advantage of it to produce more precise results. This sort of combination is not accommodated by the Group Lasso.

To see the advantage of concave penalties in this context, consider a problem where $X_{\cdot,1}$ and $X_{\cdot,2}$ are in the same group, but $X_{\cdot,3}$ is in a separate group. In such a small problem, group sparsity and sparsity within groups are rather artificial and not very useful, but the example helps illustrate issues that also appear in larger scenarios.

Now, for the pairs (β_1, β_2) and (β_1, β_3), consider plotting the level set of coefficient values that satisfy a certain level of penalisation, with the remaining coefficients set as 0. Now, at the 'corners' of the level set, the angle of the boundary indicates the necessary ratio by which one coefficient must decrease so that the other could increase and keep $P_\lambda(\beta)$ constant. In such figures, then, sharper corners indicate additional sparsity, because this means that to decrease $\ell_\lambda(\beta)$ by deviating from the $\beta_j = 0$ solution, a small increase in $|\beta_j|$ must offset, in terms of reducing $\|Y - X\beta\|_n^2$, a large decrease in $|\beta_k|$, for whatever $\beta_k \neq 0$.

Hence, for group sparsity to be present, we require sharper corners in the (β_1, β_3) plot, than the (β_1, β_2) plot, thus showing that coefficients patterns involving multiple groups are more penalised than ones involving just one group. Meanwhile, for some

within group sparsity, we require at least somewhat sharp corners be present in the (β_1, β_2) level set.

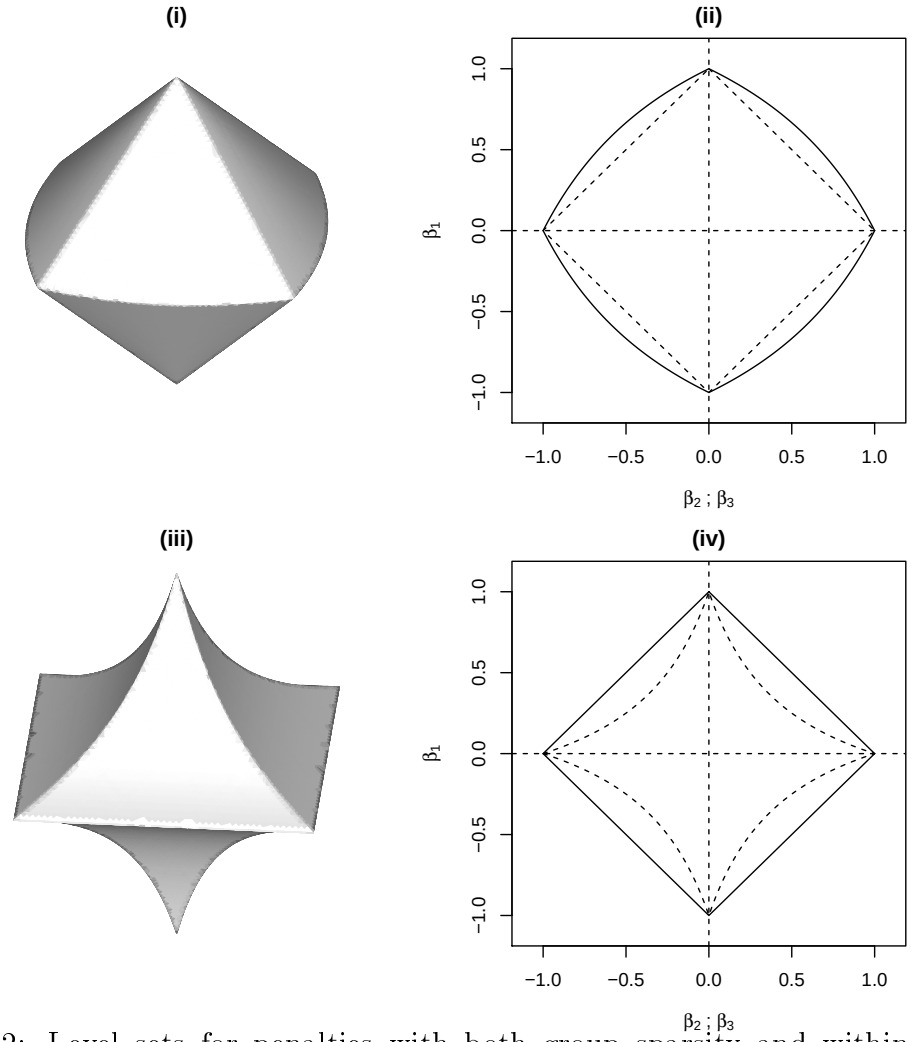

Figure 4.1.2: Level sets for penalties with both group sparsity and within group sparsity. (i) and (ii) show the SGL, while (iii) and (iv) show a concave penalty based approach. In (ii) and (iv), we show cross sections of the level set for β_1, β_2 in solid lines, and for β_1, β_3 in dashed lines. For a grouping effect the dashed lines must show sharper corners than the solid lines.

The Sparse Group Lasso [Friedman et al. 2010] is one proposed answer to this problem, which functions by additively combining a Lasso penalty and a Group Lasso penalty, producing a penalty of the form:

$$P_\lambda(\beta) = \lambda\alpha \sum_G \|\beta_G\| + \lambda(1-\alpha) \sum_{j=1}^{p} |\beta_j|, \quad \lambda, \alpha > 0$$

Under the Sparse Group Lasso, the penalty produces level sets of the form shown

in Figure 4.1.2 (i) and (ii). The inter-group level set – that is, the level set involving (β_1, β_3), the dashed line in Figure 4.1.2 (ii), is a cross-polytope, similar to that of the Lasso. However, to produce the grouping effect, then, the solid line in Figure 4.1.2 (ii) shows that the within group penalty on the pair (β_1, β_2) is effectively that of a naive elastic net penalty. Zou and Hastie [2005] observed that such a naive elastic net often performs poorly when it is not close to either a ridge penalty or a Lasso penalty, due to the issue of 'double shrinkage', and so we find in our empirical experiments that the SGL often performs poorly.

Concave penalties offer a new way to solve this problem. Suppose that we allow the penalty on (β_1, β_2) to be any sparsity promoting penalty, and in particular, the Lasso itself. Then for a grouping effect, one possible way is to simply promote *additional sparsity* in the between group penalisation $(\beta_1, 3)$. This can be simply be accomplished by using a concave penalty to aggregate the appropriate within group penalty functions. For example, a penalty of the form

$$P_\lambda(\beta) = \lambda \sum_G \log\left(\sum_{j \in G} |\beta_j| + \delta\right), \quad \lambda > 0 \tag{4.1.2}$$

would produce the level sets in Figure 4.1.2 (iii) and (iv). Thus, we are freed from requiring double shrinkage on the within-group components, as in the SGL case, if we wish to involve some sort of grouping structure. This rationale lies behind the 'Co-adaptive Lasso' procedure of Chapter 5.

4.2 Implementation and Reweighting

The benefits we see in Section 4.1 are general in the sense that they apply to any penalty function that involve a differential discontinuity at 0, and have a diminished partial derivative far away from zero, or in the case of 4.1.3, work to encode the complex assumption in some way. Several specific concave penalties have been proposed and considered in the literature.

- The Adaptive, or Logarithm Penalty [Zou and Li 2008]:

$$P_\lambda(\beta) = \lambda \sum_k \log(|\beta_k| + \delta)$$

- The MCP penalty [Zhang 2007], for $\gamma > 0$:

$$P_\lambda(\beta) = \lambda \sum_k \int_0^{|\beta_k|} (1 - \gamma t)_+ \, dt$$

- The SCAD penalty [Fan and Li 2001], for $\gamma_1, \gamma_2 > 0$:

$$P_\lambda(\beta) = \lambda \sum_k \int_0^{|\beta_k|} (1 - \gamma_2 \max(t, \gamma_1))_+ \, dt$$

Each can be used within the formulation in Equation (4.1.2) to obtain the effect of grouping with sparsity.

However, the problem with concave penalties is that with the loss of convexity, it is no longer the case that local minimums of the criterion correspond to global minimums. Further, many standard convex optimisation algorithms cannot be guaranteed to converge.

The focus of implementation is, nevertheless, the computation of such local minimums.

4.2.1 Implementation of concave penalties

One possibility for computation is to create locally quadratic approximations, as suggested in Fan and Li [2001], but this is often unsuccessful, because the quadratic optimisation fails to preserve sparsity of solutions. Direct thresholding to delete small coefficients is thus required, which itself requires careful, and difficult tuning to work. Stochastic search via, for example, simulated annealing [Kirkpatrick et al. 1983], can also be available, but for large problems these can be unreliable and take long to converge.

In the case of MCP and SCAD, Zhang [2007] observed that the local minima of their solutions, together with λ, form piecewise linear subspaces in $\mathbb{R}^{p+1}$. Indeed, in general, any penalty that is linear in its first derivative would have this property. Hence, for such types of penalties, a Least Angle Regression/Homotopy type algorithm in the style of Algorithm 1 is available. However, MCP type penalties involve a second tuning parameter, γ, and the solutions are not locally linear in this parameter. The second parameter can have a large impact on the solutions, because since $(\partial P_{\lambda,\gamma}/\partial \beta_k)(\beta) = 0$ for $|\beta_k| > 1/\gamma$, they determine values beyond which each

covariate is effectively unpenalised. Optimisation for these parameter is therefore important, and may pose a challenge in practical implementation. Homotopy type algorithms are also slow for some large problems, as can also be observed in the case of the Lasso.

One final method of implementation is the Locally Linear Approximation (LLA) method of Zou and Li [2008]. Here, we consider an iterative scheme to compute a local minimum of this criterion by taking repeated linear approximations of the penalty, in particular producing a Lasso computation at each stage. Specifically, for a general formulation as in (4.1.1), with a current estimate of $\widehat{\beta}^{(m)}$, we approximate

$$P_\lambda(\beta) \approx \sum_{k=1}^{p} \frac{\partial P_\lambda}{\partial \beta_k}(\widehat{\beta}^{(m)})(\beta_k - \widehat{\beta}_k^{(m)}) + P_\lambda(\widehat{\beta}^{(m)}) = \sum_{k=1}^{p} \frac{\partial P_\lambda}{\partial \beta_k}(\widehat{\beta}^{(m)})\beta_k + C,$$

where C is independent of β, with $\partial P_\lambda / \partial \beta_k$ being a subgradient in the cases where the derivative is discontinuous at that point.

If P_λ is an even function in each coefficient, we may thus calculate our new estimate as

$$\widehat{\beta}^{(m+1)} = \arg\min_\beta \frac{1}{2} \|Y - X\beta\|_n^2 + \sum_{k=1}^{p} \frac{\partial P_\lambda}{\partial \beta_k}(|\widehat{\beta}^{(m)}|)|\beta_k|,$$

which is a weighted Lasso estimate. Note that we use the absolute value of β to ensure our estimate obtains sparse results in each step. We then iterate this process until convergence is achieved, or some other stopping criteria is reached. Zou and Li [2008] proved that, as the above produces a majorising function for the original penalised likelihood, we thus produce a MM algorithm [Lange et al. 2000] and so will converge to a local minimum of (4.1.1). (Note that Zou and Li [2008] considered only penalties that are additive in β, though this is not necessary. We have relaxed this constraint, so as to consider penalties such as that in Equation (4.1.2).)

Because the individual steps of the algorithm are each the Lasso, that enables us to use any of a range of individual methods to compute it, including the Homotopy algorithm, and Coordinate Descent. With the Adaptive penalty, the LLA gives rise to a method similar to the Adaptive Lasso of Zou [2006].

For our work, we shall focus on variants of the Adaptive penalty via the LLA. However, we fix $\delta = 0$ in the definition of P_λ. This is because while the value of δ offers a potential second way to tune the estimator, in practice, the penalty obtained by setting δ close to zero is successful in many cases. Using the LLA allows us to

avoid issues arising from the degeneracy of the solution near zero – while the gradient of the penalty increases to infinity, the optimum at each stage of the LLA just simply sets the corresponding next value of $\widehat{\beta}_k$ equal to 0. Removing consideration of this parameter, then, means we only have one parameter to fix in applying the algorithms, an important advantage when it comes to applying cross validations and other such computationally intensive procedures. We do pay a price in that, for many weight formulations, initial mistakes that set too many $\widehat{\beta}_k = 0$ may not be corrected in future steps, but this appears to make not much of a difference in practice.

4.2.2 Reweighting methods

However, the fact remains that that these methods compute only local minimums of $\ell_\lambda(\beta)$. Zhang [2007] proved various uniqueness results for MCP type penalties, under conditions relating to high levels of sparsity. In those situations, the local minimums obtained by their Homotopy algorithm are exactly the global minimums. In general, however, there can be multiple local minima if these conditions fail. With the LLA, it can be seen that if the sequence of λ values for each stage of the algorithm becomes eventually constant and equal to λ_∞, say, then the sequence of $\widehat{\beta}^{(m)}$ arising from the LLA must converge to a local minimum of the corresponding $\ell_{\lambda_\infty}(\beta)$. But different sequences can clearly produce very different estimates, including ones that give $\widehat{\beta} = 0$. Hence, while the benefits shown in Section 4.1 apply to some local minima, it is unclear that they necessarily apply to the particular local minima our algorithms converge to.

Further, Zou and Li [2008] observed that while the convergence theory for the LLA suggest infinite repetitions of the procedure, in empirical experiments, a finite number of repetitions – indeed, often just two calculations – can produce good results.

Our strategy in this thesis to solve this is to adopt the LLA type of procedure, but to consider *the properties of the procedure itself*, and not the concave penalty it is approximating.

We allow individual tuning parameter selection at each stage of the procedure, instead of a single fixed value. We do not repeat the algorithm to convergence, but instead go through a pre-set, finite number of stages, typically just two. And in our later theory, we allow various filters and transformations of covariate weightings that are strictly speaking not in correspondence to any possible penalty. In other words, we use the concave penalty ideas simply to inspire *'Reweighting Methods'* along the lines of:

1. Conduct an initial Lasso.

2. Compute weights by some method.

3. Compute a second weighted Lasso.

4. (Optionally, recompute weights and do additional Lasso computations.)

A brief empirical example serves to motivate our work. Consider the case of the Adaptive Lasso, for a problem where $X_{i,j} \sim N(0,1)$, with $n = 200, p = 400$, $Y|X \sim N(X\beta, 2)$ and β is such that $\beta_k = I_{\{k \leq 10\}}$. Using 10-fold CV to compute tuning parameters, Figure 4.2.1 compares the effects of individually optimising at each stage, compared to using a single fixed λ, on the estimation error $\left\|\widehat{\beta} - \beta\right\|^2$. We take the average of 100 replications.

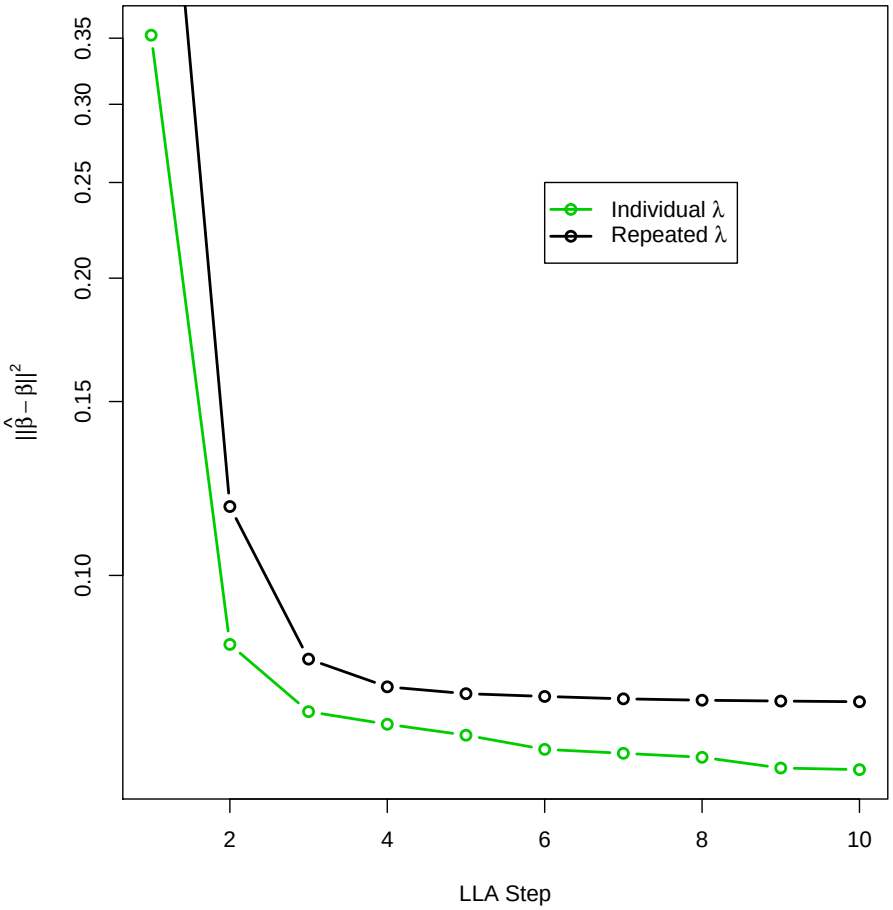

Figure 4.2.1: Estimation error for repeated-λ vs individual-λ, on each stage of a LLA computation, for the scenario in Section 4.2.2. The individual λ is chosen by a 10-fold CV at each stage, while the repeated λ is that which minimised the 10-fold CV error on the 10th and final stage estimate.

We see that while repeated-λ LLA procedures do eventually produce good results, better results are produced, and in fewer steps, by individually optimising each tuning parameter. In Chapter 5 we will develop more rigorously a regression procedure based on this insight, and prove some performance bounds.

4.3 Covariate reweighting with the Liso

Now, returning to the Liso, we see that a variety of extensions and variations of the basic Liso procedure may be proposed, that may offer improvements in some circumstances. For instance, bagging [Breiman 1996] may be used with the Liso, by aggregating the results of applying the Liso to a number of bootstrap samples through any of a variety of methods. However, this method is not reliably a great improvement and will almost inevitably reduce the degree of sparsity in the fit, for any given degree of regularisation.

What is more valuable is to make use of reweighting methodology. Observe that a problem with the Liso is that it treats the constituent steps of each fit individually. After all, the Liso penalty decomposes easily into a sum of components:

$$\sum_{k=1}^{p} \Delta f_k = \sum_{k=1}^{p} \sum_{i=2}^{n} |f_k(X_{(i),k}) - f_k(X_{(i-1),k})| = \sum_{j=1}^{(n-1)p} |\beta_j|,$$

with the β_j corresponding to the implied Lasso problem as in Equation (3.3.2).

In other words, there is no difference, in the eyes of the optimisation, between a fit that involves simple fits in a large number of different covariates, and a more complex fit in a small number of covariates. As a result, the method may not achieve a great deal of sparsity in terms of covariates used, and moreover, in failing to respect this sparsity sufficiently if it is present, can produce inferior prediction and estimation performance. We want to rectify this by making the algorithm in some sense recognise the natural grouping of steps in the step function basis.

Consider applying the ideas from the previous section in the following way: We first conduct an ordinary Liso optimisation, arriving at an initial fit

$$\widehat{f}^0 = \left(\widehat{f}^0_0, \widehat{f}^0_1, \ldots, \widehat{f}^0_p\right).$$

Then, we conduct a second Liso procedure, this time introducing covariate weights

$w_1, \ldots, w_p$ based on the first fit, and use the results of this as the output. We define the Adaptive Liso as the implementation of this, with $w_k = 1/\Delta\widehat{f}_k^0$, for $k = 1, \ldots, p$. This is given in Algorithm 4.

Algorithm 4 Adaptive Liso

Require: $Y \in \mathbb{R}^n, X \in \mathbb{R}^{n \times p}, \lambda, \mu > 0$.

1: Calculate initial fit $\widehat{f}^0$ using Liso with tuning parameter λ. (For instance, using Algorithm 3.)
2: Set $w_k = 1/\Delta(\widehat{f}_k^0)$, for $k = 1, \ldots, p$.
3: Calculate, using e.g. Algorithm 3,

$$\widehat{f} = \arg\min_{f_0, f_1, \ldots, f_p} \frac{1}{2} \left\| Y - f_0 - \sum_{k=1}^{p} f_k(X_{\cdot,k}) \right\|^2 + \mu \sum_{k=1}^{p} w_k \Delta(f_k),$$

with $f_k \in \mathcal{F}_{\text{iso},k}$, $k = 1, \ldots, p$.
4: If necessary, set $\widehat{f}^0 \equiv \widehat{f}$, and repeat from Step 2.

The analogy to the Adaptive Lasso is that we apply a relaxation of the shrinkage for covariates with large fits in the initial calculation, and strengthen the shrinkage for covariates with small fits – indeed, omitting entirely from consideration covariates initially fitted as zero. Usually, we do not require more than one reweighted calculation.

The Adaptive Liso encourages grouping of the underlying Lasso optimisation because large steps contribute to relaxation of other steps in the same covariate. In addition, it means that we in general require less regularisation of true fits in order to shrink irrelevant covariates to zero. We will also always enhance covariate-wise sparsity through this procedure – indeed, the fact that we reject straight away previously zero variables ensures the computational complexity of the method is usually at most equal to that of repeating the original Liso procedure for each iteration.

For the choice of tuning parameter, we implement a procedure based on individual prediction error minimising cross validation at every step, and our empirical studies suggest that this can pose significant improvements over the basic Liso. In our experiments, we also implement a variant of the Adaptive procedure, Liso-SCAD, where instead the weights are derived from a group-wise SCAD penalty. Liso-SCAD and Liso-Adaptive hence both fit under a broad group of possible reweighting procedures.

Remark 4.3.1. An additional application for the Adaptive Liso is in the case where the direction or the presence of monotonicity for the model function component in each covariate is not known. One possible heuristic of dealing with this situation

is to choose signs by a preliminary correlation check with the response. However, correlation is not invariant under general monotonic transformations, and examples exist where covariates have positive marginal effects, but, due to correlations between the covariates, turn out to have negative contributions in the final model.

Now, it is a well known fact [Itô 1993] that functions of bounded variation have an unique Jordan decomposition into monotonically increasing and decreasing functions,

$$f \equiv f^+ + f^-,$$

such that $\Delta(f) = \Delta(f^+) + \Delta(f^-)$. It follows that the monotonicity-relaxed form of the Liso, the total variation penalised estimate, can be dealt with by Algorithm 3 by including as 'covariates', both the original covariates and sign reversed versions of themselves. The original covariate is used to estimate f^+ while the reversed covariate finds f^-. These can then be combined to give an estimate. In this scheme, Liso can be thought of as setting the penalty on the decreasing component to infinity, and that on the increasing component to a finite quantity. For a related approach, see Tibshirani et al. [2010].

This suggests a possible variation to the Liso-Adaptive scheme above: Conduct firstly a non-monotonic total variation penalised fit, consider the fit in terms of its increasing and decreasing components, and then compute separate weights to be placed on the increasing and decreasing components in a second stage fit. We see in this case the same effect seen in the Adaptive Liso, where we have strengthened shrinkage of small function fits towards zero compared to the total variation penalised fit. However, in addition, fits found to be monotonic in the first stage will remain monotonic in the second stage, while functions with small negative or positive components in the initial fit will be shrunk towards a monotonically increasing or decreasing function respectively.

4.4 Empirical results for the Adaptive Liso

We will present a series of numerical examples designed to illustrate the effectiveness of the Adaptive Liso in handling additive isotone problems. The experiments are calculated in R, using a standard desktop workstation. The comparison studies and fits are conducted using an implementation of the backfitting algorithm, with a logarithmic grid for the tuning parameter.

4.4.1 Boston Housing dataset

The Boston Housing dataset, as detailed in Harrison and Rubinfeld [1978], is a dataset often used in the literature to test estimators – see e.g. Hastie et al. [2003] and Ravikumar et al. [2007]. The dataset comprises of $n = 506$ locations in the suburbs of Boston, where at each location 13 covariates, plus one response variable are measured. The response here is the median house price at each observation location, known to be censored at the value 50. Meanwhile the covariates range from crime statistics to discrete variables like index of accessibility to highways. We use here the version included in the R `MASS` library, though we shall discard the indicator covariate `chas`, for ease of presentation. (Experiments suggest that this variable does not have a great effect on the response, in any case.)

As suggested in Ravikumar et al. [2007], we will test the selection accuracy of the model by adding $U(0, 1)$ irrelevant variables. We add 28, so that our final $p = 40$. Since signs are not known, we will apply the sign discovery version of the Liso from Remark 4.3.1, by first conducting a non-monotonic total variation fit, and then a weighted second fit. Tuning parameters are chosen by two 10-fold cross validations.

After application of these algorithms, our selected model is

$$\begin{aligned} Y =& f_0 + f_1(\texttt{crim}) + f_2(\texttt{nox}) + f_3(\texttt{rm}) + f_4(\texttt{dis}) + f_5(\texttt{tax}) \\ &+ f_6(\texttt{ptratio}) + f_7(\texttt{lstat}) + \varepsilon. \end{aligned}$$

The remaining covariates are judged to have an insignificant effect on the response, with zero regression fits. f_3 was found to be monotonically increasing, f_1 slightly non-monotonic, and the remaining functions monotonically decreasing. The full results are shown in Figure 4.4.1.

We see in our experiments that for higher values of λ, we successfully remove all the irrelevant variables, and end up with only a small number of selected variables to explain the response. However, in the one step procedure, the amount of shrinkage required is often large. With cross validation as a criterion, we choose a λ that involves some irrelevant variables as well, though these are in general small in magnitude. A second step greatly improves the model selection characteristics, as well as creating monotonicity which is often absent in the first step - observe especially the case for `nox`.

It is interesting to contrast our fit with the findings from using SpAM [Ravikumar et al. 2007]. Bearing in mind that our problem was in some sense more difficult,

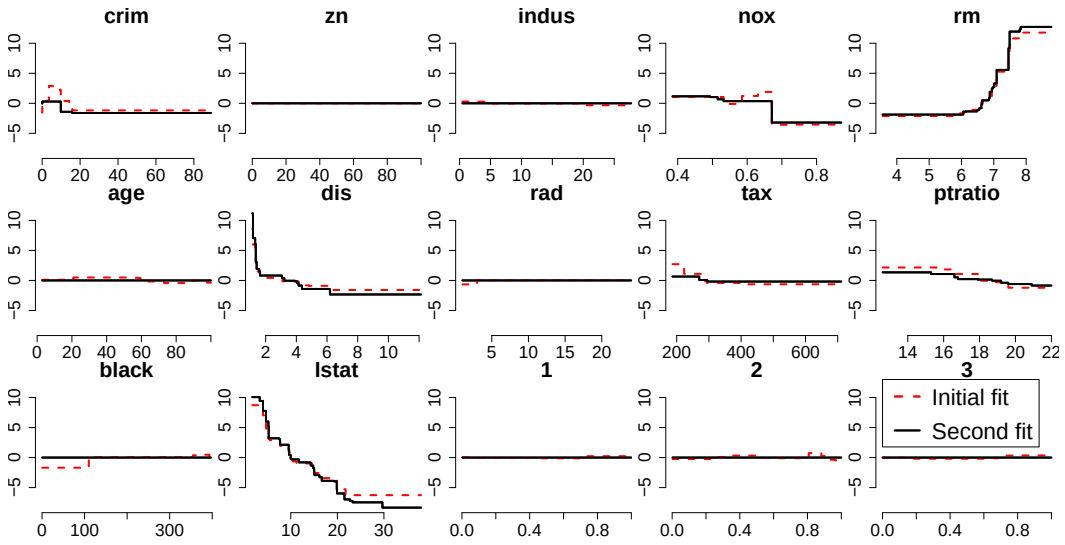

Figure 4.4.1: Fitted component functions on the Boston Housing dataset, for covariates originally present in the data plus three others. The dashed line shows the selected model after the first Liso step, while the solid black line shows the final result of the adaptive sign finding procedure. The single step fit produced additional non-zero fits in some of the artificial covariates, which are not shown, while the two step procedure fit all of them as zero.

since we had 12 original covariates instead of 10 (`rad` and `zn` were not included in the SpAM study), and 28 artificial covariates instead of 20, our findings are largely similar. In addition to the covariates selected in SpAM, we add a fairly large effect from `nox`, and smaller effects in `dis` and `tax`. The most significant fits on `rm` and `lstat` are very similar, though the Liso fit is clearly less smooth. However, while almost all of the fits from SpAM exhibit non-monotonicity, the Liso fit we have found is monotonic, aside from a small step fit near 0 in `crim`.

The non-monotonicity found in `crim` may seem problematic, given the interpretation of that covariate as a crime rate. However, the small increasing step found near `crim` $= 0$ might be reasonable if such areas are qualitatively different from others. On the other hand, perhaps it would be reasonable to directly impose a monotonicity constraint instead.

4.4.2 Comparison studies

We shall now compare the Liso and Adaptive Liso to a range of other procedures in some varying contexts. Varying f between scenarios, consider generating pairs X, Y by, for each repetition,

$$X_i^{(j)} \sim \mathit{Uniform}\,(-1,1)\,, \quad i = 1,\ldots,n,\ j = 1,\ldots,p$$

$$\varepsilon_i \sim N(0,1), \quad i = 1,\ldots,n$$

$$Y_i = f(X_{i,\cdot}) + \sigma\varepsilon_i, \quad i = 1,\ldots,n.$$

100 repetitions were done of each combination of model and noise level, with σ chosen to give $SNR = 1, 3$ or 7, plus one further case where we have $SNR = 3$ but X is instead generated to have stronger correlation between the covariates, as a rescaled (to the range $-1 \leq X \leq 1$) version of $\Phi(Z), Z \sim N(0,\Sigma)$, with $\Sigma_{i,j} = 2^{-|i-j|}$. Here Φ is the Normal CDF.

For comparison, we will compare the performance of Liso, Liso-Adaptive and Liso-SCAD, calculated using the backfitting algorithm, to

- Random Forests (RF), from Breiman [2001]. A tree based method using aggregation of trees generated using a large number of resamplings.
- Multiple Adaptive Regression Splines (MARS), from Friedman [1991], using the `earth` implementation in R. A method using greedy forward/backward selection with a hockeystick shaped basis. We use a version restricted to additive model fitting.
- Sparse Additive Models (SpAM), from Ravikumar et al. [2007]. A Group Lasso based method using soft thresholding of component smoother fits.
- Sparsity Smoothness Penalty (SSP), from Meier et al. [2009]. A Group Lasso based method using two penalties – a sparsity penalty and an explicit smoothness penalty.

For the choice of tuning parameter in all algorithms, we take the value that minimises the prediction error on a separate validation set of the same size as the training set, shared between all the methods. (Note that in the case of SSP, due to the slowness of finding two separate tuning parameters, we instead perform a small number of initial full validation runs for each scenario. We then plug in the averaged smoothness tuning parameter in all following runs, optimising for only the sparsity parameter.)

We record both the mean value across runs of the MSE on predicting a new test set (generated without noise), and, in brackets, the mean relative MSE, defined for

the K-th algorithm on each individual run as

$$MSE^K_{\textbf{Relative}} := \frac{MSE^K}{\min_{J=1,\ldots,7} MSE^J}.$$

We show in bold text the best performing estimator in each scenario.

4.4.2.1 All components linear

In this case, we have the response being just a scaled sum of 5 randomly chosen covariates, plus a noise term. $n = 200$, $p = 50$ overall. In the test set, the variance of the response (and hence the MSE of a constant prediction) was approximately 1.7.

Algorithm	**SNR = 7**	**SNR = 3**	**SNR = 1**	**SNR = 3, Cor**
Liso	0.113 (4.70)	0.186 (3.33)	0.358 (2.41)	0.203 (3.43)
Liso-Adaptive	0.070 (2.94)	0.118 (2.18)	0.242 (1.62)	0.134 (2.27)
Liso-SCAD	0.113 (4.71)	0.186 (3.33)	0.437 (3.00)	0.202 (3.41)
SpAM	0.082 (3.29)	0.149 (2.57)	0.346 (2.24)	0.159 (2.59)
SSP	**0.026 (1.00)**	**0.061 (1.00)**	**0.167 (1.02)**	**0.065 (1.00)**
RF	0.286 (11.97)	0.319 (5.85)	0.504 (3.36)	0.361 (6.21)
MARS	0.146 (6.28)	0.354 (6.72)	1.027 (6.91)	0.417 (7.19)

Because of the sparsity and additivity in the data, all Lasso-like methods do better than RF, a pattern that continues in all of these simulation studies. Indeed, due to the random selection of covariates in the RF algorithm, the presence of spurious covariates seems to produce a phenomenon of excess shrinkage, which can be clearly see in plots of fitted values versus response values. Using the scaling corrections provided in the R implementation improves things, but not to a great extent. MARS, similarly, has difficulty in finding the correct variables. With such large p, the set of possible hockey stick bases MARS has to search through is very large, and hence the underlying greedy stepwise selection component of the algorithm is in general unsuccessful at handling this problem.

Amongst the Lasso-like methods, perhaps unsurprisingly, the SSP method performs by far the best, owing to the large degree of smoothness in the true model function. Liso-Adaptive is second best, however, beating SpAM even though it does

not have an internal smoothing effect. The basic Liso method itself underperforms, perhaps because it does not strongly enforce sparsity amongst the original covariates.

Unexpectedly, Liso-SCAD performs fairly equivalently to the Liso itself in this and all following simulations. A likely explanation is that for sufficient regularisation to take place to take spurious covariates to zero, the penalty function is such that the solution lies mostly on the part of the penalty where it is identical to the original total variation penalty.

The introduction of a moderate amount of correlation does not greatly affect the performance of any of the algorithms.

4.4.2.2 Mixed powers

In this case, the response has a more complex relation to the covariates:

$$
\begin{aligned}
Y_i &= \sum_{k=1}^{5} f_k(X_i^{(a_k)}) + \sigma\varepsilon_i \\
f_1(x) &= \operatorname{sign}(x + C_1)\,|x + C_1|^{0.2} \\
f_2(x) &= \operatorname{sign}(x + C_2)\,|x + C_2|^{0.3} \\
f_3(x) &= \operatorname{sign}(x + C_3)\,|x + C_3|^{0.4} \\
f_4(x) &= \operatorname{sign}(x + C_4)\,|x + C_4|^{0.8} \\
f_5(x) &= x + C_5
\end{aligned}
$$

In this case, we have again $n = 200, p = 50$. $C_1, \ldots, C_5$ are small shifts, randomly generated as $\mathit{Uniform}(-1/4, 1/4)$, and $a_1, \ldots, a_5$ are covariates randomly chosen without replacement. In the test set, the variance of the response was approximately 2.6.

Algorithm	**SNR = 7**	**SNR = 3**	**SNR = 1**	**SNR = 3, Cor**
Liso	0.128 (1.49)	0.230 (1.50)	0.459 (1.41)	0.255 (1.50)
Liso-Adaptive	**0.088 (1.01)**	**0.160 (1.00)**	**0.352 (1.06)**	**0.177 (1.01)**
Liso-SCAD	0.128 (1.49)	0.229 (1.49)	0.587 (1.82)	0.254 (1.50)
SpAM	0.157 (1.83)	0.267 (1.75)	0.539 (1.68)	0.285 (1.69)
SSP	0.126 (1.47)	0.226 (1.49)	0.429 (1.33)	0.252 (1.51)
RF	0.358 (4.21)	0.450 (2.96)	0.721 (2.26)	0.495 (2.96)
MARS	0.319 (3.78)	0.678 (4.54)	1.936 (6.32)	0.783 (4.71)

With the new, non-linear model function, the Liso and Liso-SCAD now perform equally as well as the SSP, while the Adaptive Liso performs significantly better, being the best in almost all runs. All four methods outperform SpAM, and greatly outperform RF and MARS.

In this case, the explanation is that for fractional powers, the component functions are relatively flat in the extremes of the covariate range, with most of the variation occurring in the middle of the range. SpAM and SSP are unable to capture the sharp transition point of the small root functions without introducing inappropriate variability at the ends of the fit, and hence both perform significantly worse than previously. The Liso based methods, however, do not explicitly smooth the fit and only threshold the extremes. Being thus adapted to this sort of function, they actually improve their performance in proportional terms relative to the variance of the test set.

4.4.2.3 Mixed powers, large p

In this scenario, our model is the same as before, save that we have many more spurious covariates, resulting in $n = 200, p = 200$. The variance of the test response is unchanged at approximately 2.6.

Algorithm	**SNR = 7**	**SNR = 3**	**SNR = 1**	**SNR = 3, Cor**
Liso	0.166 (1.89)	0.283 (1.86)	0.638 (1.78)	0.286 (1.84)
Liso-Adaptive	**0.090 (1.00)**	**0.156 (1.00)**	**0.384 (1.01)**	**0.160 (1.01)**
Liso-SCAD	0.169 (1.93)	0.292 (1.91)	0.935 (2.71)	0.296 (1.90)
SpAM	0.201 (2.32)	0.329 (2.17)	0.779 (2.21)	0.331 (2.14)
SSP	0.156 (1.78)	0.274 (1.80)	0.604 (1.73)	0.274 (1.78)
RF	0.504 (5.86)	0.588 (3.86)	0.992 (2.84)	0.593 (3.84)
MARS	0.805 (9.27)	1.704 (11.49)	4.707 (13.84)	1.763 (11.60)

Due to the effect of high dimensionality, all algorithms see their performance decline significantly - except the Adaptive Liso, which has an increased MSE of less than 3% in the low noise case. This is due to the Adaptive reweighting, which retains a very sparse fit, picking the relevant variables even as the number of predictors grows. By eliminating the additional spurious covariates, a successful estimate is made on the remainder.

Chapter 5

The Co-adaptive Lasso

Following on from the heuristics and empirical evidence shown in Chapter 4, in this chapter we define rigorously a method we call the Co-adaptive Lasso based on those principles, and establish theoretical properties and comparisons.

5.1 Notation and Definitions

As before, let $Y \in \mathbb{R}^n$ be the response vector. Assume, subtracting by a constant intercept term if necessary, that $\sum_{i=1}^n Y_i = 0$. $X \in \mathbb{R}^{n \times p}$ is the matrix composed of covariate column vectors, which we assume also to have mean zero. Hence, n is the sample size, and p is the number of covariates.

Now, assume that in our problem, the covariates have an a-priori known group structure, which defines the patterns of group sparsity that we intend to achieve. We specify this by

$$\mathcal{G} = \{G_j\}_{j=1}^q, \quad \text{with} \quad \bigcup_{j=1}^q G_j = \{1, \ldots, p\},$$

which defines the membership indicators of each group. We say that $\mathcal{G}$ is non-overlapping if its elements are all disjoint.

For any set S, we denote by $\mathcal{G}_{\cap S} = \{G \in \mathcal{G} : G \cap S \neq \emptyset\}$, and $\mathcal{G}^c_{\cap S} = \{G \in \mathcal{G} : G \cap S = \emptyset\}$: the groups that do, or do not intersect with that set. We write $\widetilde{S}$ to be S together with its in-group neighbours - that is, $\widetilde{S} = \cup \mathcal{G}_{\cap S}$.

For $S = \{j \in \{1, \ldots, p\} : \beta_j \neq 0\}$, S is group sparse if $\mathcal{G}_{\cap S}$ is a small subset of $\mathcal{G}$. We say the group structure is evenly sized if each $G \in \mathcal{G}$ have the same size, and the

problem is all-in-all-out (AIAO) if S can be expressed exactly as an union of sets in $\mathcal{G}$.

Recall that for a tuning parameter $\lambda > 0$, the Lasso [Tibshirani 1996] estimate is defined as

$$\widehat{\beta}_{\lambda}^{(l)} = \arg\min_{\beta} \frac{1}{2} \|Y - X\beta\|_n^2 + \lambda \|\beta\|_1 . \tag{5.1.1}$$

As established in Section 4.1.3, we are interested in adapting the Lasso to handle the case of group sparsity and sparsity within groups. We shall now propose a modification that uses the reweighting technique suggested in Section 4.2.2.

Definition 5.1.1. Let $\mu > 0$. Suppose $\widehat{\beta}^{(l)}$ is a Lasso solution for X, Y, for some appropriately chosen tuning parameter value λ. Then, for $\mathcal{G}$ non-overlapping, we define the Co-adaptive Lasso weights, for a given covariate $j \in G$, as

$$w_j = \sqrt{|G|} \left\| \widehat{\beta}_G^{(l)} \right\|^{-1} . \tag{5.1.2}$$

The Co-adaptive Lasso solution is then

$$\widehat{\beta}_{\mu}^{(c)} = \arg\min_{\beta} \frac{1}{2} \|Y - X\beta\|_n^2 + \mu \sum_{j=1}^{p} w_j |\beta_j|. \tag{5.1.3}$$

Remark 5.1.1. A different weighting scheme is possible, using the L1 norm within groups, producing

$$w_j = |G| \left\| \widehat{\beta}_G^{(l)} \right\|_1^{-1} .$$

This L1 norm formulation follows more directly from the previous chapter, being the LLA computation of the grouped adaptive penalty of Equation (4.1.2). In the version in this section, we opt for the L2 norm weights, because this produces conditions that are more similar to those conditions in the Lasso and Group Lasso that are used to prove L2 estimation and prediction error properties. Replacing the condition 'A2' later defined in this chapter, with a condition more similar to the compatibility condition of van de Geer and Bühlmann [2009], similar convergence properties can be shown for the above weighting.

Note that with the weighting in (5.1.2), we are no longer making a LLA computation, because w_j does not represent the derivative of any function. This can be seen

by the fact that for $i, j \in G$,

$$\frac{dw_j}{d\widehat{\beta}_i^{(l)}} = -\frac{\sqrt{|G|}\widehat{\beta}_i^{(l)}}{\left\|\widehat{\beta}_A\right\|^3} \neq \frac{dw_i}{d\widehat{\beta}_j^{(l)}}.$$

Nevertheless, we shall show good results for this method.

Remark 5.1.2. In either reweighting scheme, observe that if each group contains only one covariate, then the weight calculations we have here is identical to the standard Interated Lasso implementation of the Adaptive Lasso [Zou 2006][Huang et al. 2008]. A similarity also exists to the Group Bridge methodology of Breheny and Huang [2009]. The authors there similarly use a combined concave penalty, but in that paper, they focus on iterating until convergence using a Local Coordinate Descent algorithm with a fixed choice of λ. With this process, they are able to show that the algorithm achieves in its stage-by-stage estimates monotone convergence in terms of the penalised likelihood criterion $\ell_\lambda(\widehat{\beta}^{(m)})$, but do not present any theoretical performance results.

Note that we have restricted our consideration to non-overlapping groups. In the case of overlapping groups, a range of different weight formulations may be considered, depending on the type of overlap and signal sparsity pattern. We discuss this later in Section 5.5.

As before, for clarity, we will suppress the subscripts λ and μ in our notation for $\widehat{\beta}^{(c)}$.

5.1.1 Restricted eigenvalues and the Lasso

Key to the performance of the Lasso and most variants of it are conditions on the covariance matrix - in particular its restricted eigenvalue, or compatibility properties [van de Geer and Bühlmann 2009]. Broadly speaking, these properties measure the minimal eigenvalues or generalised eigenvalues of the matrix under some set of restrictions. Failure of the relevant condition implies that there exists feasible alternative solutions that give the same fitted values, thus implying the failure of the algorithm.

In the Lasso case, given some observed training covariate dataset X, we define the restricted eigenvalue $\phi^2(L, S, m)$ as the function,

$$\phi^2(L,S,m) = \min_{\delta,M}\left(\frac{\|X\delta\|_n^2}{\|\delta_M\|^2} : M \supset S, |M| \leq m, \|\delta_{S^c}\|_1 \leq L\sqrt{|S|}\,\|\delta_S\|\right), \qquad (5.1.4)$$

with $\phi^2(L,S) = \phi^2(L,S,|S|)$.

The restricted eigenvalue therefore describes the minimal generalised eigenvalue of X, restricted to sparse subsets that contain some fixed target set S, and which satisfy a certain L1-L2 norm ratio. Low restricted eigenvalue is generally characteristic of the presence of closely correlated covariates. Note that given the form of the above, M must contain the largest values of δ_{S^c}. Therefore, while in van de Geer and Bühlmann [2009] and van de Geer et al. [2010], a slightly different form is given, the above is equivalently applicable. A further variant is available in Bickel et al. [2009]. We say the $RE(L,S,m)$ condition holds if $\phi^2(L,S,m) \geq 0$.

The usual results for the standard Lasso in this case are

Lemma 5.1.1. *Let*

$$\lambda > 2\max|X'\varepsilon/n|.$$

Then the Lasso estimate $\widehat{\beta}_\lambda$ satisfies

$$\left\|X\widehat{\beta}_\lambda - X\beta\right\|_n^2 \leq \frac{14\lambda^2|S|}{\phi^2(6,S)},$$

$$\left\|\widehat{\beta}_{\lambda S} - \beta_S\right\| \leq \frac{7\lambda\sqrt{|S|}}{\phi^2(6,S)}.$$

Further,

$$\left\|\widehat{\beta}_\lambda - \beta\right\| \leq \frac{28\lambda\sqrt{|S|}}{\phi^2(6,S,2|S|)}.$$

A proof of this is available in Theorem 7.1 of van de Geer et al. [2010]. Similar theorems have been proven by other authors, including Bickel et al. [2009]. Use of compatibility conditions [van de Geer and Bühlmann 2009] will yield an L1 and prediction error convergence result under similar conditions. The choice $L = 6$ in the instances of ϕ can usually be replaced with other values of $L > 1$, at the price of changing the constants in the bounds.

The Adaptive Lasso in general uses the same conditions. In the Group Lasso, a similar RE property is required, but which uses the group L2 norms instead of the

L1 norm in the restriction, and furthermore restricts the considered sets M to be combinations of groups. This second point is one of the reasons that the required conditions for the Group Lasso are somewhat weaker than for the Lasso in the case of L2 estimation. Each produces different performance bounds, as given in, for example van de Geer et al. [2010] and Huang and Zhang [2010].

5.1.2 Conditions for the Co-adaptive Lasso

For our work, we introduce an additional variant of the RE property.

Definition 5.1.2. We define the group restricted eigenvalue (GRE) statistic as

$$\phi^2_{\mathcal{G}}(L,S) := \min_{\delta,M} \left(\frac{\|X\delta\|_n^2}{\|\delta_M\|^2} : M \in \mathcal{G}, \|\delta_{S^c}\|_1 \leq L\sqrt{|S|}\,\|\delta_S\| \right).$$

Compared to Equation (5.1.4), we have again a similar restricted eigenvalue computation. However, instead of a general minimisation over sets $M \supset S$ of specified size, the M considered in the GRE are only those groups in our group specification $\mathcal{G}$, which may or may not contain S. It follows that we have the following set of inequalities relating ϕ_G and ϕ:

Lemma 5.1.2. *Let $\mathcal{H} \subset \mathcal{G}_{\cap S}$ be any covering set of S. We have inequalities:*

$$|\mathcal{H}|\phi^2(L,S) \geq \phi^2_{\mathcal{G}}(L,S) \geq \phi^2(L,S,|S| + \max_{G\in\mathcal{G}}|G|) \geq \phi^2(L,S)/(1+L^2|S|).$$

This Lemma supplies an idea of the types of values that might be expected. However, the second inequality, $\phi^2_{\mathcal{G}}(L,S) \geq \phi^2(L,S,|S| + \max_{G\in\mathcal{G}}|G|)$, can be very loose if $|G|$ is large, because we consider a much restricted set of subsets. In addition, presence of a few small groups may also not matter as much as it appears above, because these groups may not coincide with directions where δ can be large without increasing greatly $\|X\delta\|$. In particular, $\phi^2_{\mathcal{G}}$ is now comparable with the Group Lasso version of the RE property. We note that the form of our inequalities are distinguished from the form in van de Geer et al. [2010] in that $\phi^2(L,S,m) > 0$ if and only if $\phi^2(L,S) > 0$. Note that in practice, RE type values are known to be often difficult to compute – we leave this as the subject of potential further research.

For these properties to translate into the bounds we later prove, we require a series of conditions, both in terms of restricted eigenvalue properties, and in terms of

scaling speeds of various aspects of the problem, for asymptotics. These conditions are defined below, with their effects illustrated in Figure 5.1.1.

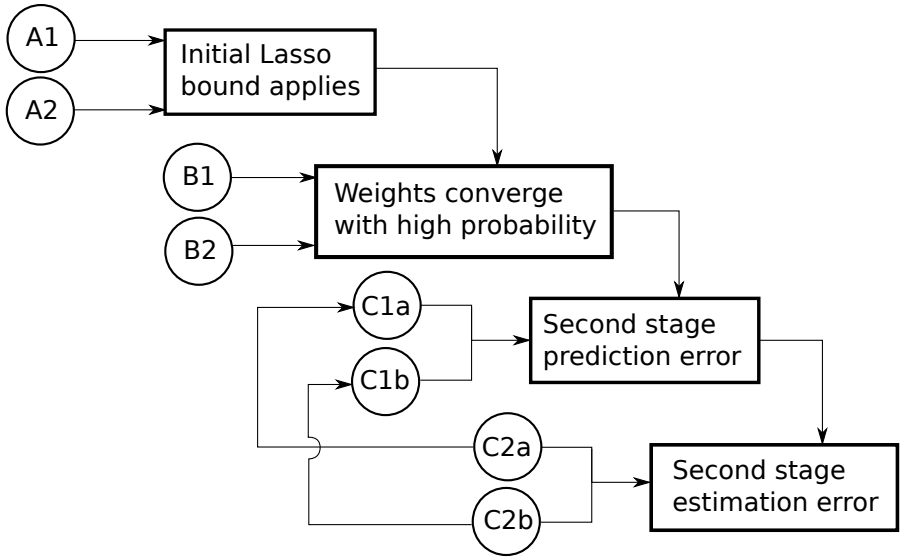

Figure 5.1.1: Conditions for the Co-adaptive Lasso. The arrows represent implications, with all arrows into a box required for satisfaction of sufficient conditions.

Definition 5.1.3. The group restricted RE properties lead to the definition of the following conditions:

Condition A1: RE conditions for initial Lasso

There exists $C > 0$ such that for all sufficiently large n,

$$\phi^2(3, S) > C.$$

Condition A2: Group RE conditions for initial Lasso

There exists $C > 0$ such that for all sufficiently large n,

$$\phi^2_{\mathcal{G}}(3, S) > C.$$

Condition B1: Normal noise, bounded variance

The noise ε is independent normal, with variance less than σ^2. $\|X_{\cdot,j}\|_n$ is bounded for all j. Without loss of generality, assume, rescaling if necessary, that the $\|X_{\cdot,j}\|_n$ is identically equal to 1.

Condition B2: Scaling conditions

There exists $\gamma_1 \geq 0, \gamma_2 \geq 0$ such that

$$\max_G \frac{\sigma^2 |S| \log(p) |G|^{\gamma_1}}{n \max(\|\beta_G\|^2, \min_{H \in \mathcal{G}_{\cap S}} \|\beta_H\|^2)} = o\left(n^{-\gamma_2}\right).$$

Under this assumption, write

$$L_1^0 = \max_{\substack{G \in \mathcal{G}_{\cap S}, \\ H \in \mathcal{G}_{\cap S}^c}} \sqrt{\frac{|G|}{|H|^{1+\gamma_1} n^{\gamma_2}}}.$$

Condition C1: RE conditions for second stage, prediction error
(a): [AIAO version] Given condition B2, there exists $C > 0$, $\delta > 0$ such that

$$\phi^2\left(\delta L_1^0, \widetilde{S}\right) > C.$$

(b): [Within-group version] There exists $C > 0$, $\delta > 0$ such that there exists T, $S \subseteq T \subseteq \widetilde{S}$ satisfying

$$\phi^2\left(\max\left(\delta L_1^0, \max_{\substack{G \in \mathcal{G}_{\cap S}, \\ H \in \mathcal{G}_{\cap(\widetilde{S} \setminus T)}}} 2(1+\delta) \frac{\|\beta_H\| / \sqrt{|H|}}{\|\beta_G\| / \sqrt{|G|}}\right), T\right) > C.$$

Condition C2: RE conditions for second stage, estimation error
(a): [AIAO version] Given B2, there exists $C > 0$, $\delta > 0$, $T \supset S$, such that:

$$\phi^2\left(\delta L_1^0, \widetilde{S}, |\widetilde{S}| + |T|\right) > C.$$

(b): [Within-group version] There exists $C > 0$, $\delta > 0$ such that there exists T, $S \subseteq T \subseteq \widetilde{S}$ satisfying

$$\phi^2\left(\max\left(\delta L_1^0, \max_{\substack{G \in \mathcal{G}_{\cap S}, \\ H \in \mathcal{G}_{\cap(\widetilde{S} \setminus T)}}} 2(1+\delta) \frac{\|\beta_H\| / \sqrt{|H|}}{\|\beta_G\| / \sqrt{|G|}}\right), T, 2|T|\right) > C.$$

Remark 5.1.3. Conditions A1 and A2 are restricted eigenvalue type conditions to ensure the Lasso performs sufficiently well in the first stage. They generally hold so long as the covariates are not too highly correlated, and $|S|$ and p are not too large

relative to n. Generally, these RE and GRE conditions are a bit weaker than the relevant Restricted Eigenvalue conditions required for the Lasso.

Conditions B1 and B2 govern the noise level and scaling of the various dimensions of the problem. B1 is a standard error assumption common in most analyses of such problems, and the normality assumption can be trivially relaxed to any general sub-Gaussian noise distribution. Meanwhile, B2 relates to the way the initial Lasso's error scales relative to the group-wise weights using the true β. It holds, for example, in any case where the Lasso itself converges and $\min \|\beta_G\|$ does not decrease. In the case where, in addition, the expected sizes of the individual coefficients remain constant as $|G|$ increases, B2 holds with $\gamma_1 \geq 1$, $\gamma_2 > 0$. Faster convergence of the initial Lasso grants greater lee-way in how $\|\beta_G\|$ might be small with increasing n. Note that many bounds in this article may hold even if B2 fails. However, the bounds may no longer be useful.

The A and B conditions together imply asymptotic convergence in the weights w.

C1 and C2 are restricted eigenvalue type conditions for the second stage. They are similar to those in A1-A2. but include allowances for, on the plus side, the first stage's success in removing irrelevant groups, and on the negative side, differences between the groups that make it difficult to identify relevant or irrelevant variables. The (a) versions are most useful in cases where there is little within-group sparsity; while the (b) versions allow success when the group sizes are too large for C1(a) to be satisfied, as long as there is substantial within-group sparsity.

Because L_1^0 can be quite small, C1 and C2 are typically not very strong conditions, and hold fairly often in cases where the Lasso or the Adaptive Lasso succeed. However, more research is required to be sure of this.

Remark 5.1.4. Relations exist between the conditions. Condition C2(a) implies C1(a), while correspondingly C2(b) implies C1(b), and both require B2 to be true to be meaningful. By Lemma 5.1.2, if $\min\{|\mathcal{H}| : \cup\mathcal{H} \supset S\}$ remains bounded, then A2 implies A1. Further, if $RE(3, S, |S| + \max|G|)$ is satisfied, then both conditions A1 and A2 are satisfied. The main theorems in this article require the satisfaction of all the A and B conditions, plus one of conditions C1 and one of the conditions C2.

Conditions C1(a) and C2(a) simplifies in the case where the groups are evenly sized, in which case we require only that $\phi^2(L, \widetilde{S})$ is bounded for some fixed L for this to eventually automatically be fulfilled. Indeed, in this evenly sized case, if $\phi^2(3, \widetilde{S}, 2|\widetilde{S}|)$ is bounded, then conditions C1(a) and C2(a) and A1-2 are satisfied, since $\max|G| \leq |\widetilde{S}|$. In the evenly sized groups, AIAO context, then, satisfaction

of the RE conditions for the Lasso implies the RE and GRE conditions for the Co-adaptive Lasso are satisfied.

In this case of evenly sized groups, condition C1(b) and C2(b) become dependent on the ratio $\|\beta_H\| / \|\beta_G\|$, for $G \in \mathcal{G}_{\cap S}, H \in \mathcal{G}_{\cap(\tilde{S}\setminus T)}$. Noting that this ratio is always greater or equal to 1, satisfying the condition becomes a trade-off between choosing T large enough so that this ratio is kept small, and small enough so that restricted eigenvalues do not fall too low. Because of the role of $|T|$ in our theorems, condition C2 is most useful when a T can be found with size $|T| = O(|S|)$.

5.2 Main results

From a heuristic point of view, due to the re-weighting nature of the algorithm, the Co-adaptive Lasso necessarily inherits some of the properties of the Adaptive Lasso, as described in van de Geer et al. [2010]. Whenever the conditions necessary for the the Adaptive Lasso are fulfilled, for example, the adaptive weights must successfully distinguish between the zero and the non-zero coefficients, with the weights on the non-zero coefficients converging to a negligible fraction of the weights on the zero coefficients. In this instance, the co-adaptive weights, as an aggregation of adaptive weights within each group, must also successfully distinguish between zero groups $\mathcal{G}^c_{\cap S}$ and non-zero groups $\mathcal{G}_{\cap S}$. Hence the second Lasso step converges to a Lasso performed on the subset of covariates that belong to these non-zero groups, instead of the whole p-dimensional dataset. In the asymptotic setting of the sample size increasing to infinity, taking $\mu \to 0$, we obtain results in the second stage similar to an ordinary linear regression conducted only on the relevant covariate groups, implying that the Co-adaptive Lasso is consistent for AIAO sparsity in the cases where the Adaptive Lasso succeeds.

However, there may be situations where the Co-adaptive Lasso succeeds though the Adaptive Lasso does not, or at least attains a better performance.

The following theorem gives some asymptotic bounds:

Theorem 5.2.1. *Suppose conditions A1-2 and B1-2 are satisfied. Suppose in addition $\mathcal{G}$ is non-overlapping.*

If for some T with $S \subseteq T \subseteq \widetilde{S}$, either C1(a) or C1(b) is satisfied, then, writing

$$\widetilde{\mu} = \max\left(\sqrt{\log(p)}L_1^0, \sqrt{\log|\widetilde{S}|}\max_{\substack{G\in\mathcal{G}_{\cap S},\\ H\in\mathcal{G}_{\cap(\widetilde{S}\setminus T)}}}\frac{\|\beta_H\|\sqrt{|G|}}{\|\beta_G\|\sqrt{|H|}}\right)$$

$$\widetilde{L} = \max\left(L_1^0, \max_{\substack{G\in\mathcal{G}_{\cap S},\\ H\in\mathcal{G}_{\cap(\widetilde{S}\setminus T)}}}\frac{\|\beta_H\|\sqrt{|G|}}{\|\beta_G\|\sqrt{|H|}}\right),$$

taking the latter part of the maximisation equal to 0 if $\widetilde{S}\setminus T$ is empty, we have that for any $\eta > 0$, there exists λ, μ such that

$$\left\|X\left(\widehat{\beta}^{OLS,T} - \widehat{\beta}^{(c)}\right)\right\|_n^2 = O\left(\sigma^2\frac{|T|}{n}\widetilde{\mu}^2\right) \tag{5.2.1}$$

$$\left\|X\left(\beta - \widehat{\beta}^{(c)}\right)\right\|_n^2 = O\left(\sigma^2\frac{|T|}{n}\left(1+\widetilde{\mu}\right)^2\right) \tag{5.2.2}$$

with probability exceeding $1-\eta$. Here $\widehat{\beta}^{OLS,T}$ represents the ordinary least squares solution restricted to the covariate set T.

Further, if either condition C2(a) or C2(b) holds, simultaneously

$$\left\|\widehat{\beta}^{OLS,T} - \widehat{\beta}^{(c)}\right\|^2 = O\left(\sigma^2\frac{|T|}{n}\widetilde{\mu}^2\left(1+\widetilde{L}\right)^2\right) \tag{5.2.3}$$

$$\left\|\beta - \widehat{\beta}^{(c)}\right\|^2 = O\left(\sigma^2\frac{|T|}{n}\left(1+\widetilde{\mu}\left(1+\widetilde{L}\right)\right)^2\right). \tag{5.2.4}$$

In short, so long as the initial Lasso can be guaranteed to not perform too badly, making allowances for a limited set of variables $T\setminus S$ where it is difficult to distinguish relevant from irrelevant, the Co-adaptive Lasso performs within a multiplicative factor of the optimal result, with this factor being dependent on variability in group sizes and coefficient group norms.

This asymptotic theorem derives from some finite sample lemmas that are more technically involved. These we leave for the appendix.

In some common situations, Theorem 5.2.1 can be simplified greatly with some additional assumptions.

Corollary 5.2.2. *Suppose that $\mathcal{G}$ is non-overlapping, and conditions A1-2 and B1-2 are satisfied. Suppose additionally that*

$$\max_{H \in \mathcal{G}^c_{\cap S}} |H| = O\left(\min_{G \in \mathcal{G}_{\cap S}} |G|\right)$$

and there exist $T \supseteq S$ with $|T| = O(|S|)$ so that C1(a) or C1(b) is satisfied and

$$\max_{H \in \mathcal{G}_{\cap (\widetilde{S} \setminus T)}} \|\beta_H\| / |H| = O\left(\min_{G \in \mathcal{G}_{\cap S}} \|\beta_G\| / |G|\right).$$

Then for any $\eta > 0$, there exists λ, μ such that for any G,

$$\left\|X\left(\beta - \widehat{\beta}^{(c)}\right)\right\|_n^2 = O\left(\sigma^2 \frac{|S|}{n} \max\left(\frac{\log|\mathcal{G}|}{|G|^{\gamma_1} n^{\gamma_2}}, \log|\widetilde{S}|\right)\right),$$

with probability exceeding $1 - \eta$.

If C2(a) or C2(b) is satisfied then simultaneously

$$\left\|\beta - \widehat{\beta}^{(c)}\right\|^2 = O\left(\sigma^2 \frac{|S|}{n} \max\left(\frac{\log|\mathcal{G}|}{|G|^{\gamma_1} n^{\gamma_2}}, \log|\widetilde{S}|\right)\right).$$

Corollary 5.2.3. *Suppose that $\max_{H \in \mathcal{G}^c_{\cap S}} |H| = O\left(\min_{G \in \mathcal{G}_{\cap S}} |G|\right)$, and $\mathcal{G}$ is non-overlapping. Then if conditions B1-2 and A2 are satisfied and A1 satisfied replacing S with $\widetilde{S}$, then, for any $\eta > 0$, there exist λ, μ such that for any G,*

$$\left\|X\left(\beta - \widehat{\beta}^{(c)}\right)\right\|_n^2 = O\left(\sigma^2 \frac{|\widetilde{S}|}{n} (1 + \sqrt{\log|\mathcal{G}| |G|^{-\gamma_1} n^{-\gamma_2}})^2\right),$$

with probability exceeding $1 - \eta$.

If in addition there exists fixed L, C such that $\phi(L, \widetilde{S}, 2|\widetilde{S}|) > C$, then it is simultaneously the case that

$$\left\|\beta - \widehat{\beta}^{(c)}\right\|^2 = O\left(\sigma^2 \frac{|\widetilde{S}|}{n} (1 + \sqrt{\log|\mathcal{G}| |G|^{-\gamma_1} n^{-\gamma_2}})^2\right).$$

In particular, Corollary 5.2.3, in the case of AIAO signals, requires conditions that are the same or weaker than those of the ordinary Lasso or Adaptive Lasso.

We stress that these bounds in general only provide worst case results. In contrast to the Lasso, where the penalisation also provides a tight lower bound on the prediction and estimation errors [Huang and Zhang 2010], in realistic cases, the distribution of errors amongst the irrelevant groups means that the weight calculations can be, and usually will be, much better than Theorem 5.2.1 suggests. Since the Lasso selects a maximum of $\min(n, p)$ variables, a simple calculation will show that in the best case, under the conditions of Corollary 5.2.3, we spread the error in the initial estimate across $\min(n, |\mathcal{G}|)$ groups, resulting in

$$\left\|X(\beta - \widehat{\beta}^{(c)})\right\|_n^2 = O\left(\sigma^2 \frac{|S|}{n}(1 + \sqrt{\log |\mathcal{G}||G|^{-\gamma_1} n^{-\gamma_2} / \min(n, |\mathcal{G}|)})^2\right),$$

with similar results for the other inequalities.

5.3 Comparisons

Let us compare the above bounds to performance bounds for some related methods. In the following, we focus on the prediction error rate $\left\|X(\beta - \widehat{\beta}^{(c)})\right\|_n^2$. We note that in general similar results can be obtained for the estimation error. Note that we are comparing upper bounds for each method, whereas strictly speaking min-max type results are preferred to show one method is clearly better than another. Our analysis in this chapter is therefore more illustrative of potential advantages, and not completely definitive.

5.3.1 Comparison to the Lasso

Now, from van de Geer and Bühlmann [2009], the standard Lasso has a prediction error on the order of

$$\left\|X(\beta - \widehat{\beta}^{(l)})\right\|_n^2 = O\left(\sigma^2 \frac{|S|}{n} \log(p)\right).$$

Under the conditions of Corollary 5.2.3, however, the Co-adaptive Lasso has a

prediction error of

$$\left\|X(\beta - \widehat{\beta}^{(c)})\right\|_n^2 = O\left(\sigma^2 \frac{|\widetilde{S}|}{n}\left(1 + \sqrt{\log|\mathcal{G}||G|^{-\gamma_1} n^{-\gamma_2}}\right)^2\right).$$

Hence, when the conditions are satisfied, the Co-adaptive Lasso can outperform the ordinary Lasso, assuming that the group or sample size is large, and the grouping structure is meaningful in that $\widetilde{S}$ is kept small. Indeed, in the AIAO case, if $n \to \infty$, then the Co-adaptive Lasso attains the oracle rate, removing the contribution from p. Examination of the Irrepresentable Condition [Zhao and Yu 2006] indicates the Co-adaptive Lasso should be consistent for variable selection under much weaker design conditions than the Lasso.

If on the other hand the conditions of Corollary 5.2.2 are satisfied, and $|\mathcal{G}|$ does not grow exponentially in $|G|n$, then

$$\left\|X(\beta - \widehat{\beta}^{(c)})\right\|_n^2 = O\left(\sigma^2 \frac{|S|}{n} \log|\widetilde{S}|\right).$$

In effect, the co-adaptive re-weighting has successfully screened out all of the variables that do not belong in the same group as the relevant variables. Since often $|\widetilde{S}| = O(n)$, while p can be anything up to exponential in n, this can be a great improvement. Indeed, if $|\widetilde{S}| = O(n)$, we arrive at bounds asymptotically within a constant factor of the oracle rate.

On the other hand, if condition B2 cannot be satisfied, or $\widetilde{S}$ is too large relative to S, the Co-adaptive Lasso can under-perform. In empirical experiments, though, we see that the Co-adaptive Lasso often outperforms the Lasso even with randomly chosen groups - after all, for small enough groups, the Co-adaptive Lasso acts similarly to the Adaptive Lasso, which can have performance advantages.

5.3.2 Comparison to the Adaptive Lasso

The usefulness of Theorem 5.2.1 lies in the dependence on $(\min_{G \in \mathcal{G}_{\cap S}} \|\beta_G\|)$ implied in condition B2. In contrast, an Adaptive Lasso approach, being equivalent to a Co-adaptive Lasso with group size 1, would use $\min|\beta_S|$ instead. Suppose that the average squared β amongst the true S remains constant or at least bounded below. Then as group sizes increase, $(\min_{G \in \mathcal{G}_{\cap S}} \|\beta_G\|) = \Omega(\min_G \sqrt{|G|})$, satisfying condition B2 with $\gamma_1 = 1$. Meanwhile, in many set-ups, the minimum $\min|\beta_S|$ would not

increase, but indeed decrease, and hence if the initial Lasso gives errors that are larger than this, performance guarantees cannot be given. In terms of γ_2, if the Adaptive Lasso satisfies B2 for any particular γ_2, it is implied that the Co-adaptive Lasso must also satisfy it for that γ_2. Similar results arise if we focus on a harmonic mean based formulation that gives slightly tighter bounds. This means that the conditions for the Adaptive Lasso are typically more stringent, because the weight averaging implicit in our formulation helps us to deal with this coefficient variability while the adaptive weights cannot.

On the other hand, if there is too much within group sparsity, the Adaptive Lasso may be superior, because the Co-adaptive Lasso fails to discriminate as strongly between relevant and irrelevant variables within groups.

Ignoring the benefit with covariate coefficient variability, consider a case with evenly sized non-overlapping groups, constant coefficients amongst the β_S, and a constant within-group sparsity amongst $\mathcal{G}_{\cap S}$. Let β_S, $|G|$, $|\text{Supp}(\beta_G)|$ for $G \in \mathcal{G}_{\cap S}$, and $\sigma^2|S|\log(p)/n$ each scale according to some power of n:

$$\begin{aligned} |\beta_S| &= \Theta\left(n^{-\gamma_\beta}\right) \\ |G| &= \Theta\left(n^{\gamma_g}\right) \\ |\text{Supp}(\beta_G)| &= \Theta\left(n^{\gamma_s}\right) \\ \sigma^2|S|\log(p)/n &= \Theta\left(n^{-\gamma_l}\right). \end{aligned}$$

Assuming all the conditions of Lemma 5.2.2 are satisfied for both the Co-adaptive Lasso and the Adaptive Lasso. Then the Adaptive Lasso, as the $|G| = 1$ special case, attains

$$\left\|X\left(\beta - \widehat{\beta}^{(a)}\right)\right\|_n^2 = O\left(\sigma^2 \frac{|S|}{n} \max\left(\frac{\log(p)}{n^{\gamma_l - 2\gamma_\beta}}, \log|S|\right)\right).$$

In contrast, the Co-adaptive Lasso attains

$$\left\|X\left(\beta - \widehat{\beta}^{(c)}\right)\right\|_n^2 = O\left(\sigma^2 \frac{|S|}{n} \max\left(\frac{\log(p) - \gamma_g \log(n)}{n^{\gamma_l - 2\gamma_\beta + \gamma_s}}, \log|S| + (\gamma_g - \gamma_s)\log(n)\right)\right).$$

Now, if $\log(p)/n^{\gamma_l - 2\gamma_\beta} \to 0$, then the Adaptive Lasso obtains a near optimal rate, and we cannot expect the Co-adaptive Lasso to exceed it in the asymptotic framework

– though for small sample results the Co-adaptive Lasso can still do better. Otherwise, the Co-adaptive Lasso can do substantially better if γ_β or γ_g are large, so long as γ_s is not too small relative to γ_g.

As a particular case, if the number of non-zero variables in each group G is a fixed proportion of the full group size, the Co-adaptive Lasso in this situation will always at least match the Adaptive Lasso, and do better if $\gamma_g > 0$.

5.3.3 Comparison to the Group Lasso

From Lounici et al. [2010], it can be inferred that the Group Lasso produces prediction error bounds of the form

$$\begin{aligned}\left\|X(\beta - \widehat{\beta}^{(g)})\right\|_n^2 &= O\left(\frac{\sigma^2}{n}\sum_{G\in\mathcal{G}_{\cap S}}\left(|G| + \sqrt{|G|\log(p/|G|)} + \log(p/|G|)\right)\right) \\ &= O\left(\frac{\sigma^2}{n}\left(|\widetilde{S}| + |\mathcal{G}_{\cap S}|\sqrt{|G|\log|\mathcal{G}|} + |\mathcal{G}_{\cap S}|\log|\mathcal{G}|\right)\right),\end{aligned}$$

in the case of non-overlapping, evenly sized groups.

Suppose Corollary 5.2.3's conditions are satisfied. Then the Co-adaptive Lasso attains

$$\begin{aligned}\left\|X\left(\beta - \widehat{\beta}^{(c)}\right)\right\|_n^2 &= O\left(\sigma^2\frac{|\widetilde{S}|}{n}(1 + \sqrt{\log|\mathcal{G}||G|^{-\gamma_1}n^{-\gamma_2}})^2\right) \\ &= O\left(\sigma^2\frac{|\widetilde{S}|}{n}(1 + 2\sqrt{\log|\mathcal{G}||G|^{-\gamma_1}n^{-\gamma_2}} + \log|\mathcal{G}||G|^{-\gamma_1}n^{-\gamma_2})\right) \\ &= O\left(\frac{\sigma^2}{n}\left(|\widetilde{S}| + |\mathcal{G}_{\cap S}|\sqrt{\frac{|G|\log|\mathcal{G}||G|}{n^{\gamma_2}|G|^{\gamma_1}}} + \frac{|\mathcal{G}_{\cap S}|\log|\mathcal{G}||G|}{n^{\gamma_2}|G|^{\gamma_1}}\right)\right).\end{aligned}$$

If $|G|^{1-\gamma_1}n^{-\gamma_2} = o(1)$, the Co-adaptive Lasso is superior to the Group Lasso. In particular, if B2 is satisfied with $\gamma_1 = 1$, $\gamma_2 > 0$, then the Co-adaptive Lasso attains quickly the $O\left(\sigma^2|\widetilde{S}|/n\right)$ rate.

The Co-adaptive Lasso holds a further advantage if within group sparsity exists. Suppose that the conditions of Corollary 5.2.2 and those necessary for the Group Lasso are satisfied, and we adopt the same scaling context as in Section 5.3.2. Then

substituting in, we have

$$\left\|X(\beta - \widehat{\beta}^{(g)})\right\|_n^2 = O\left(\sigma^2 \frac{|S|}{n}\left(\frac{n^{\gamma_g}}{n^{\gamma_s}} + \frac{\sqrt{\log(p) - \gamma_g \log(n)}}{n^{\gamma_s - \gamma_g/2}} + \frac{(\log(p) - \gamma_g \log(n))}{n^{\gamma_s}}\right)\right).$$

Meanwhile, recall that the Co-adaptive Lasso achieves

$$\left\|X\left(\beta - \widehat{\beta}^{(c)}\right)\right\|_n^2 = O\left(\sigma^2 \frac{|S|}{n} \max\left(\frac{\log(p) - \gamma_g \log(n)}{n^{\gamma_l - 2\gamma_\beta + \gamma_s}}, \log|S| + (\gamma_g - \gamma_s)\log(n)\right)\right).$$

Recalling that typically $S < n$, if the $\log|S| + (\gamma_g - \gamma_s)\log(n)$ part of the $\max(\cdot, \cdot)$ in this bound is active, then the Co-adaptive Lasso can be a great improvement if $\gamma_g > \gamma_s$. Otherwise, we still get a better bound for the Co-adaptive Lasso if $\gamma_l - 2\gamma_\beta > 0$, or if

$$\frac{\sqrt{\log|\mathcal{G}|}}{n^{\gamma_g/2 - \gamma_l + 2\gamma_\beta}} < 1.$$

In particular, if $|\widetilde{S}|/n$ fails to converge, the Co-adaptive Lasso can succeed if $|S|$ diminishes quickly enough, while we cannot usually expect success with the Group Lasso. This is because the within group penalisation of the Co-adaptive Lasso is the Lasso itself, as opposed to the Ridge penalty of the Group Lasso.

Nevertheless, the Group Lasso can do better if the conditions for the Co-adaptive Lasso are too difficult to satisfy, especially if the initial Lasso fails completely to converge, or if the coefficients in each group are very small. In particular, the multi-task learning context analysed in Lounici et al. [2010], where a set of separate regressions are related by a common sparsity pattern, fails to converge for the Lasso as the number of tasks alone increases, and so will generally fail with the Co-adaptive Lasso.

5.3.4 Summary

From a high level analysis, then, our impression is that compared to the other methods in this section, the Co-adaptive Lasso requires weaker conditions than the Adaptive Lasso, stronger conditions than the Group Lasso, and slightly stronger conditions than the Lasso.

In most situations it will outperform the Lasso, unless group-wise coefficient norms are very small relative to the initial Lasso's error rate, in which case neither algorithm will perform well.

In the AIAO situation, or more generally the situation where the within group sparsity proportion remains constant, the Co-adaptive Lasso will usually at least match the Adaptive Lasso, and do better than it if the Adaptive Lasso does not achieve the optimal rate anyway – for example, if there is substantial variability in the non-zero coefficient values β_S and not fast enough convergence in the initial Lasso. This can occur quite often in problems with large group sizes. If the group *averaged* coefficient sizes remain large enough relative to the rate of convergence of the initial Lasso, the Co-adaptive Lasso will do better than the Group Lasso as well.

In the non-AIAO situation where the proportion of non-zeroes in each $\mathcal{G}_{\cap S}$ diminishes, the the Co-adaptive Lasso will be inferior to the Adaptive Lasso if the Adaptive Lasso succeeds in attaining its optimal rate. If it does not, then the Co-adaptive Lasso can still be superior, so long as there is not too much within group sparsity, or there is variability of non-zero coefficients present that the Co-adaptive Lasso can handle better though its weight averaging. Compared to the Group Lasso, the Co-adaptive Lasso will do better if the weight convergence is such that $|\widetilde{S}|$ becomes the limiting factor, if there is good weight convergence, or if the group sizes are large.

5.4 Winsorised weights

The results in Section 5.2, in some sense, act as a worst case result. For many cases, the Co-adaptive Lasso will perform far better than Theorem 5.2.1 would indicate. Further, we might seek to use slight modifications of the weight formulation of (5.1.2) to improve results. Of this, one particular example is the application of robust statistics methodology, and in particular winsorisation.

Consider one important case where the Co-adaptive Lasso can potentially fail to produce very good results – the case where in the initial Lasso estimate, much of the error is concentrated into a very small number of $\widehat{\beta}_j^{(l)}, j \in \widetilde{S}^c$. If the error is not so concentrated, then the averaging implicit in the weight formulation (5.1.2) will ensure that $w_{\widetilde{S}^c}$ is kept small. But if the error is so concentrated, then it becomes likely that much of the spurious $\widehat{\beta}_j^{(l)}$ will also be concentrated in the weight calculation for a small number of groups. This therefore means that for those groups, the penalisation in the second stage is much too weak, leading to weak performance.

However, if the error were so concentrated, then even within those groups, the large w for variables belonging to those groups will be due to only a small number of

$\widehat{\beta}_j^{(l)}$. Winsorisation is one possible way of taking advantage of this fact to control for this case. Consider weights of the form:

$$\widetilde{w}_j = \left(\sum_{G \ni j} \left\| \min \left(|\widehat{\beta}_G^{(l)}|, |\widehat{\beta}_G^{(l)}|_{(|G|-\alpha_G)} \right) \right\| \right)^{-1},$$

where α_G is a small constant, and $|\widehat{\beta}_G^{(l)}|_{(i)}$ denotes the ith order statistic of $|\widehat{\beta}_G^{(l)}|$.

Lemma 5.4.1. *Let*

$$\lambda > 2 \max |X'\varepsilon/n|.$$

Suppose that A1 holds, and the group structure $\mathcal{G}$ is independent of the variables. Then with high probability, for any $U \geq 1$, there exists α'_G that can be explicitly bounded such that for $G \in \mathcal{G}_{\widetilde{S}^c}$, winsorising by α_G gives

$$\widetilde{w}_G \geq \left(\frac{9\lambda \left(\sqrt{\alpha_G} + \sqrt{\alpha'_G} \right)}{\phi^2(3, S)} \frac{|S|}{U} \right)^{-1}.$$

Correspondingly, for $G \in \mathcal{G}_{\cap S}$,

$$\widetilde{w}_G \leq \left(\sum_{G \ni j} \left\| \min \left(|\beta_G|, |\beta_G|_{(|G|-\alpha_G)} \right) \right\| - \frac{3\lambda\sqrt{|S|}}{\phi^2(3, S)} \right)^{-1}.$$

Notice that our convergence results rely only on $A1$ instead of the stronger $A2$. Furthermore, in contrast to unwinsorised weights which correspond to the case of fixed U, Lemma 5.4.1 allow us to offset increasing $|S|$ with increasing U, paying a price in terms of α_G, α'_G, that can be much smaller. This leads to potentially much stronger capability at removing irrelevant covariates.

Indeed, bearing in mind that the Lasso estimate always chooses a maximum of $\min(n, p)$ variables, we have in the case of $p \gg n$ the possibility of eliminating the irrelevant groups altogether with a sufficiently high choice of α_G. For many cases, choosing $\alpha_G = \lceil |G|/2 \rceil$ would do:

Lemma 5.4.2. *Suppose that $Supp(\widehat{\beta}_{S^c}^{(l)})$ is a random sample without replacement from S^c. Let C be the event that there exists a group $G \in \mathcal{G}^c_{\cap S}$ with $|G|/2$ elements belonging to this set. Then if the groups are evenly sized, with $2 \leq |G| \leq |S| \leq n \leq p$,*

$$Pr(C) \leq |\mathcal{G}^c_{\cap S}| \left(\frac{4e^2 n}{p - 2n} \right)^{|G|/2}.$$

Note that a slightly better bound is possible, using Stirling's formula. In any case, the consequence of Lemma 5.4.2 is that so long as $p^{1-2/|G|}$ grows faster than linearly in n, we will with high probability throw out all the irrelevant groups $\mathcal{G}^c_{\cap S}$ in the initial run. Then, assuming that the initial Lasso does not do too badly, the second stage calculation will give at worst the OLS estimate restricted to $\widetilde{S}$.

The disadvantage in both cases is that we require a true signal that is resistant to winsorisation and at least some, possibly slow, convergence from the initial estimator in terms of $\left\|\widehat{\beta}^{(l)}_{\widetilde{S}} - \beta_{\widetilde{S}}\right\|$, so that the initial estimate still returns some signal on the true groups that would not be eliminated by our weight calculation. If the initial estimate fails altogether, then winsorisation cannot be expected to rescue the estimation. Finally, the independence assumption is a very strong one for most situations, though mild violation of the independence may still leave some usefulness to this approach.

Practical application of Lemma 5.4.1 may be difficult. The calculation of the optimal U and corresponding α_G requires knowledge of the true signal. Bootstrapping or cross validation may be helpful in some cases to estimate these. We note that dependence of α_G on G is only via $\log(|G|)$, so it would be reasonable to use the same α_G for all G, even if $|G|$ varies modestly. In some cases, from subject knowledge we may be able to assume that there is a significant margin between α_G large enough to eliminate most of the Lasso's mistakes, and α_G that is too large and will reduce the true signal groups to apparent irrelevance. For example, with β_S normal mean 0 with constant variance σ^2 within each group, even in the extreme case of winsorising at the 50th percentile, we have by simple calculation that the winsorised true weights would, with high probability, be no worse than a constant times worse than the unwinsorised case in terms of the sizes of $\widetilde{w}_{\widetilde{S}}$. However, if the signal is indeed very sparse within groups, winsorisation is dangerous and should not be used.

5.5 Overlapping groups

We have considered so far only non-overlapping groups. The existence of overlaps amongst groups poses a problem to the practitioner as to how to handle these overlaps. It is then necessary to tailor the algorithm to deal with overlapping groups in the appropriate fashion.

For the Group Lasso penalty $P_{group}(\beta) = \sum_G \|\beta_G\|$:

$$\frac{\partial}{\partial \beta_i} P_{group}(\beta) = \sum_{G \ni i} \frac{\beta_i}{\|\beta_G\|}$$

has a singularity at $\beta_i = 0$ if and only if there exists a group G with $i \in G$ and $\|\beta_G\| = 0$. As the presence of singularities indicate 'corners' in the penalty for which exactly zero estimates are possible, we have that allowable sparsity patterns take the form of intersections of complements of overlapping groups, with coefficients in overlaps between groups especially unlikely.

While this is useful in some cases [Huang et al. 2009], alternative interpretations of group overlaps might be more desirable in others. In the general case, a typical application is to find signals that are unions of groups. Jacob et al. [2009] proposes one way to accommodate this situation, via the overlapping group norm:

$$\|\beta\|_{overlap} = \sum_{G \in \mathcal{G}} \|\widehat{v}_G\|,$$

where $\widehat{v}_G \in \mathbb{R}^p$ for each $G \in \mathcal{G}$, and satisfies

$$\{\widehat{v}_G\} = \arg\min_{\{v_G\}} \sum_{G \in \mathcal{G}} \|v_G\| \quad s.t. \quad \forall G, \, \text{Supp}(v_G) \subset G, \quad \sum_{G \in \mathcal{G}} v_G = \beta.$$

This norm can then be used as a penalty to produce the Overlapping Group Lasso.

However, Percival [2011] established that there are several problems with this formulation - the computational cost can be very burdensome, and the conditions required for success are stringent, with generally poor results if the structure of the groups are too complex. Particular examples raised were nested group structures, and presence of sparsity in the true groups.

However, in the co-adaptive framework, a flexible, and natural alternative framework can be constructed to deal with overlapping groups. For example, we can replicate a Group Lasso style behaviour for the Co-adaptive Lasso by calculating weights as, for each $j = 1, \ldots, p$,

$$w_j = \sum_{G \ni j} \sqrt{|G|} / \left\| \widehat{\beta}_G^{(l)} \right\|.$$

For behaviour involving selecting unions of groups, a variety of methods for choosing weights are possible, and we suggest here two possibilities for various scenarios:

5.5.1 Overlapping group norm minimisation

We may begin with one approach to improving the performance of the Overlapping Group Lasso, suggested in Percival [2011], which is to calculate the Adaptive Overlapping Group Lasso. Specifically, Percival [2011] suggests using the OLS estimate $\widehat{\beta}^{OLS}$ to calculate group weights. Let $\widehat{v}_G$ be the associated v_G in the computation of $\left\|\widehat{\beta}^{OLS}\right\|_{overlap}$. Then we can calculate for each group weights w_G as, for some $\gamma < 0$, $w_G = \|\widehat{v}_G\|^{\gamma}$. This can then be applied to a new Overlapping Group Lasso computation.

Under this, and assuming some conditions – in particular, that the above decomposition is unique in a neighbourhood around around the true value, and that the minimising decomposition of the true value itself gives support for each $\widehat{v}_G$ that is tight around the true relevant covariates – they are able to prove consistency in the fixed design case where n alone goes to infinity.

Such a strategy may be similarly applied to the Co-adaptive Lasso. For example, we may conduct a weighted Lasso using, for each covariate k, a weight w_k equal to the minimal w_G, computed above, amongst groups G containing k. Then the same proof given in Percival [2011] will suffice to show the same consistency result under the same conditions. A second modification would be to use $\widehat{\beta}^{(l)}$ instead of $\widehat{\beta}^{OLS}$, which would allow use when $p \geq n$.

On the other hand, group norm minimisation weights has a variety of shortcomings. As discussed in Percival [2011], the uniqueness criterion is strong, and rules out possibilities like nested groups.

5.5.2 Maximum covariate-wise group norm

A simple alternative approach is to take the initial estimate, which can be the Lasso $\widehat{\beta}^{(l)}$, calculate the grouped L2 norms $\|\beta_G\|$ for each group G, and weigh each covariate j as

$$w_j = \min_{G \ni j} \sqrt{|G|} / \left\|\widehat{\beta}^{(l)}_G\right\|.$$

This approach reduces to the previous case of Equation (5.1.2) when the groups do not overlap.

The shortcoming of this approach is that the lightly penalised covariates – that is, the variables j for which $w_j \not\to \infty$ as $\widehat{\beta}^{(l)} \to \beta$ will correspond to $\cup \mathcal{G}_{\cap S}$, the union of groups that overlap S. This can be potentially very large, especially if there exists many large groups. In that case, the Co-adaptive re-weighting will eliminate

an insufficient proportion of the variables, thus leading to a fit that is not much better than the Lasso. Note that this situation creates similarly problems with the overlapping Group Lasso.

One possible way to fix this issue is if we know or can infer the degree of overlap of groups with S, or whether there exists a covering set of groups for which there is very little within group sparsity. Then, we can compute instead of $\|\beta_G\|^2$, appropriately trimmed or winsorised sums of β_G^2. By this method, we reduce the mistakenly picked up overlapping groups to only those with large overlaps with the truly relevant ones, paying the price of potentially losing covariate groups that have very high levels of within group sparsity. This provides one further potential benefit of the methodology in Section 5.4 in this setting.

Chapter 6

Applying the Co-adaptive Lasso

In this chapter, we discuss details of the practical application of the Co-adaptive Lasso. We address the selection of tuning parameters, and then discuss the potential of two important extensions – a version using repeated co-adaptive steps and an online variant of the procedure. We give empirical examples comparing the procedure to several other methods, and finally discuss an application of the co-adaptive principle to the problem of fitting Rule Ensembles.

6.1 Implementation of the Co-adaptive Lasso

The Co-adaptive Lasso allows a variety of possible implementations. In general, any algorithm to compute the Lasso can be used. For our empirical work, we have mainly used the coordinate descent algorithm of Friedman et al. [2007], due to its speed and simplicity. Other algorithms are possible, that may provide greater computational efficiency.

One useful point in adapting algorithms for reweighted Lasso calculations is that it is generally not necessary to produce specialised variants of algorithms to accommodate the covariate weights. This is because an identical effect can be produced by instead rescaling the covariates prior to computation. This can be easily seen:

$$
\begin{aligned}
\widehat{\beta}^{(c)} &= \arg\min_{\beta} \frac{1}{2} \left\| Y - \sum_{k=1}^{p} X_{\cdot,k}\beta_k \right\|_n^2 + \lambda \sum_{k=1}^{p} w_k |\beta_k| \\
&= \arg\min_{\beta} \frac{1}{2} \left\| Y - \sum_{k=1}^{p} \frac{X_{\cdot,k}}{w_k} w_k\beta_k \right\|_n^2 + \lambda \sum_{k=1}^{p} |w_k\beta_k| \\
w\widehat{\beta}^{(c)} &= \arg\min_{\widetilde{\beta}} \frac{1}{2} \left\| Y - \sum_{k=1}^{p} \frac{X_{\cdot,k}}{w_k} \widetilde{\beta}_k \right\|_n^2 + \lambda \sum_{k=1}^{p} |\widetilde{\beta}_k| \qquad (6.1.1)
\end{aligned}
$$

Hence it suffices to compute a Lasso computation using X with each column divided by the corresponding component of w, and then divide the estimated $\widehat{\beta}$ by w. For $1/w_k = 0$, we may simply remove the corresponding covariate from the calculation.

Practical computation of the Co-adaptive Lasso requires the use of the correct tuning parameter. While for the Lasso, the work of Zou et al. [2007] have identified degrees of freedom based methodologies, such approaches are more difficult for our reweighting methods. While we can use the number of selected variables in the second stage as an estimate of the degrees of freedom, this will lead to overfitting because it fails to take into account variability in the first stage weights. In our experiments, we have instead used 10-fold cross validation for choosing the tuning parameter.

The particular implementation of 10-fold cross validation for the co-adaptive Lasso is the two step procedure of Algorithm 5. While a variant may be conducted where we optimise over μ and λ simultaneously, by way of a two dimensional grid, we adopt this version because it means we only need to select one tuning parameter at a time. Thus, we reduce the computational burden, especially for more fine grids and the results in Chapter 5 suggest this can still be quite successful. Note that for each fold, we calculate a new set of weights, using the training data available $X_{P_t^c,\cdot}, Y_{P_t^c}$, instead of using computed weights from X, Y. This allows the cross validation procedure to properly capture the variability arising from the first stage, thus avoiding overfitting in our estimation of $\widehat{\mu}$.

6.2 Repeating the Co-adaptive Lasso

While we have shown good results for a two step procedure with the Co-adaptive Lasso, it can be the case that repeating the reweighting and recalculation can lead to

Algorithm 5 Cross validation for the Co-adaptive Lasso

Require: X, Y.

1: Compute $\widehat{\lambda}$ for the Lasso by standard 10-fold cross validation.
2: Choose a random partition of $\{1, \ldots, n\}$ into 10 subsets each partition the same size, $P_1 \ldots, P_{10}$.
3: **for** $t = 1, \ldots, 10$ **do**
4: Calculate $\widehat{\beta}^{(l)}$ according to the Lasso, using $X_{P_t^c,\cdot}, Y_{P_t^c}$, and $\widehat{\lambda}$.
5: Compute w from $\widehat{\beta}^{(l)}$.
6: Calculate $\widehat{\beta}_{\mu}^{(c)}$ according to the reweighted Lasso, using $X_{P_t^c,\cdot}, Y_{P_t^c}$, and w, over a log grid of μ.
7: Compute $MSE_t(\mu) = \left\| Y_{P_t} - X_{P_t,\cdot}\widehat{\beta}_{\mu}^{(c)} \right\|^2 / |P_t|$.
8: **end for**
9: Aggregate $\overline{MSE}(\mu) = \sum_{t=1}^{10} MSE_t(\mu)$.
10: Choose $\widehat{\mu} = \arg\min_{\mu} \overline{MSE}(\mu)$.
11: **return** $\widehat{\lambda}, \widehat{\mu}$.

improvements in our results. Observe for example the improvement reached with the simple Adaptive Lasso in the example shown by Figure 4.2.1. A similar procedure can be done with the Co-adaptive Lasso, as described in Algorithm 6. Note that the non-repeated Co-adaptive Lasso we discussed previously is simply this algorithm terminated upon the completion of step $m = 2$. Repeating until convergence here is very similar to the LCD procedure of Breheny and Huang [2009], save that we don't fix a single value of λ, and we iterate until convergence before updating the weight calculation, instead of computing the weighting calculation as we go.

Algorithm 6 Repeated Co-adaptive Lasso

Require: X, Y

1: Set $m = 0$.
2: Initialise fits $\widehat{\beta}^{(0)}$ as identically 1.
3: **repeat**
4: $m \leftarrow m + 1$.
5: Calculate weights $w_G^{(m)} \leftarrow \sqrt{|G|} / \left\| \widehat{\beta}_G^{(m-1)} \right\|$ for each $G \in \mathcal{G}$.
6: Choose an appropriate $\lambda^{(m)} \geq 0$.
7: Calculate new estimate $\widehat{\beta}^{(m)}$,

$$\widehat{\beta}^{(m)} = \arg\min_{\beta} \frac{1}{2} \|Y - X\beta\|^2 + \lambda^{(m)} \left\| w^{(m)}\beta \right\|_1$$

8: **until** sufficient convergence is achieved, or a m exceeds some $m_{\max}$.
9: Output final estimate $\widehat{\beta}^{(m)}$.

Now, the cost of additional steps would be that we must necessarily increase the

computational requirements, though we can improve efficiency somewhat when using Coordinate Descent by warm starting each stage with the estimate from the previous one. The question then is how much of an improvement we can expect with each additional step, which will enable us to determine when we should terminate the computation.

Theorem 6.2.1 (Prediction error for Repeated Co-adaptive Lasso)**.** *Assume conditions B1 and B2 apply.*

Let $\widetilde{w}_{\widetilde{S}}$ be defined by Equation (5.1.2)*, but using β in place of $\widehat{\beta}^{(l)}$.*

Suppose there exists $m_0 > 1$, $\delta < 1$ such that for all $m \geq m_0$, and an appropriate choice $\lambda^{(1)}, \ldots, \lambda^{(m_0)}$,

$$(1-\delta)\widetilde{w}_{\widetilde{S}} \leq w_{\widetilde{S}}^{(m)} \leq (1+\delta)\widetilde{w}_{\widetilde{S}}$$
$$1/w_{\widetilde{S}^c}^{(m)} < \delta,$$

with the inequalities taken component-wise.

Further, suppose there exists $C > 0$, T satisfying $\widetilde{S} \supseteq T \supseteq S$ such that for

$$L_1 = 2(1+\delta)(\max \widetilde{w}_{\widetilde{S}})\delta, \quad L_2 = 2\left(\frac{1+\delta}{1-\delta}\right)\frac{\max \widetilde{w}_{\widetilde{S}}}{\min \widetilde{w}_{\widetilde{S}}},$$

we have the RE and GRE conditions:

$$\max\left(\phi(L_1, \widetilde{S}), \phi(\max(L_1, L_2), T)\right) > C \tag{6.2.1}$$
$$\max\left(\phi_{\mathcal{G}}(L_1, \widetilde{S}), \phi_{\mathcal{G}}(\max(L_1, L_2), T)\right) > C. \tag{6.2.2}$$

Then for $m > m_0$, there exists a sequence $\lambda^{(m_0)}, \lambda^{(m_0+1)}, \ldots$, such that either

$$\left\|X(\widehat{\beta}^{OLS,T} - \widehat{\beta}^{(m)})\right\|_n^2 \leq \rho \left\|X(\widehat{\beta}^{OLS,T} - \widehat{\beta}^{(m-1)})\right\|_n^2,$$

for some $\rho = o\left(n^{-\gamma_2/2}\sqrt{\frac{|T|}{|S|}}\max_{\substack{G\in\mathcal{G}_{\cap S},\\ H\in\mathcal{G}_{\cap S}^c}}\sqrt{\frac{|G|^{1-\gamma_1}}{|H|}}\right)$, or for sufficiently large m,

$$\left\|X(\widehat{\beta}^{OLS,T} - \widehat{\beta}^{(m)})\right\|_n^2 = O\left(\sigma^2 \frac{\log(|\widetilde{S}|)|T|}{n}\frac{\max \widetilde{w}_{\widetilde{S}}^2}{\min \widetilde{w}_{\widetilde{S}}^2}\right).$$

In other words, so long as we reach a certain level of accuracy, we can place an bound on the performance of multiple stage Co-adaptive Lasso computations that is exponential in number of stages, up until we reach a limiting performance which corresponds to the Lasso with only the variables $\widetilde{S}$, instead of the full number of covariates p. Beyond this, we expect no improvement in performance.

Since ρ can be quite small, this limiting performance can be reached quite quickly. However, assuming we do not reach it with the initial step $m = 2$, there is some potential for improvement on the results in Chapter 5 with Algorithm 6. We see a little of this in Figure 6.2.1, though the effect is very modest.

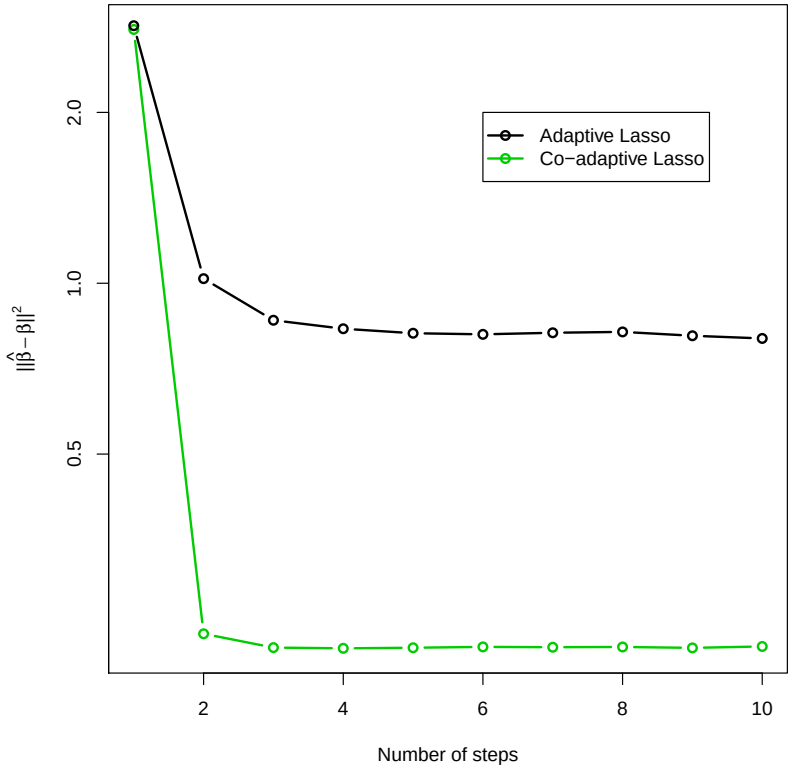

Figure 6.2.1: Estimation performance for repeated Co-adaptive and Adaptive Lasso. We generated $X \sim N(0,1)$, with $n = 250$, $p = 2000$. $\mathcal{G}$ is such that each G are the consecutive indices $\{1,\ldots,10\},\{11,\ldots 20\},\ldots$. We choose S to be the first 8 indices in each of the first 5 groups, so that $|S| = 40$. β_S are each generated as $N(0,1)$, and $Y \sim N(X\beta, 1)$. The graph is compiled as the average of 50 iterations, with each step's $\lambda^{(m)}$ chosen by 10-fold CV.

Further extensions are possible. One possibility is to change the type of weighting calculation in later stages. For example, assuming that the second stage $\widehat{\beta}^{(c)}$ is a good estimate of β, we might use this as an initial estimate in an Adaptive Lasso procedure – that is, perform one further step but this time without considering the grouping

structure. This will likely improve the selection accuracy within groups, assuming that the non-zero coefficients within each group is large in absolute value. It would be interesting to see if such a procedure produces a significant improvement over the Co-adaptive Lasso or the Adaptive Lasso.

6.3 Online algorithm

For very large datasets, the standard batch calculation methodologies are too slow. We can solve this by, for example, taking random samples of the data, and aggregating the results from multiple samples. However, this can perform poorly, especially when the dataset is so large that our samples must necessarily be small fractions of the full sample.

An alternative solution is to adopt an online algorithm. Online algorithms accept the data sequentially, updating the solution sample per sample, using only one individual sample (or perhaps a small batch of samples) at a time. This means that inherently, they scale at $O(n)$ in computational complexity and $O(1)$ in memory requirements with increasing sample size n. As established in Chapter 2, an online algorithm exists for the Lasso itself – the Regularised Dual Averaging (RDA) algorithm, and the related Enhanced L1-RDA method [Xiao 2010]. This algorithm obtains not only acceptable regret bounds, but also produces sparse solutions at each stage of the computation.

We can construct an extension of this method that calculates solutions for the Adaptive and Co-adaptive Lasso, as Algorithm 7.

This algorithm works by having two separate RDA processes running concurrently. The first process, in steps 4-7, computes the current Lasso-style estimate, according to the usual Enhanced RDA method of Xiao [2010]. The second process, in steps 8-12, however, takes the averaged output $\overline{\beta}^{(1)}$ from the other process, computes weights, and applies them as a scaling factor to X, before doing the usual RDA update. The intuition is that the first process converges to the Lasso solution, while the second process produces a solution to an optimisation that eventually converges to the second stage of the Co-adaptive Lasso. Note that an online version of the Adaptive Lasso can be obtained, as usual, by setting $\mathcal{G}$ to be the set of single element sets.

Several tuning parameters are introduced in this formulation, which have an interpretation as in Xiao [2010]. The γ parameters determine the learning rate of the algorithm - a small γ converges more quickly, but at the cost of increased instability,

Algorithm 7 Coadaptive Enhanced L1 RDA

Require: Data: $Y_t \in \mathbb{R}, X_{t,\cdot} \in \mathbb{R}^p$ for $t = 1, 2, \ldots$, grouping structure $\mathcal{G}$.
Coadaptive tuning parameters $\lambda > 0, \mu > 0$.
Online algorithm tuning parameters $\rho^{(1)}, \rho^{(2)}, \gamma^{(1)}, \gamma^{(2)}$.

Initialise: $\overline{g}_k^{(1)} = 0$, $\overline{g}_k^{(2)} = 0$, $\beta_k^{(1)} = 0$, $\beta_k^{(2)} = 0$, $\overline{\beta}_k^{(1)} = 0$, $\overline{\beta}_k^{(2)} = 0$ for $k = 1, \ldots, p$.
for $t = 1, 2, \ldots$ **do**
  Set current penalties: $\widetilde{\lambda} \leftarrow \lambda + \gamma^{(1)}\rho^{(1)}/\sqrt{t}, \quad \widetilde{\mu} \leftarrow \mu + \gamma^{(2)}\rho^{(2)}/\sqrt{t}$.

Compute unreweighted update:

  Calculate gradient: $g^{(1)} \leftarrow \left(\sum_{k=1}^p X_{t,k}\beta_k^{(1)} - Y_t\right) X_{t,\cdot}$.
  Update dual average: $\overline{g}^{(1)} \leftarrow (t-1)\overline{g}^{(1)}/t + g^{(1)}/t$.
  Make an estimate of: $\beta_k^{(1)} \leftarrow -\frac{\sqrt{t}}{\gamma^{(1)}}\text{sign}(\overline{g}_k^{(1)})\left(|\overline{g}_k^{(1)}| - \widetilde{\lambda}\right)_+, \quad k = 1, \ldots, p.$
  Update primal average: $\overline{\beta}^{(1)} \leftarrow \frac{t-1}{t}\overline{\beta}^{(1)} + \frac{1}{t}\beta^{(1)}$.

Compute reweighted update:

  Compute weights: $1/w_k \leftarrow \sqrt{\sum(\overline{\beta}_G^{(1)})^2/|G|}$, for $k \in G$.
  Calculate reweighted gradient: $g^{(2)} \leftarrow \left(\sum_{k=1}^p X_{t,k}\beta_k^{(2)}/w_k - Y_t\right) X_{t,\cdot}/w$.
  Update dual average: $\overline{g}^{(2)} \leftarrow (t-1)\overline{g}^{(2)}/t + g^{(2)}/t$.
  Make an estimate: $\beta_k^{(2)} \leftarrow -\frac{\sqrt{t}}{\gamma^{(2)}}\text{sign}(\overline{g}_k^{(2)})\left(|\overline{g}_k^{(2)}| - \widetilde{\mu}\right)_+, \quad k = 1, \ldots, p.$
  Update primal average: $\overline{\beta}^{(2)} \leftarrow \frac{t-1}{t}\overline{\beta}^{(2)} + \frac{1}{t}\beta^{(2)}$.
end for
return $\widehat{\beta}_t^{(o)} = \overline{\beta}^{(2)}/w$.

with too small values of γ often in practice leading to the estimate rapidly exploding and so failing to converge at all. Meanwhile, the ρ factor applies an increased sparsity at the early stages of the calculation. Large ρ again implies more stable results and greater sparsity, but slower convergence to the target optimisation λ and μ. To avoid having to determine too many parameters, a reasonable heuristic seems to be to set

$$\gamma^{(2)} = \gamma^{(1)}, \quad \rho^{(2)} = \frac{\mu}{\lambda}\rho^{(1)},$$

thus applying the same learning rate on both sequences, and applying the same proportional sparsity promotion.

The theorems in Xiao [2010] may be applied to Algorithm 7 to show that this converges under certain conditions:

Theorem 6.3.1. *Suppose that* $\left\|g^{(1)}\right\|$ *and* $\left\|g^{(2)}\right\|$ *are bounded, and* w_k^{-1} *is a continuous function of* $\widehat{\beta}^{(l)}$ *for all* k*. Writing* $\widehat{\beta}_\infty^{(c)}$ *as the population co-adaptive estimate for the Co-adaptive Lasso,*

$$\widehat{\beta}_\infty^{(c)} = \arg\min_\beta E\left\|Y - X\beta\right\|^2/2 + \lambda\sum_{k=1}^{p}|\beta_k|E(w_k),$$

we have that as $t \to \infty$*,*

$$\widehat{\beta}_t^{(o)} \xrightarrow{p} \widehat{\beta}_\infty^{(c)}.$$

However, note that in general, convergence rates in online procedures are slow, and this slowness can eclipse the improvements of the Co-adaptive reweighting. In addition, selection of tuning parameters can be difficult. Without the sharing of information from results computed at adjacent λ and μ grid points present in coordinate descent, a fairly coarse grid is required with a RDA estimator for each grid, which may be evaluated by online cross validation with using multiple RDA chains. ρ and γ also present additional tuning parameters which the algorithm can be sensitive to, and these must also be determined. Nevertheless, the procedure is effective in some cases.

To illustrate the method, consider the following artificial scenario:

Let us generate $X_{t,\cdot} \in \mathbb{R}^p$ with each $X_{t,k} \sim N(0,1)$. Choose $p = 500$, and set $Y_t = \sum_{k=1}^{50} X_t\beta_t + \varepsilon_t$, where each ε_t is independently $N(0,1)$. β is set so that $\beta_k = k \mod 5$ for $k \leq 50$, and 0 otherwise. Hence, β_S is the repeated sequence $(1,2,3,4,0)$ for $S = \{1,\ldots,50\}$.

We choose tuning parameters by experiment to minimise, approximately, the estimation error. Figure 6.3.1 shows the results of this 'best' choice of tuning parameter. Note that in real situations, because the 'true' β is unavailable, we would not be able to attain such performance.

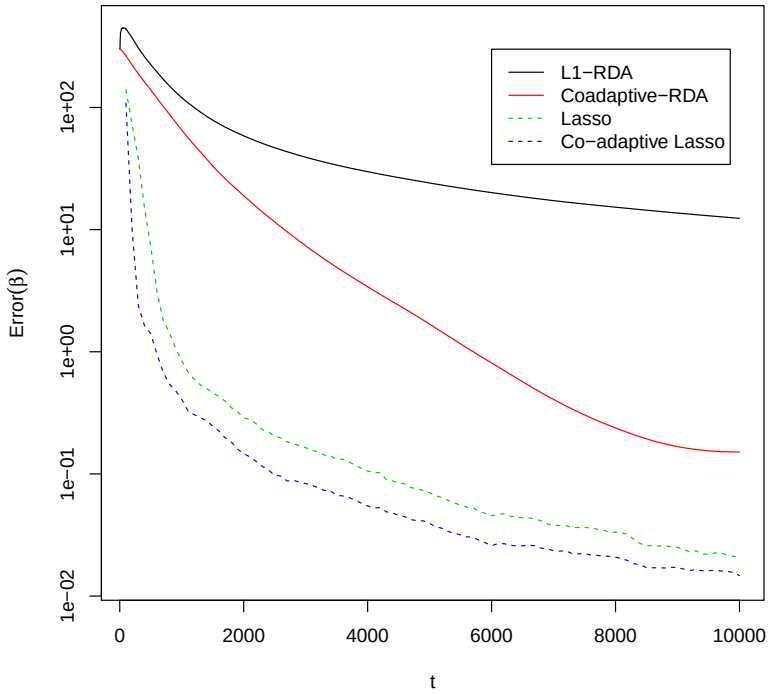

Figure 6.3.1: Estimation performance for online methods and batch methods, as a function of supplied samples. Estimation error is given as $\left\|\beta - \widehat{\beta}_t\right\|^2$ for each estimator's $\widehat{\beta}_t$, with the batch estimators using as input all of the data from 1 to t to build $\widehat{\beta}_t$.

As shown in the figure, the Co-adaptive RDA method greatly outperforms the L1-RDA method, but is, in our scenario, inferior to both the Lasso and Co-adaptive Lasso batch methods. Nevertheless, processing for the online methods is much faster than for the batch methods with large sample sizes, especially if we are repeatedly producing new estimates as additional samples come in. While several practical and theoretical aspects remain to be solved, this approach seems promising as an avenue for further research, as a way of acquiring the benefits of concave penalties in a new context.

6.4 Empirical results

We shall investigate the performance of the Co-adaptive Lasso in a variety of datasets, both real and generated.

6.4.1 Splice site prediction

As considered in Meier et al. [2008], the splice problem concerns the prediction of splice sites – the regions between coding (exons) and non-coding (introns) DNA segments. In particular, the task of predicting 'donor' splice sites – the 5' splice site end of introns – has been commonly used to demonstrate the Group Lasso [Meier et al. 2008], [Roth and Fischer 2008].

As in Meier et al. [2008], we consider the MEMset dataset [Yeo and Burge 2008], which consists of sub-sequences of DNA that contain the consensus position "GT", which are either true donor splice sites, or otherwise. Removing the consensus position then gives sequences of length 7 with 4 levels ($\{A, C, G, T\}$). The original dataset consists of 8415 true and 170438 false human donor sites, with an additional test set of 4208 true and 89717 false donor sites. As per the original, we use the training set to build a smaller balanced training dataset with 5610 true and 5610 false donor sites, and an unbalanced validation set with the remainder, having the same ratio of true to false sites as the test set. This is done through sampling at random, without replacement.

Our goal is accurately predict whether a candidate site from the test set, which we keep unobserved during fitting, is a true or a false donor splice set. This is measured by the maximum attainable correlation coefficient [Yeo and Burge 2008] between a predicted vector and the true one,

$$\mathrm{cor}_{\max} = \max_{\tau} \mathrm{cor}\left(y_{test}, I\{\widetilde{p}(x_{test}) \geq \tau\}\right).$$

We consider dummy variables as in Meier et al. [2008], using scaled versions of the indicator variables of ($\{A, C, G\}$), plus all interactions up to three way. We use the same grouping structure as they do, grouping together variables corresponding to each original site, and interaction level. We train on the balanced dataset, using the validation set to choose tuning parameters – unlike Meier et al. [2008], we select according to maximising $\mathrm{cor}_{\max}$ on the validation set instead of maximising likelihood. This allows us to dispense with the need to correct the intercept.

We compare logistic versions of the Lasso, the Group Lasso [Yuan and Lin 2006], the Adaptive Lasso [Zou 2006], the Adaptive Group Lasso [Wang and Leng 2008], and the Co-adaptive Lasso. For the two stage procedures, we use the same training and validation set for each stage of the procedure, choosing the first stage tuning parameter λ to minimise error on the validation set.

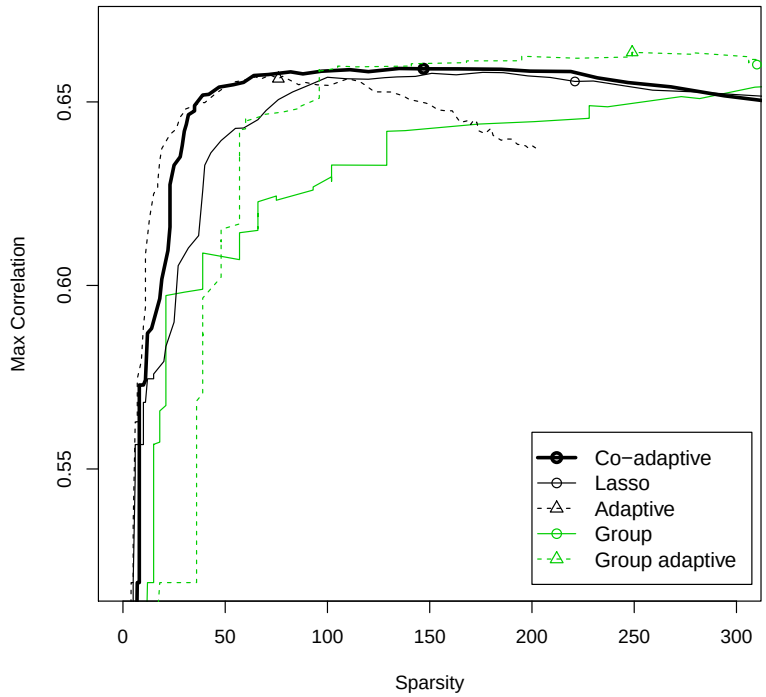

Figure 6.4.1: Sparsity level $|S|$ vs $\text{cor}_{\max}$ on the test set. The points on each curve represents the model chosen by reference to the validation set. The chosen model for the Group Lasso uses 966 variables and so is omitted from the graph for presentation reasons.

From Table 6.1, overall, the attained $\text{cor}_{\max}$ is highly similar for all algorithms. The Adaptive Group Lasso achieves the best results, with the Group Lasso and Co-adaptive Lasso having similar results. The Adaptive Lasso and Lasso perform less well, but still reasonably. These results suggest there is not much within group sparsity on this dataset - the high level of correlation (which, on the training set, reached a maximum of 0.88) means there is some advantage in selecting redundant variable sets to reduce variability on the test set.

More interesting is the breakdown of performance according to group sparsity and sparsity. Figure 6.4.1 and Figure 6.4.2 give these results, which we generate by supplying each algorithm with a grid of tuning parameters, and recording for

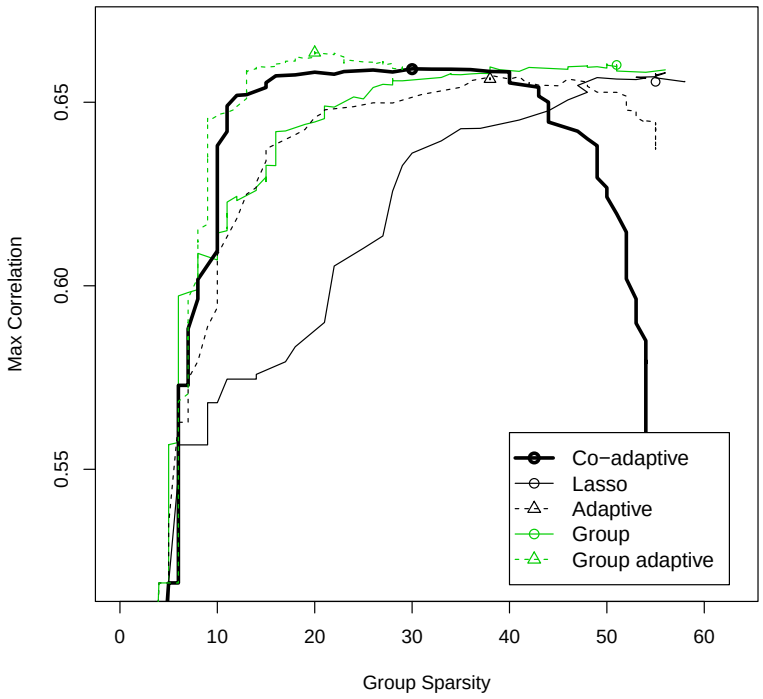

Figure 6.4.2: Group sparsity level $|\mathcal{G}_{\cap S}|$ vs $\text{cor}_{\max}$ on the test set. The points on each curve represents the model chosen by reference to the validation set.

each value of tuning parameter, the sparsity level, the group sparsity level, and the performance on the test dataset. We see that while the Group Lasso and Adaptive Group Lasso perform well overall, they require a much large set of selected variables to do so. The Lasso and the Adaptive Lasso meanwhile select a small number of variables, but fail in terms of attaining good results at any level of group sparsity. The Co-adaptive Lasso therefore managed to attain an excellent trade-off, by attaining much better within group sparsity, while still remaining competitive with the Adaptive Lasso in achieving good group sparse solutions. Indeed, for the most group sparse solutions, it even beats the Group Lasso.

Algorithm	$\text{cor}_{\max}$
Co-adaptive Lasso	0.659
Lasso	0.656
Adaptive Lasso	0.656
Group Lasso	0.660
Adaptive Group Lasso	0.664

Table 6.1: Maximum correlation for splice site test dataset.

Similar results were obtained when indicator variables that include $I_{\{\text{Pos}j=\text{'}T\text{'}\}}$ were used to create an intentionally over-specified design matrix. In our experiments, the Lasso based algorithms were the fastest, though the majority of time was consumed in handling the large dataset. It is likely that an online formulation would have been more efficient.

6.4.2 Artificial datasets

6.4.2.1 Comparison studies

We set up a series of simulation experiments to test the Co-adaptive Lasso against a range of similar algorithms. In particular, we compare against

- The Lasso [Tibshirani 1996]
- The Adaptive Lasso [Zou 2006]
- The Group Lasso [Meier et al. 2008]
- The Sparse Group Lasso [Friedman et al. 2010].

We generate a range of scenarios showing group sparsity and varying levels of within group sparsity in the linear regression context. Specifically, we generate

$$Y = X\beta + \varepsilon,$$

with X as a $n \times p$ matrix from an i.i.d. standard normal distribution. Then, choosing non-overlapping groups with group size $|G|$, we choose S as random subsets of 2 selected groups, with the same within group sparsity level in each group. We repeat each scenario with the final selected variables β_S as either independently standard normal or constant at 1. Finally, we generate the noise ε as independent normal, with variance chosen so that each iteration would have a SNR of 2.

Our scenarios are:

1. $n = 150$, $p = 2000$, $|G| = 10$, $|S| = 10$
2. $n = 150$, $p = 2000$, $|G| = 10$, $|S| = 20$
3. $n = 150$, $p = 2000$, $|G| = 100$, $|S| = 10$

	Scenario 1		Scenario 2		Scenario 3	
	Const	Norm	Const	Norm	Const	Norm
Lasso	2.81	2.11	16.2	7.64	2.84	1.88
Adaptive	1.35	1.14	15.87	5.48	1.38	**1.11**
Group Lasso	1.09	1.09	**2.25**	2.09	8.45	7.50
SGL	1.84	1.39	10.05	5.09	2.09	1.42
Co-adaptive	**0.55**	**0.53**	3.51	**1.21**	**1.35**	1.40

	Scenario 4		Scenario 5	
	Const	Norm	Const	Norm
Lasso	2.10	1.74	0.57	0.50
Adaptive	1.08	1.01	**0.21**	**0.20**
Group Lasso	0.62	0.58	2.12	1.93
SGL	1.47	1.20	0.41	0.34
Co-adaptive	**0.37**	**0.34**	0.36	0.39

Table 6.2: Results for simulated datasets. The table shows estimation error. The best performer in each scenario is highlighted in bold.

4. $n = 500$, $p = 2000$, $|G| = 10$, $|S| = 20$

5. $n = 500$, $p = 2000$, $|G| = 100$, $|S| = 10$.

Hence, Scenario 2 and 4 are AIAO situations, while the remaining scenarios have some degree of within group sparsity.

We conduct 100 iterations of each scenario, and compute estimates choosing tuning parameters by 10 fold cross validation. In the case of the SGL, which has 2 parameters, we use the default implementation which fixes the mixing parameter α at 0.95. We compare the estimation error $\left\|\widehat{\beta} - \beta\right\|^2$ – given the independence of our covariates, this is equivalent to comparing the expected error on a new test set, minus the contribution from the new ε. Table 6.2 shows the results of our simulations.

From these results, the Co-adaptive Lasso performs well in all of the scenarios. In most, it performs the best, or nearly the best, with the exceptions of scenario 5 and scenario 2, constant version. In the former case, understandably it performs less well than the Adaptive Lasso because the groups in this case are not very informative, while the sample size is large so the initial Lasso provides good weights for the adaptive second stage. In the latter case, there is no within group sparsity, and the inherent L2 penalty of the Group Lasso helps encourage it to make an estimate that is more constant in magnitude within groups, which matches the true signal. In comparison, the SGL, which is the other algorithm attempting within group sparsity, performs

roughly in between the Group Lasso and the Lasso. It needs to be noted that this behaviour was observed using the default values for the second tuning parameter - more success might be reached by selecting this, for instance, through a grid based cross validation. However such an approach would necessarily greatly increase the time require for computation.

In terms of computational time, the Lasso based methods were much faster than the Group Lasso or SGL. However, this may be due to the specific implementation of the algorithms.

6.4.2.2 Changing sparsity level

We also conduct some experiments to see, systematically, the effect of changing the level of within-group sparsity on the estimate, compared to some other algorithms.

Fix $n = 250$, $p = 2000$, and $\mathcal{G}$ such that each G are the consecutive indices

$$\{1, \dots, 10\}, \{11, \dots 20\}, \dots,$$

and generate $X \in \mathbb{R}^{n \times p}$ as $N(0, 1)$.

For each $k = 1, \dots, 10$, we choose S to be the first k indices in each of the first 5 groups, so that $|S| = 5k$. β_S are each generated as $N(0, 1)$, and $Y \sim N(X\beta, 1)$.

The graph is compiled as the average of 50 iterations, with λ chosen by 10-fold CV. We compare Co-adaptive Lasso to the Adaptive Lasso, the Adaptive Group Lasso, and the Group Lasso, measuring the estimation error.

Figure 6.4.3 gives the results of this. We see that the Co-adaptive Lasso generally performs well. In very sparse situations, including the worst case scenario of only one non-zero coefficient per non-zero group, it is second best, behind the Adaptive Lasso, while in the AIAO situation it is just slightly worse than the Adaptive Group Lasso. However, between $k = 3$ and $k = 9$, it is in fact the best performing method.

6.4.2.3 Correlated covariates

All of our artificial experiments so far have been with independent covariates. However, this may unrealistic in practice. We now generate a correlated dataset to see how vulnerable the Co-adaptive procedure is to correlation.

We again fix n, p, $\mathcal{G}$ as before, and this time fix $k = 5$ in our generation of β. Once again, given X, β, $Y \sim N(X\beta, 1)$. However, we generate X to have a correlation structure.

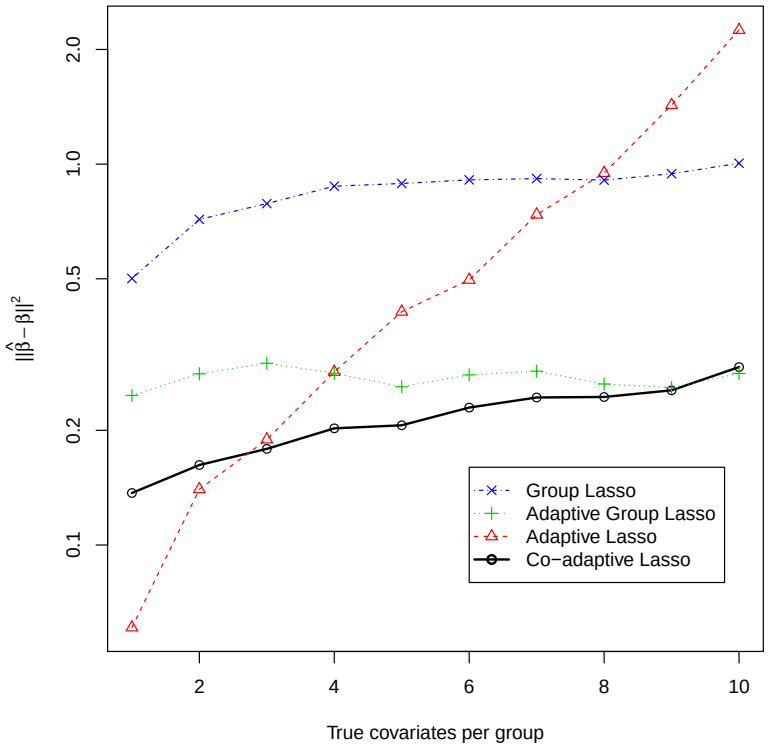

Figure 6.4.3: Estimation performance for various algorithms as a function of per-group sparsity. Each $|G| = 10$, so 10 true covariates in a group corresponds to AIAO.

Take any j, with $j \in G_k$, say. Then over a grid of ρ ranging from 0 to 1.25, we have

$$X_{i,j} = (U_{i,j} + \rho V_{i,k})/\sqrt{1+\rho^2}.$$

Here $U_{\cdot,j}$ and $V_{\cdot,k}$ are each $N(0,1)$ random variables corresponding to individual and grouped random effects. The sharing of V amongst covariates in the same group, then, creates a correlation between covariates in the same group, and we scale so that marginally each covariate is standard normal. Our procedure therefore creates a correlation that ranges from 0, to $1.25^2/(1+1.25^2) \approx 0.61$. We also produce a scenario where the correlation structure has no relation to the grouping structure, by randomly permuting the columns of X.

We do the same comparison in the previous section on these different scenarios, varying ρ. This provides the results in Figure 6.4.4. As we can see, the Co-adaptive Lasso maintains its relative superiority. Curiously, its performance, and those of others, deteriorates more for within-group correlation than randomised correlation. It is not clear why this is the case – perhaps this is due to the difficulty in determining

the correct β_S, and the increased likelihood of a solution that attains a similar $X\beta$ but has a smaller $\|\beta\|_1$.

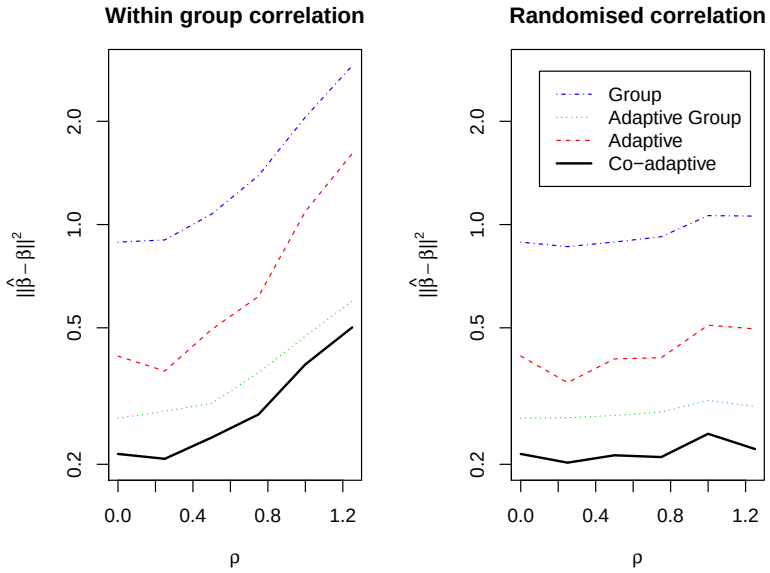

Figure 6.4.4: Estimation performance for various algorithms as a function of ρ, which corresponds to correlation.

6.5 Co-adaptive Rule Ensembles

We have discussed linear methods, and also additive modelling methods with the Liso. However, in many problems, neither of these methodologies are sufficient. The response's relation to the covariates may simply be too complex to be characterised in this way, with both non-linearity and interactions between multiple covariates present. Dealing with these situations is very difficult, especially in the case of high dimensionality - simply, the space of possible models would be huge, and we do not have enough data to distinguish between them.

In the case where the number of variables is small, tree based methods have seen widespread use. In these methods, a classifier or regression function is built as a set of repeated partitions of the covariate space, such as in Figure 6.5.1. Predictions from the tree are then made by feeding the new covariates associated with a new sample through the tree, choosing at each node according to the split at it until we arrive at a terminal node, which gives the prediction.

Breiman et al. [1984] proposes a set of techniques, Classification and Regression Trees (CART), for building such trees. These broadly involve growing trees through recursive partitioning, and then pruning to shrink trees, so as to minimise some type of

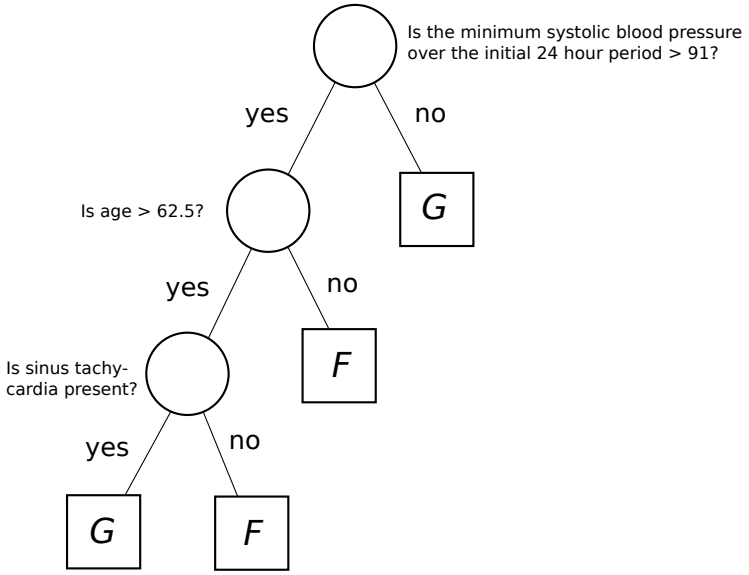

Figure 6.5.1: An example of a classification tree. (Recreated from Breiman et al. [1984].)

criterion, such as cross validation error. An important variant of this approach is the Random Forest [Breiman 2001], in which multiple trees are grown using randomised versions of the data. These are generally grown without pruning, and additional randomisation is provided by randomly selecting a subset of covariates to optimise over at each node. Then, results are given as the aggregation of answers from each tree, for instance through a majority voting rule. Random Forests are popular and successful, but the large number of trees involved leads to loss of interpretability of results, and they also frequently require a large amount of data to be effective.

Previous authors have proposed Node Harvest [Meinshausen 2010] and Rule Ensembles [Friedman and Popescu 2008] to incorporate ideas from sparse learning, and the Lasso in particular, to solve this problem. In the case of Node Harvest, we treat the individual trees built during the Random Forest procedure as covariates, and use the Lasso to find optimal combinations of small numbers of trees to predict the result. The result is a simpler version of the forest built by Random Forest, that frequently performs just as well, if not better, than the original, averaged result. In the case of Rule Ensembles, however, we do not consider the trees as trees at all, but instead as collections of 'rules'. The Random Forest procedure, then, as well as a number of other procedures, are each considered as a method of generating collections of candidate rules, which are selected from and fit using a Lasso algorithm. In this section, we shall explain in more detail what this involves, and examine the possibility of extending the work done in Friedman and Popescu [2008] with co-adaptive style reweighting.

We adopt the definitions used in Friedman and Popescu [2008]. We define a rule, as a set of restrictions, which defines a hypercube in the covariate space.

Definition 6.5.1. For any j, let s_{kj}, $k = 1, \ldots, p$ be each a subset of all the possible values of $X_{\cdot,k}$, and in particular an interval of the form $(\inf s_{kj}, \sup s_{kj}]$ if $X_{\cdot,k}$ takes continuous values. This subset may encompass the entire set of possible values. In any case, the corresponding rule $r_j(x)$ is defined as

$$r_j(x) = \prod_{k=1}^{p} I_{\{x_k \in s_{kj}\}}.$$

The insight of Friedman and Popescu [2008] is to consider a decision tree as a base learner for the generation of such rules. This is done by considering the restrictions represented by each node. For example, in Figure 6.5.1, the bottom left terminal node has the associated rule:

$$r_j(x) = I_{\{x_{\text{systolic}} > 91\}} I_{\{x_{\text{age}} > 62.5\}} I_{\{x_{\text{sinus}} > 0\}}.$$

Note that we consider as rules not just the terminal nodes, but the interior nodes as well. For example, $r_j(x) = I_{\{x_{\text{systolic}} > 91\}}$ is also included. Specifically, for a tree with t_m terminal nodes, there would be $2(t_m - 1)$ such rules. From a tree learner like the Random Forest, then, we may generate a ensemble of $q = \sum_{m=1}^{|\mathcal{T}|} 2(t_m - 1)$ rules, where T is the set of trees in the Random Forest model. Clearly, there can be a very large number of potential rules generated.

Friedman and Popescu [2008] suggest the use of the Lasso to resolve this selection problem. Writing the predictive model as

$$f(x) = \beta_0 + \sum_{j=1}^{q} \beta_j r_j(x),$$

we estimate $\beta_0 \ldots \beta_q$ as $\widehat{\beta}_0 \ldots \widehat{\beta}_q$ solving

$$\widehat{\beta} = \arg\min_{\beta} \left\| Y - \beta_0 - \sum_{j=1}^{q} \beta_j r_j(X) \right\|^2 + \lambda \sum_{j=1}^{q} |\beta_j|.$$

In other words, we solve a Lasso problem using the expanded design matrix

$$\widetilde{X}_{i,\cdot} = \left(r_1(X_{i,\cdot}) \ldots r_q(X_{i,\cdot}) \right), \quad i = 1, \ldots n.$$

It is possible to rescale $\widetilde{X}$ to give normalised predictors, though Friedman and Popescu [2008] advise against such practice, so as to penalise rules with very small support. The authors discuss further details on rule generation, and suggest the inclusion of a linear term to fit better linear dependencies that would be difficult for tree-like models to fit. They term this overall procedure 'RuleFit', and find that it behaves fairly well in experiments.

However, we note that the RuleFit procedure misses one piece of information. This is that for each rule, there is a small subset of the covariates that determine whether or not a particular sample matches that rule or not, which is exactly those k for which s_{kj} is not equivalent to the full set of possible values. In complex, high dimensional systems, it may be the case that many of the original variables are irrelevant to the model. Together, these two facts suggest a grouping structure, and since the rules are built from the training data, this can be expressed as:

Let $\mathcal{G} = \{G_1, \ldots, G_p\}$, with

$$j \in G_k \quad \text{iff} \quad \exists X_{i,k} \notin s_{kj}.$$

Due to the random nature of building rules under Random Forest, the more complex rules can involve very large numbers of variables, and so would be present in a large number of groups. Hence, an union based weight calculation may not be very useful. Instead, we can attempt to reduce the number of variables selected by the algorithm by using a calculation based on the least significant of the groups corresponding to each rule. One method of doing this is to calculate weights for each rule as the following:

$$w_j^{-1} = \min_{G \ni j} \sqrt{\sum_{l \in G} \beta_l^2}. \tag{6.5.1}$$

Notice that we do not include an adjustment for group size as we do in the Co-adaptive weight calculations earlier, so as to penalise very small groups.

Using this weight calculation, we arrive at the Co-adaptive RuleFit, which we define as conducting the RuleFit procedure, and then, using the same rule ensemble, conducting a second weighted lasso fit,

$$\widehat{\beta}^{(c)} = \arg\min_{\beta} \left\| Y - \beta_0 - \sum_{j=1}^{q} \beta_j r_j(X) \right\|^2 + \mu \sum_{j=1}^{q} w_j |\beta_j|,$$

and thus giving the estimate

$$f(x) = \widehat{\beta}_0^{(c)} + \sum_{j=1}^{q} \widehat{\beta}_j^{(c)} r_j(x).$$

As with the ordinary Co-adaptive Lasso, cross validation is effective for choosing the correct tuning parameters. In this case, particular care should be taken that for each cross-validation fold, a separate rule ensemble is computed using the available training data for that fold, and not using the set $\{r_j\}$ from the full data set X, Y. This is because with most tree builders, the r_j's constructed will have a dependence on the full X, Y, not just the $X_{i,\cdot}, Y_i$ with i not in the hold out set, and the cross validation procedure will therefore underestimate the variance arising from the rule building procedure.

Note that a broader range of possible ways to use the Co-adaptive Lasso with rule ensembles exist. We use the procedure with weights as in Equation (6.5.1) because of its simplicity. Nevertheless, provided that the rule ensemble $\{r_j\}$ generated contains many rules involving irrelevant variables, we expect, and see in empirical experiments, substantial improvements over the ordinary RuleFit. The following example is an illustration:

We produce an artificial dataset similar to that in Friedman and Popescu [2008]. Here, $n = 500$, $p = 500$, and X is generated to be discrete, with each $X_{\cdot,k}$ being independently drawn with replacement from $\{0, 1/10, \ldots, 9/10\}$. We generate Y as $Y \sim N(f(X), \sigma^2)$, with

$$f(X_{i,\cdot}) = 9 \exp\left(-3 \sum_{k=1}^{2} (1 - X_{i,k})^2\right) + 2 \sin(\pi X_{i,3})^2,$$

and σ chosen so that the signal to noise ratio, $SNR = 2$.

Notice that with this formulation, $X_{\cdot,1}$ and $X_{\cdot,2}$ have a strong interaction, while $X_{\cdot,3}$ has an individual additive effect. The remaining 497 covariates have no impact on the response.

To build our classifier, we run a Random Forest on the dataset, with 333 trees and each tree to a maximum depth of 4, with each tree built on a bootstrap sample of 234 samples. These specifications are based on those in Friedman and Popescu [2008]. Decomposing the forest thus obtained, we produce an expanded design matrix $\widetilde{X}$ with 1998 columns. We conduct a RuleFit procedure with this, using 10-fold cross validation to find the correct tuning parameter. Then we refit using our Co-adaptive RuleFit procedure. Note that we deviate slightly from the procedure from the original RuleFit paper in that we do not add on a set of linear covariates. This is so that the

problem is more difficult, as linear fits are especially hard to accommodate using the discontinuous basis implied by our Rule Ensemble, as we have seen with the earlier Liso examples.

After repeating for 100 iterations, we find that the Co-adaptive procedure obtains an average prediction error rate $E(Y-\widehat{Y})^2$ of 0.40, which compares favourably to that obtained for the RuleFit of 0.77. Figure 6.5.2 gives an example fit for this procedure. We see that we achieve smaller bias on the conditional fits on the true covariates 1, 2, and 3, though this effect is fairly subtle. We also make a significantly smaller, and more smooth, erroneous fit on the two most selected irrelevant covariates.

The achievement of our procedure is clearer on the diagnostic plots shown in Figure 6.5.3. In terms of the response versus fitted values curve, we obtain a closer relationship between the estimated $\widehat{Y} = \widehat{f}(X)$ and the true one. In the second plot, we examine variable importance, which we measure by comparing the variance of $f(X)$ upon re-sampling $X_{\cdot,k}$, for each covariate k in turn, as a fraction of the total empirical variance of Y. In other words, we construct an analogue of the 'proportion of variance explained' by each covariate. With this, we see clearly that the CoadRuleFit procedure has been more successful at removing the spurious covariates, and explains more of the variability using the three true covariates.

The helpfulness of the procedure does depend on the rule ensemble constructed. It may be the case that the rules used as an input to the Lasso as part of the RuleFit involve only the true covariates. In that case, the procedure we have defined will be unhelpful, since we do not remove any covariates. Overall though, the CoadRuleFit procedure seems to be an useful extension.

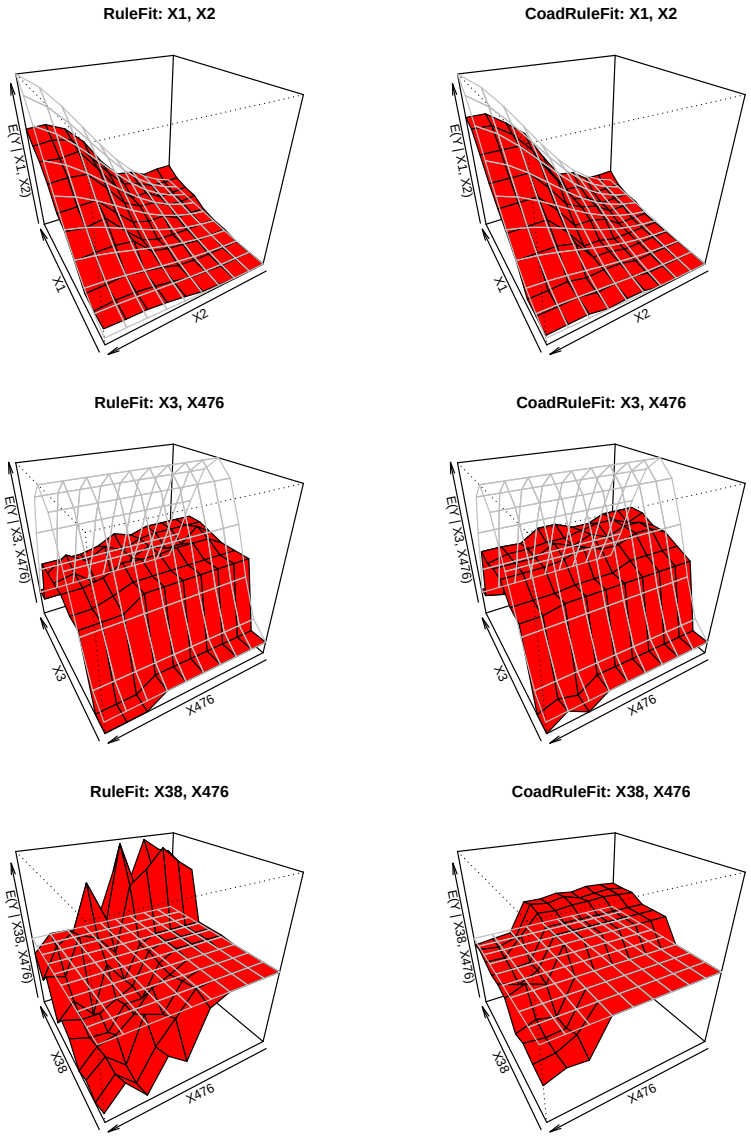

Figure 6.5.2: Two variable conditional fits for RuleFit and CoadRuleFit. For each plot, the red surface gives the conditional expectation for Y given the particular variables, as estimated by each algorithm. The overlaid grey grid gives that as implied by the model used to generate the data. We consider variables X_1, X_2, X_3, each present in the original model, and X_{476}, the non-present covariate with the most important fit in both algorithms.

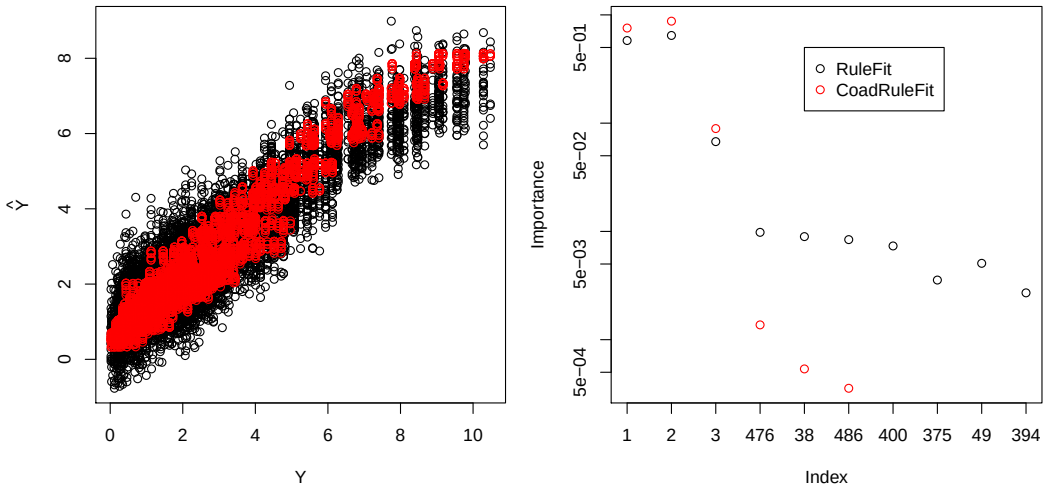

Figure 6.5.3: Diagnostic graphs for RuleFit vs CoadRuleFit. The first plot shows response versus fitted values on a new test dataset. The second shows variable importance for the true covariates, plus the 7 irrelevant covariates that are found to be most important for RuleFit. Note that the CoadRuleFit removed 4 of these from the full model, and selected no other covariates, while RuleFit selected a total of 121 covariates, and so 118 irrelevant ones.

Chapter 7

Summary

In this thesis, we have defined a fast calculating method of conducting variable selection under a group structure, that facilitates the use of within group sparsity. The Co-adaptive Lasso can be easily coded using any existing methods of calculating the standard Lasso, including online computation methods. We have proven some convergence properties for the method, and illustrated its competitiveness relative to several state of the art methods. The procedure may be applied to a range of contexts.

En route, we have defined the Liso estimator for high dimensional isotonic regression, and provided methods for its calculation. We have shown how the co-adaptive reweighting ideas can work to improve this method, and allow a variant of it to be used in the additional application of detection of monotonicity. We have also constructed the CoadRuleFit algorithm, and shown by example its usefulness in enhancing the current RuleFit methodology. A range of further approaches were also discussed, including repeating the estimator, winsorised weight computations, and analyses for overlapping groups.

Several areas warrant further investigation:

Theoretical questions remain about the Co-adaptive Lasso. Our focus in the theoretical work has been on prediction error and estimation error, but group selection and variable selection are also key aspects that are useful in practical application of such methods. Empirically, we see that the Co-adaptive Lasso does well in both setting $\widehat{\beta}_G^{(c)} = 0$ for $\beta_G = 0$, and in obtaining correct sparsity patterns within groups, and so work may be done on showing this from a theoretical viewpoint. Additional work is also suggested in checking how often the various eigenvalue conditions are fulfilled in practice. Empirical evidence again indicates that these are of similar strength to those relevant Restricted Eigenvalue conditions required for the Lasso

itself to succeed. Finally, the bounds we have shown may not be very tight. As our work with winsorised weights implies, the Co-adaptive Lasso can outperform what is suggested substantially with a minor modification, albeit in the fairly restrictive case of randomly assigned $\mathcal{G}^c_{\cap S}$. It is likely this generalises to a wider range of grouping structures where the distribution of first stage false positives is quite uniform, and so it would be interesting to investigate further.

We have provided a range of methods, including the Online Co-adaptive Lasso, the CoadRuleFit and the Adaptive Liso, but while we have shown performance bounds for the Co-adaptive Lasso at the heart of each procedure, and presented some empirical studies, further research is required in showing theoretical properties for these derivative algorithms. In the case of the CoadRuleFit, one might observe that it presents a version of overlapping group structures, and it would be interesting to see how that interacts with the theorems we have proven in Section 5.

From the implementation side, the experience we have in computing the various simulations in this thesis suggest that the Co-adaptive procedures are typically very competitive with other methods. In most cases, Co-adaptive calculations are second only to the Lasso, and equal with the Adaptive Lasso in processing, and certainly much faster than Group Lasso based methodologies. Further research may more carefully pin down the difference in computational requirements from a computational complexity point of view, and determine whether this is inherent to the methodology, or just an artifact of the implementation. Additional work may also improve computational efficiency – for instance, more effective warm starting procedures, methods for prioritising covariates to update in coordinate descent, or even hybrid algorithms, may lead to substantial improvements. Inspired by the work of Zou et al. [2007], we may avoid cross validation altogether by following some variant of the SURE procedure, computing some analogue of the effective degrees of freedom for the Co-adaptive Lasso. This would reduce the computational requirements, and even potentially improve performance.

These open questions are very broad, and outside of the scope of the thesis. However, they may be useful for future investigation, and aid in developing a more complete understanding of the usefulness of the methods established and how best they may be applied.

References

Arlot, S. and Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79.

Ayer, M., Brunk, H. D., Ewing, G. M., Reid, W. T., and Silverman, E. (1955). An empirical distribution function for sampling with incomplete information. *The Annals of Mathematical Statistics*, 26:641–647.

Bacchetti, P. (1989). Additive isotonic models. *Journal of the American Statistical Association*, 84:289–294.

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

Barlow, R. E., Bartholomew, D. J., Bremner, J. M., and Brunk, H. D. (1972). *Statistical inference under order restrictions: the theory and application of isotonic regression*. Wiley.

Bickel, P. J., Ritov, Y., and Tsybakov, A. B. (2009). Simultaneous analysis of lasso and dantzig selector. *Annals of Statistics*, 37:1705–1732.

Breheny, P. and Huang, J. (2009). Penalized methods for bi-level variable selection. *Statistics and its Interface*, 2:369–380.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24:123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Breiman, L., Friedman, J. H., Stone, C. J., and Olshen, R. (1984). *Classification and Regression Trees*. Chapman and Hall.

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

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

Friedman, J. H. (1991). Multiple adaptive regression splines. *Annals of Statistics*, 19:1–141.

Friedman, J. H., Hastie, T. J., Höfling, H., and Tibshirani, R. (2007). Pathwise coordinate optimization. *Annals of Applied Statistics*, 1:302–332.

Friedman, J. H., Hastie, T. J., and Tishirani, R. (2010). A note on the group lasso and a sparse group lasso. arXiv preprint [1001.0736].

Friedman, J. H. and Popescu, B. E. (2008). Predictive learning via rule ensembles. *Annals of Applied Statistics*, 2:916–954.

Fu, W. (1998). Penalized regressions: the bridge vs the lasso. *Journal of Computational and Graphical Statistics*, 7:397–416.

Harrison, D. and Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5:81–102.

Hastie, T. J., Tibshirani, R., and Friedman, J. H. (2003). *The Elements of Statistical Learning*. Springer.

Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized Additive Models*, chapter 4, pages 82–95. Monographs on statistics and applied probability. Chapman and Hall.

Huang, J., Ma, S., and Zhang, C.-H. (2008). The iterated lasso for high-dimensional logistic regression. Technical Report 392, University of Iowa.

Huang, J. and Zhang, T. (2010). The benefit of group sparsity. *Annals of Statistics*, 38:1978–2004.

Huang, J., Zhang, T., and Metaxas, D. (2009). Learning with structured sparsity. In *ICML '09 Proceedings of the 26th Annual International Conference on Machine Learning*.

Itô, K. (1993). *Encyclopedic Dictionary of Mathematics*, chapter 166, pages 642–643. MIT Press.

Jacob, L., Obozinski, G., and Vert, J.-P. (2009). Group lasso with overlap and graph lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220:671–680.

Lange, K., Hunter, D. R., and Yang, I. (2000). Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9:21–25.

Leng, C., Lin, Y., and Wahba, G. (2006). A note on the lasso and related procedures in model selection. *Statistica Sinica*, 16:1273–1284.

Lounici, K., Pontil, M., Tsybakov, A. B., and van de Geer, S. (2010). Oracle inequalities and optimal inference under group sparsity. arXiv preprint [1007.1771].

Mammen, E. and van de Geer, S. (1997). Locally adaptive regression splines. *Annals of Statistics*, 25:387–413.

Mammen, E. and Yu, K. (2007). Additive isotone regression. In *Asymptotics: Particles, Processes and Inverse Problems*, volume 55 of *IMS Lecture Notes - Monograph Series*, pages 179–195. IMS.

Meier, L., van de Geer, S., and Bühlmann, P. (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society B*, 70:53–71.

Meier, L., van de Geer, S., and Bühlmann, P. (2009). High-dimensional additive modeling. *Annals of Statistics*, 37:3779–3821.

Meinshausen, N. (2010). Node harvest. *Annals of Applied Statistics*, 4:2049 – 2072.

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

Osborne, M. R., Presnell, B., and Turlach, B. A. (1998). Knot selection for regression splines via the lasso. In Weisberg, S., editor, *Dimension Reduction, Computational Complexity, and Information*, Proceedings of the 30th Symposium on the Interface, Interface 98, pages 44–49. Interface Foundation of North America.

Osborne, M. R., Presnell, B., and Turlach, B. A. (2000). A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20:389–404.

Percival, D. (2011). Theoretical properties of the overlapping groups lasso. Submitted to Annals of Statistics.

Ravikumar, P., Liu, H., Lafferty, J., and Wasserman, L. (2007). Spam: Sparse additive models. *Advances in Neural Information Processing Systems (NIPS)*.

Roth, V. and Fischer, B. (2008). The group-lasso for generalized linear models: Uniqueness of solutions and efficient algorithms. In *Proceedings of the 25th International Conference on Machine Learning*.

Shalev-Shwartz, S. and Tewari, A. (2009). Stochastic methods for l1 regularized loss minimization. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 929–936, New York, NY, USA. ACM.

Stein, C. M. (1981). Estimation of the mean of a multivariate normal distribution. *Annals of Statistics*, 9:1135–1151.

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

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

Tibshirani, R. J., Hoefling, H., and Tibshirani, R. (2010). Nearly-isotonic regression. In preparation.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109:475–494.

van de Geer, S., Bühlmann, P., and Zhou, S. (2010). The adaptive and the thresholded lasso for potentially misspecified models. arXiv preprint [1001.5176].

van de Geer, S. A. and Bühlmann, P. (2009). On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360 – 1392.

Wainwright, M. (2009). Sharp thresholds for high-dimensional and noisy recovery of sparsity. *IEEE Transactions on Information Theory*, 55:2183–2201.

Wang, H. and Leng, C. (2008). A note on adaptive group lasso. *Journal of Computational Statistics and Data Analysis*, 52:5277 – 5286.

Wu, T. T., Chen, Y. F., Hastie, T. J., Sobel, E., and Lange, K. (2009). Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25:714–721.

Xiao, L. (2010). Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596.

Yang, H., Xu, Z., King, I., and Lyu, M. R. (2010). Online learning for group lasso. In *Proceedings of the 27th International Conference on Machine Learning*.

Yeo, G. W. and Burge, C. B. (2008). Maximum entropy modeling of short sequence motifs with applications to ma splicing signals. *Journal of the Royal Statistical Society B*, 70:53–71.

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

Zhang, C.-H. (2007). Penalized linear unbiased selection. Technical report, Rutgers University.

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

Zhao, P., Rocha, G., and Yu, B. (2009). The composite absolute penalties family for grouped and hierarchical variable selection. *Annals of Statistics*, 37:3468–3497.

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

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

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

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

Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood methods. *Annals of Statistics*, 36:1509–1533.

Appendix A

Proofs of theorems

A.1 Proofs for Chapter 3

Proof of Theorem 3.3.1

Our methodology is to show that adding boundary constraints to constrained or unconstrained isotonic regression problems result in unique solutions that are simply Winsorised PAVA estimates, and then demonstrate a method of constructing any LISO solution, in the univariate case, though boundary constraints. Differentiation then allows us to find what these boundary constraints are.

We prove first the following lemma, which provides an induction step in our eventual argument:

Lemma A.1.1. *Suppose for $A \leq B$, $X = (X_1, \ldots, X_n), Y = (Y_1, \ldots, Y_n)$, $X_i \in \mathbb{R}$ for all i, the Winsorised PAVA,*

$$f_1(x) = \widehat{f}_{>A,<B}(x) := \begin{cases} A & \text{if } \widehat{f}_{PAVA}(x) < A \\ B & \text{if } \widehat{f}_{PAVA}(x) > B \\ \widehat{f}_{PAVA}(x) & \text{otherwise.} \end{cases}$$

solves the boundary constrained isotonic regression problem,

$$\min_f \|Y - f(X)\|^2 \quad \text{such that } f \text{ monotone, } A \leq f(x) \leq B, \quad \forall x. \tag{A.1.1}$$

Then for $A \leq A' \leq B' \leq B$, $f_2 \equiv \widehat{f}_{>A',<B'}$ *solves the further constrained isotonic regression problem*

$$\min_f \|Y - f(X)\|^2 \quad \textit{such that } f \textit{ monotone, } A' \leq f(x) \leq B', \quad \forall x. \tag{A.1.2}$$

Further, this solution is unique, in terms of its fitted values $f(X_1), \ldots, f(X_n)$.

Proof of Lemma A.1.1. It suffices to prove the lemma for the case of $A \leq A' \leq B' = B$, since the argument for $A = A' \leq B' \leq B$ is identical, and we can proceed to the full Lemma by adding the top and bottom constraints one by one.

Note that, specifying f through the fitted values $f(X_1), \ldots, f(X_n)$, $\|Y - f(X)\|^2$ is strictly convex when considered as a function of f, and the constraints give a convex feasible set. Hence, for any combination of A, B, solutions must exist and be unique at $X_1, \ldots, X_n$.

Therefore, let g be the solution of the optimisation (A.1.2), for $A \leq A' \leq B' = B$. Suppose for contradiction that $g \not\equiv f_2 \equiv \widehat{f}_{>A',<B'}$.

Let u_f, u_g be the points where the functions f_2, g respectively exceed A'.

$$u_f = \inf \{X_i \text{ s.t. } f_2(X_i) > A'\}$$
$$u_g = \inf \{X_i \text{ s.t. } g(X_i) > A'\}.$$

Then we have two cases.

(a) If $u_f \leq u_g$, then consider a new function $\widetilde{f}$ where

$$\widetilde{f}(x) = \begin{cases} f_1(x) & \text{if} \quad x < u_f \\ g(x) & \text{if} \quad x \geq u_f. \end{cases} \tag{A.1.3}$$

$\widetilde{f}$ would be an increasing function satisfying the conditions of (A.1.1). Because $g \not\equiv f_2$, and g and f_2 are both equal to A' when restricted to $\{x : x < u_f\}$, it must be the case that $g \not\equiv f_2 \equiv f_1$ when restricted to $\{x : x \geq u_f\}$. Therefore, $\widetilde{f} \not\equiv f_1$. The

residual sum of squares is then, applying (A.1.2) optimality of g,

$$\begin{aligned}
\left\|Y - \widetilde{f}(X)\right\|^2 &= \sum_{X_i<u_f} (Y_i - f_1(X_i))^2 + \sum_{X_i\geq u_f} (Y_i - g(X_i))^2 \\
&= \sum_{X_i<u_f} (Y_i - f_1(X_i))^2 + \|Y - g(X)\|^2 - \sum_{X_i<u_f} (Y_i - A')^2 \\
&\leq \sum_{X_i<u_f} (Y_i - f_1(X_i))^2 + \|Y - f_2(X)\|^2 - \sum_{X_i<u_f} (Y_i - A')^2 \\
&= \sum_{X_i<u_f} (Y_i - f_1(X_i))^2 + \sum_{X_i\geq u_f} (Y_i - f_1(X_i))^2 \\
&= \|Y - f_1(X)\|^2 .
\end{aligned}$$

Therefore, $\widetilde{f}$ is optimal for (A.1.1). This contradicts uniqueness and (A.1.1) optimality of f.

(b) If $u_f > u_g$, then if we define $\widetilde{f}$ this time as

$$\widetilde{f}(x) = \begin{cases} f_1(x) & \text{if} \quad x < u_g \\ g(x) & \text{if} \quad x \geq u_g, \end{cases} \tag{A.1.4}$$

we obtain another increasing function satisfying the conditions of (A.1.1). As before, because we have assumed that $g \not\equiv f_2$, and yet $g \equiv f_2 \equiv A'$ for $\{x : x < u_g\}$, $g \not\equiv f_2 \equiv f_1$ when restricted to $\{x : x \geq u_g\}$. This means that $\widetilde{f} \not\equiv f_1$, so unique optimality of f_1 versus $\widetilde{f}$ means that

$$\begin{aligned}
\sum_{X_i\geq u_g} (Y_i - f_1(X_i))^2 &= \|Y - f_1(X)\|^2 - \sum_{X_i<u_g} (Y_i - f_1(X_i))^2 \\
&< \left\|Y - \widetilde{f}(X)\right\|^2 - \sum_{X_i<u_g} (Y_i - f_1(X_i))^2 \\
&= \sum_{X_i\geq u_g} (Y_i - g(X_i))^2 .
\end{aligned}$$

In addition, because $u_f > u_g$, setting $\delta = (g(u_g) - A')/(g(u_g) - f_1(u_g))$ makes $\widetilde{g}$,

defined as

$$\widetilde{g}(x) = \begin{cases} A' & \text{if} \quad x < u_g \\ (1-\delta)g(x) + \delta f_1(x) & \text{if} \quad x \geq u_g, \end{cases} \tag{A.1.5}$$

an increasing function satisfying the conditions of (A.1.2). By definition, $g(u_g) > A'$ and $f_1(u_g) < A'$, so $\delta \in (0,1)$, implying that $\widetilde{g}$ is a nontrivial convex combination of g and f_1, when restricted to $\{x : x \geq u_g\}$.

But by convexity,

$$\begin{aligned} \|Y - \widetilde{g}(X)\|^2 &= \sum_{X_i < u_g} (Y_i - A')^2 + \sum_{X_i \geq u_g} (Y_i - (1-\delta)g(X_i) - \delta f_1(X_i))^2 \\ &< \sum_{X_i < u_g} (Y_i - A')^2 + \sum_{X_i \geq u_g} (Y_i - g(X_i))^2 = \|Y - g(X)\|^2 . \end{aligned}$$

This contradicts optimality of g.

Therefore, $g \equiv f_2$ is the unique solution to (A.1.2). □

The above lemma allows us to prove a fact about the structure of any LISO solution:

Lemma A.1.2. *For $A \leq B$, denote by $\widehat{f}_{>A,<B}$ the Winsorised PAVA estimate. Then if $p = 1$, there exist thresholds $A_\lambda \leq B_\lambda$ for each value of $\lambda \geq 0$ such that the LASSO-Isotone solution is given by $\widehat{f}_\lambda \equiv \widehat{f}_{>A_\lambda,<B_\lambda}$.*

Proof of Lemma A.1.2. We can prove Lemma A.1.2 as a simple corollary.

When $\lambda = 0$, our objective function $\ell_\lambda^{\text{liso}}$ is strictly convex and quadratic, and indeed is the same as the PAVA optimisation. Hence, an unique optimal solution exists, and is given by $\widehat{f}_0 = \widehat{f}_{PAVA}$. Set $A_0 = -\infty, B_0 = \infty$. Then $\widehat{f}_0 \equiv \widehat{f}_{>A_0,<B_0}$ is feasible for, and so, must also solve the constrained optimisation (A.1.1), with constraints at infinity.

For $\lambda > 0$, $\ell_\lambda^{\text{liso}}$ and the domain we maximise it in are both still convex, with $\ell_\lambda^{\text{liso}}$ strictly convex and increasing away from the origin outside of a neighbourhood. Therefore, an unique bounded solution $\widehat{f}_\lambda$ must exist. Set A_λ, B_λ to be the upper and lower bounds of this solution.

$$A_\lambda = \min_i \widehat{f}_\lambda(X_i), \quad B_\lambda = \max_i \widehat{f}_\lambda(X_i).$$

Then consider the solution to the constrained isotonic least squares problem

$$\widetilde{f}_\lambda = \arg\min_f \|Y - f(X)\|^2 \quad \text{such that } f \text{ monotone, } A_\lambda \leq f(x) \leq B_\lambda, \quad \forall x. \tag{A.1.6}$$

$\Delta(\widetilde{f}_\lambda) \leq B_\lambda - A_\lambda = \Delta(\widehat{f}_\lambda)$, and since $\widehat{f}_\lambda$ is feasible for A.1.6, $\left\|Y - \widetilde{f}_\lambda(X)\right\|^2 \leq \left\|Y - \widehat{f}_\lambda(X)\right\|^2$.

Therefore, $\widetilde{f}_\lambda$ is optimal for (2.1). Hence, by uniqueness, $\widetilde{f}_\lambda \equiv \widehat{f}_\lambda$, so for suitable A_λ, B_λ it suffices to solve the bound constrained least squares optimisation (A.1.6), to find the LISO solution.

But by Lemma A.1.1, the solution of (A.1.6) for $> A_\lambda < B_\lambda$ is just the Winsorised PAVA solution $\widehat{f}_{>A_\lambda,<B_\lambda}$, so $\widehat{f}_\lambda \equiv \widetilde{f}_\lambda \equiv \widehat{f}_{>A_\lambda,<B_\lambda}$. □

Proof of Theorem 3.3.1. Given Lemma A.1.2, it remains to show that A_λ, B_λ satisfy the conditions given in Theorem 3.3.1.

For $\lambda = 0$, $\widehat{f}_0 \equiv \widehat{f}_{PAVA}$, and $\min(\widehat{f}_{PAVA}(x))$, $\max(\widehat{f}_{PAVA}(x))$ satisfy (3.2) and (3.3). Assume therefore $\lambda > 0$.

Now, from Barlow et al. [1972], the unregularised PAVA solution $\widehat{f}_{PAVA}$, considered as a right continuous step function, is an example of a regressogram. In other words, there exists a partition into disjoint intervals of $\mathbb{R}$, $P_1, \ldots, P_m$, with the value of $\widehat{f}_{PAVA}(x)$ on each interval being the mean of the observed Y for X falling within the interval.

$$\widehat{f}_{PAVA}(x) = \frac{\sum_{i=1}^n Y_i I_{\{X_i \in P_j\}}}{\sum_{i=1}^n I_{\{X_i \in P_j\}}} = \widehat{f}_{PAVA}(P_j) \quad \text{for all } x \in P_j$$

Using $\widehat{f}_0 \equiv \widehat{f}_{PAVA}$, the LASSO criterion for the thresholded function $\widehat{f}_{>A,<B}$ then

can be written as,

$$
\begin{aligned}
\ell_\lambda^{\text{liso}}(\widehat{f}_{>A,<B}) &= \frac{1}{2}\left\|Y-\widehat{f}_{>A,<B}\right\|^2 + n\lambda(B-A) \\
&= \frac{1}{2}\sum_{i=1}^{n}\Big((Y_i-\widehat{f}_0(X_i))^2 I_{\{\widehat{f}_0(X_i)\in[A,B]\}} + (A-Y_i)^2 I_{\{\widehat{f}_0(X_i)<A\}} \\
&\quad +(Y_i-B)^2 I_{\{\widehat{f}_0(X_i)>B\}}\Big) + n\lambda(B-A) \\
&= \frac{1}{2}\left\|Y-\widehat{f}_0\right\|^2 + n\lambda(B-A) \\
&\quad + \frac{1}{2}\sum_{i=1}^{n}\Big((A-\widehat{f}_0(X_i))^2 + 2(A-\widehat{f}_0(X_i))(\widehat{f}_0(X_i)-Y_i)\Big) I_{\{\widehat{f}_0(X_i)<A\}} \\
&\quad + \frac{1}{2}\sum_{i=1}^{n}\Big((\widehat{f}_0(X_i)-B)^2 + 2(\widehat{f}_0(X_i)-B)(Y_i-\widehat{f}_0(X_i))\Big) I_{\{\widehat{f}_0(X_i)>B\}} \\
&= \frac{1}{2}\left\|Y-\widehat{f}_0\right\|^2 + \frac{1}{2}\sum_{i=1}^{n}\Big((A-\widehat{f}_0(X_i))_+^2 + (\widehat{f}_0(X_i)-B)_+^2\Big) + n\lambda(B-A) \\
&\quad + \sum_{j=1}^{m}\sum_{i=1}^{n}\Big((\widehat{f}_0(X_i)-B)(Y_i-\widehat{f}_0(X_i))I_{\{\widehat{f}_0(X_i)>B\}}\Big) I_{\{X_i\in P_j\}} \\
&\quad + \sum_{j=1}^{m}\sum_{i=1}^{n}\Big((A-\widehat{f}_0(X_i))(\widehat{f}_0(X_i)-Y_i)I_{\{\widehat{f}_0(X_i)<A\}}\Big) I_{\{X_i\in P_j\}} \\
&= \frac{1}{2}\left\|Y-\widehat{f}_0\right\|^2 + \frac{1}{2}\sum_{i=1}^{n}\Big((A-\widehat{f}_0(X_i))_+^2 + (\widehat{f}_0(X_i)-B)_+^2\Big) + n\lambda(B-A) \\
&\quad + \sum_{j=1}^{m}(\widehat{f}_0(P_j)-B)_+ \sum_{i=1}^{n}\Big((Y_i-\widehat{f}_0(X_i))\Big) I_{\{X_i\in P_j\}} \\
&\quad + \sum_{j=1}^{m}(A-\widehat{f}_0(P_j))_+ \sum_{i=1}^{n}\Big((\widehat{f}_0(X_i)-Y_i)\Big) I_{\{X_i\in P_j\}} \\
&= \frac{1}{2}\left\|Y-\widehat{f}_0\right\|^2 + \frac{1}{2}\sum_{i=1}^{n}\Big((A-\widehat{f}_0(X_i))_+^2 + (\widehat{f}_0(X_i)-B)_+^2\Big) + n\lambda(B-A).
\end{aligned}
$$

We seek a minimum to this with $A \le B$. Differentiating in A and B, and setting

equal to zero, gives,

$$\sum_{i=1}^{n}(\widehat{f}_0(X_i) - B)_+ = n\lambda \tag{A.1.7}$$

$$\sum_{i=1}^{n}(A - \widehat{f}_0(X_i))_+ = n\lambda. \tag{A.1.8}$$

Note that the left hand side in both cases is a piecewise linear, continuous and monotone (indeed, decreasing in B and increasing in A) function of the threshold, equalling zero for $A = A_0, B = B_0$. Further, since the PAVA is a regressogram, $\sum_{i=1}^{n}\widehat{f}_0(X_i) = \sum_{i=1}^{n} Y_i$, so

$$\sum_{i=1}^{n}(\widehat{f}_0(X_i) - \overline{Y})_+ = \sum_{i=1}^{n}(\overline{Y} - \widehat{f}_0(X_i))_+ = \frac{1}{2}\sum_{i=1}^{n}|\widehat{f}_{PAVA}(X_i) - \overline{Y}|.$$

We therefore have two cases. If

$$2n\lambda \leq \sum_{i=1}^{n}|\widehat{f}_{PAVA}(X_i) - \overline{Y}|,$$

then we can find solutions to (A.1.7) and (A.1.8) for which $A_\lambda \leq \overline{Y} \leq B_\lambda$, producing a minimiser for (2.1).

Otherwise, (A.1.7) and (A.1.8) require that $A > \overline{Y}$ and $B < \overline{Y}$. Hence, there are no solutions to our minimisation with $A < B$.

For a solution on the boundary, we require $A = B$, so

$$\ell_\lambda^{\text{liso}}(f_{>A,<B}) = \frac{1}{2n}\sum_{i=1}^{n}(Y_i - A)^2.$$

This is clearly minimised by $A_\lambda = B_\lambda = \overline{Y}$.

Corollary 1 follows from the fact that from properties of the PAVA,

$$\arg\max_m \left|\sum_{i=1}^{m}\left(Y_{\pi(i)} - \overline{Y}\right)\right| = \max\left\{m : \widehat{f}_{PAVA}(X_{\pi(m)}) < 0\right\}.$$

□

Proof of Theorem 3.3.3

Proof. We can apply a theorem proved in Tseng [2001]. However, the proof is simplified in the case where we go through the covariates in a random, independent order with each iteration, which we will show now.

Because we are doing repeated minimisations, for all m, $\ell_\lambda^{\text{liso}}(f^{m+1}) \leq \ell_\lambda^{\text{liso}}(f^m)$. Moreover, $\ell_\lambda^{\text{liso}}$ is bounded below by 0. Therefore, $\ell_\lambda^{\text{liso}}(f^m)$ must converge monotonically in probability as m increases.

Now, choose any $\delta_1 > 0$. We define A_m as the event that there exists $k \in \{1, \ldots, p\}$ and $g^m : \mathbb{R} \to \mathbb{R}$ such that

$$\ell_\lambda^{\text{liso}}(f_0, f_1^m, \ldots, f_p^m) - \ell_\lambda^{\text{liso}}(f_0, f_1^m, \ldots, f_{k-1}^m, g^m, f_{k+1}^m, \ldots, f_p^m) \geq \delta_1,$$

the event that we can improve on the current fit by at least δ_1 by changing only one component estimate.

Now, with the $m+1$th iteration, we have an equal probability of picking any one of p covariates as the first fitted of our new backfitting cycle, and hence

$$\text{Prob}\left(\ell_\lambda^{\text{liso}}(f^m) - \ell_\lambda^{\text{liso}}(f^{m+1}) \geq \delta_1\right) \geq \text{Prob}(A_m)/p.$$

But $\ell_\lambda^{\text{liso}}(f^m)$ converges in probability, so $\text{Prob}(A_m) \to 0$ for all $\delta_1 > 0$.

$\ell_\lambda^{\text{liso}}$ is continuously differentiable in the interior of $\oplus\mathcal{F}_k$. The set $f \in \oplus\mathcal{F}_k$ such that $\ell_\lambda^{\text{liso}}(f) \leq \ell_\lambda^{\text{liso}}(f^0)$ is closed, and compact. Therefore, the above implies that for any $\eta > 0$, $\delta_2 > 0$, there exists M_1 such that for all $m > M_1$, the subdifferential of $\ell_\lambda^{\text{liso}}$ at f^m contains a plane that is within δ_2 of zero with probability at least $1 - \eta$.

Therefore, by considering sufficiently small values of δ_2, this implies that for all $\eta > 0$, $\delta_3 > 0$, there exists M_2 such that for all $m > M_2$, there exists with probability at least $1 - \eta$, another sum of functions $\widetilde{f}^m \in \oplus\mathcal{F}_k$ satistifying $\|\widetilde{f}^m - f^m\| < \delta_3$, at which $\ell_\lambda^{\text{liso}}$ contains the zero plane in its subdifferential.

Since $\ell_\lambda^{\text{liso}}$ is convex, $\widetilde{f}^m$ is a global minimiser of $\ell_\lambda^{\text{liso}}$. Taking η, δ_3 to zero, we see by continuity of $\ell_\lambda^{\text{liso}}$ that $\ell_\lambda^{\text{liso}}(f^m)$ converges in probability to the global minimum.

In addition, this implies that if the global minimum is unique, and $\widetilde{f}^m$ is fixed, f^m must converge to it.

□

A.2 Proofs for Chapter 5

As an useful notation, for any vector v, we denote by v^+ and v^- the maximum and minimum value of v respectively.

Helpful in these proofs is a lemma from Bickel et al. [2009], which we provide here the version in van de Geer and Bühlmann [2009] without proof:

Lemma A.2.1. *(Lemma 2.2 of van de Geer and Bühlmann [2009]) Fix any $r, q > 1$ such that $1/r + 1/q = 1$, $s > 0$, and δ a vector. Let $\mathcal{N}$ be the s largest values of $|\delta|$. Then*

$$\|\delta_{\mathcal{N}^c}\|_r \leq \|\delta\|_1 / s^{1/q}$$

A.2.1 Proof of Lemma 5.1.2

Proof. Let δ be any vector satisfying

$$\frac{\|X\delta\|_n^2}{\|\delta_S\|^2} = \phi^2(L, S), \quad \|\delta_{S^c}\|_1 \leq L\sqrt{|S|}\,\|\delta_S\|\,.$$

Assume without loss of generality that $\|X\delta\|_n = 1$. Then by Definition 5.1.2, for each $H \in \mathcal{H}$, $\|\delta_H\|^2 \leq 1/\phi_G^2(L, S)$.

Hence

$$1/\phi^2(L, S) = \|\delta_S\|^2 \leq \sum_{H \in \mathcal{H}} \|\delta_H\|^2 \leq |\mathcal{H}|/\phi_G^2(L, S).$$

Similarly, let δ now instead satisfy

$$\min_{M \in \mathcal{G}} \frac{\|X\delta\|_n^2}{\|\delta_M\|^2} = \phi_{\mathcal{G}}^2(L, S), \quad \|\delta_{S^c}\|_1 \leq L\sqrt{|S|}\,\|\delta_S\|\,.$$

Assuming $\|X\delta\|_n = 1$ means by definition that

$$1/\phi_{\mathcal{G}}^2(L, S) = \max_{M \in \mathcal{G}} \|\delta_M\|^2 \leq \max_{M \in \mathcal{G}} \|\delta_{M \cup S}\|^2 \leq 1/\phi^2(L, S, |S| + \max|G|).$$

Finally, if δ satisfies $\|\delta_{S^c}\|_1 \leq L\sqrt{|S|}\,\|\delta_S\|$, and

$$\min_M \left\{ \frac{\|X\delta\|_n^2}{\|\delta_M\|^2} : S \subset M, |M| \leq |S| + \max|G| \right\} = \phi^2(L, S, |S| + \max|G|),$$

assuming $\|X\delta\|_n = 1$ means for some M, $|M \setminus S| = \max|G|$, it is the case that $\|\delta_M\|^2 = 1/\phi^2(L, S, |S| + \max|G|)$.

Then

$$\begin{aligned}\|\delta_S\|^2 &= \|\delta_M\|^2 - \left\|\delta_{M\setminus S}\right\|^2 \\ &\geq \|\delta_M\|^2 - \left\|\delta_{M\setminus S}\right\|_1^2 \\ &\geq \|\delta_M\|^2 - L^2|S|\,\|\delta_S\|^2 .\end{aligned}$$

So

$$\frac{1}{\phi^2(L,S)} \geq \|\delta_S\|^2 \geq \frac{\|\delta_M\|^2}{1 + L^2|S|} \geq \frac{1}{\phi^2(L, S, |S| + \max|G|)(1 + L^2|S|)}.$$

□

A.2.2 Proof of Theorem 5.2.1

For our theoretical analysis, let us assume there exists $\beta = \beta_S$ such that the data can be written as

$$Y = X\beta + \varepsilon$$

where S is a set of relevant covariates, and ε is a noise term. In our analysis, we assume that the model is "truly" sparse in the sense that the underlying model β has non-zeroes only in S. It is possible to encompass the more general case where β is only approximately sparse by using ε to incorporate approximation error arising from sparsification, if the removed covariates have a very small coefficient.

Our results will be in two halves - first, we show that under a broad condition, the weights w successfully separate between covariates from zero groups $\mathcal{G}^c_{\cap S}$ and covariates from non-zero groups $\mathcal{G}_{\cap S}$. Then, we show that with this separation, we achieve good results on the second optimisation.

A.2.2.1 Convergence of Group Weights

Existing work, including Bickel et al. [2009] and van de Geer and Bühlmann [2009] have proven bounds for the estimation error of the Lasso. We show as a variant bounds on the group-wise errors of the initial estimate, and so by implication, the weights used in the second estimate.

Lemma A.2.2. *Let*

$$\lambda \geq 2 \max |X'\varepsilon/n|.$$

Then for all $G \in \mathcal{G}$,

$$\left\| \widehat{\beta}_G^{(l)} - \beta_G \right\| / \sqrt{|G|} \leq \frac{2(\lambda + \max |X'\varepsilon/n|)\sqrt{|S|}}{\phi_{\mathcal{G}}(3, S)\phi(3, S)\sqrt{|G|}}.$$

Proof. Similarly to Lemma A.2.5, we have that

$$\begin{aligned}
\lambda \left\|\beta\right\|_1 - \lambda \left\|\widehat{\beta}^{(l)}\right\|_1 &\geq \frac{1}{2}\left(\left\|Y - X\widehat{\beta}^{(l)}\right\|_n^2 - \left\|Y - X\beta\right\|_n^2\right) \\
&\geq \frac{1}{2}\left\|X(\widehat{\beta}^{(l)} - \beta)\right\|_n^2 - \left(\varepsilon, X\left(\widehat{\beta}^{(l)} - \beta\right)\right)_n \\
&\geq \frac{1}{2}\left\|X(\widehat{\beta}^{(l)} - \beta)\right\|_n^2 - \max |X'\varepsilon/n| \left\|\widehat{\beta}_{S^c}^{(l)}\right\|_1 \\
&\quad - \max |X'\varepsilon/n| \left\|\widehat{\beta}_S^{(l)} - \beta_S\right\|_1.
\end{aligned}$$

Hence

$$\frac{1}{2}\left\|X(\widehat{\beta}^{(l)} - \beta)\right\|_n^2 + (\lambda - \max |X'\varepsilon/n|)\left\|\widehat{\beta}_{S^c}^{(l)}\right\|_1 \leq (\lambda + \max |X'\varepsilon/n|)\left\|\widehat{\beta}_S^{(l)} - \beta_S\right\|_1.$$

So by definition of $\phi(L, S)$ and $\phi_{\mathcal{G}}(L, S)$, we have by a similar argument to that in Lemma A.2.5 that

$$\left\| \widehat{\beta}_G^{(l)} - \beta_G \right\| \leq \frac{2(\lambda + \max |X'\varepsilon/n|)\sqrt{|S|}}{\phi(L, S)\phi_{\mathcal{G}}(L, S)}$$

with

$$L = \frac{\lambda + \max |X'\varepsilon/n|}{\lambda - \max |X'\varepsilon/n|} \leq \frac{3 \max |X'\varepsilon/n|}{\max |X'\varepsilon/n|}$$

As $\phi(L, S)$, $\phi_{\mathcal{G}}(L, S)$ decreases with L, the rest follows. □

From Lemma A.2.2, the contribution of the noise ε then is based on the maximal correlation $\max |X'\varepsilon/n|$. As other authors have identified [van de Geer and Bühlmann 2009], it is possible to bound this for normally distributed noise:

Lemma A.2.3. *Let $X \in \mathbb{R}^{n\times k}$, with $\|X_{\cdot,j}\|_n$ bounded above by some constant C for all j, and $\varepsilon \sim N(0,\sigma^2)$. Then with probability exceeding*

$$1 - \frac{\exp(-t/2)}{\sqrt{\pi}(t + 2\log(k))},$$

we have that

$$\max |X'\varepsilon/n| \leq C\sigma\sqrt{\frac{t + 2\log(k)}{n}}.$$

Proof. Similar proofs have appeared elsewhere. We present the proof here for convenience.

Now, $X'\varepsilon/\|X\|$ are distributed identically (but possibly not independently) Normal with variance σ^2. Therefore setting $\delta = \sqrt{\frac{t+2\log(k)}{n}}$, we have

$$\begin{aligned} Pr(\|X'\varepsilon\|_\infty /n \geq \delta C\sigma) &\leq |T| Pr(|X'\varepsilon|/(\|X\|\,\sigma) \geq \sqrt{n}\delta) \\ &\leq \frac{|T|}{\sqrt{n\pi}\delta}\exp(-n\delta^2/2) \\ &\leq \frac{\exp(-t/2)}{\sqrt{\pi(t + 2\log(k))}}. \end{aligned}$$

□

A similar lemma will work in the case of more general subgaussian noise, albeit with different constants in the bound.

Using the above, we have the following:

Lemma A.2.4. *Suppose that A1, A2 and B1 hold and $\mathcal{G}$ is non-overlapping. Choose*

$$\lambda = O\left(\max |X'\varepsilon/n|\right), \quad \text{with} \quad \lambda \geq 2\max |X'\varepsilon/n|$$

with $\lambda \geq 2\max|X'\varepsilon/n|$. Then, for all $\eta \geq 0$, there exists C' such that with probability exceeding $1-\eta$ we have for each $G \in \mathcal{G}$,

$$\left| w_G^{-1} - \frac{\|\beta_G\|}{\sqrt{|G|}} \right| \leq C'\left(\sigma\sqrt{\frac{|S|\log(p)}{n|G|}}\right) \tag{A.2.1}$$

where the inequalities are taken term-wise.

Proof. Result follows trivially from combining Lemma A.2.2 and Lemma A.2.3, noting that $w_j^{-1} = \left\|\widehat{\beta}_G^{(l)}\right\| / |G|$ for $j \in G$. □

A.2.2.2 Second stage convergence

To translate the bounds on the weights into bounds on the second stage, we require some theorems for the weighted Lasso. Now, several authors have proven a variety of results relating to this. In particular, van de Geer et al. [2010] proved some inequalities similar in spirit to ours. However, their focus was on convergence for the Adaptive Lasso, with the additional complication of model misspecification. The Co-adaptive Lasso provides a distinct challenge in that we expect faster weight convergence for the variables in the out of group set $\widetilde{S}^c$, and slow or non-existent weight convergence in the case of irrelevant variables within groups.

Lemma A.2.5. *Results for the second stage*

Fix any T, $\widetilde{S} \supseteq T \supseteq S$ such that $\widehat{\beta}^{OLS,T} \in \mathbb{R}^p$, the ordinary least squares estimate when computed on the restricted covariate set T, exists.

$$\widehat{\beta}_{T^c}^{OLS,T} = 0, \quad \widehat{\beta}_T^{OLS,T} = (X_T' X_T)^{-1} X_T' Y,$$

with residual $\varepsilon^{OLS} = Y - X\widehat{\beta}^{OLS,T}$.

Let $\mu^{\varepsilon_{OLS}} = \max |X'\varepsilon^{OLS}|/n$, and $\mu_{\widetilde{S}}^{\varepsilon_{OLS}} = \max |X'_{\widetilde{S}\setminus T}\varepsilon^{OLS}|/n$.

For $\mu \geq \max\left(\mu^{\varepsilon_{OLS}}/w^-_{\widetilde{S}^c}, \mu_{\widetilde{S}}^{\varepsilon_{OLS}}/w^-_{\widetilde{S}\setminus T}\right)$, setting

$$L_1 = \mu w_T^+ / \left(\mu w^-_{\widetilde{S}^c} - \mu^{\varepsilon_{OLS}}\right)$$
$$L_2 = \mu w_T^+ / \left(\mu w^-_{\widetilde{S}\setminus T} - \mu_{\widetilde{S}}^{\varepsilon_{OLS}}\right),$$

we have that

$$\left\|X(\widehat{\beta}^{OLS,T}-\widehat{\beta}^{(c)})\right\|_n \leq 2\mu w_T^+\sqrt{|T|}/\max\left(\phi(L_1,\widetilde{S}),\phi(\max(L_1,L_2),T)\right) \tag{A.2.2}$$

$$\left\|\widehat{\beta}_T^{OLS,T}-\widehat{\beta}_T^{(c)}\right\| \leq 2\mu w_T^+\sqrt{|T|}/\max\left(\phi^2(L_1,\widetilde{S}),\phi^2(\max(L_1,L_2),T)\right) \tag{A.2.3}$$

$$\left\|\widehat{\beta}^{OLS,T}-\widehat{\beta}^{(c)}\right\| \leq \frac{2\mu w_T^+\sqrt{|T|}\,(1+\max(L_1,L_2))}{\max\left(\phi^2(L_1,\widetilde{S},|\widetilde{S}|+|T|),\phi^2(\max(L_1,L_2),T,2|T|)\right)}. \tag{A.2.4}$$

Proof. Our proof is similar in spirit to those in van de Geer et al. [2010].

By definition of the weighted Lasso, we have that

$$\begin{aligned}
\mu\left\|w\widehat{\beta}^{OLS,T}\right\|_1-\mu\left\|w\widehat{\beta}^{(c)}\right\|_1 &\geq \frac{1}{2}\left(\left\|Y-X\widehat{\beta}^{(c)}\right\|_n^2-\left\|Y-X\widehat{\beta}^{OLS,T}\right\|_n^2\right)\\
&\geq \frac{1}{2}\left\|X(\widehat{\beta}^{(c)}-\widehat{\beta}^{OLS,T})\right\|_n^2-\left(\varepsilon^{OLS},X\left(\widehat{\beta}^{(c)}-\widehat{\beta}^{OLS,T}\right)\right)_n\\
&\geq \frac{1}{2}\left\|X(\widehat{\beta}^{(c)}-\widehat{\beta}^{OLS,T})\right\|_n^2\\
&\quad-\mu^{\varepsilon_{OLS}}\left\|\widehat{\beta}^{(c)}_{\widetilde{S}^c}\right\|_1-\mu^{\varepsilon_{OLS}}_{\widetilde{S}}\left\|\widehat{\beta}^{(c)}_{\widetilde{S}\setminus T}\right\|_1,
\end{aligned}$$

since $X_T'\varepsilon^{OLS}=0$.

Hence,

$$\begin{aligned}
\frac{1}{2}\left\|X(\widehat{\beta}^{OLS,T}-\widehat{\beta}^{(c)})\right\|_n^2 &\leq \mu\left(\left\|w_T\widehat{\beta}_T^{OLS,T}\right\|_1-\left\|w_T\widehat{\beta}_T^{(c)}\right\|_1\right) \qquad \text{(A.2.5)}\\
&\quad+\mu^{\varepsilon_{OLS}}\left\|\widehat{\beta}^{(c)}_{\widetilde{S}^c}\right\|_1-\mu\left\|w_{\widetilde{S}^c}\widehat{\beta}^{(c)}_{\widetilde{S}^c}\right\|_1\\
&\quad+\mu^{\varepsilon_{OLS}}_{\widetilde{S}}\left\|\widehat{\beta}^{(c)}_{\widetilde{S}\setminus T}\right\|_1-\mu\left\|w_{\widetilde{S}\setminus T}\widehat{\beta}^{(c)}_{\widetilde{S}\setminus T}\right\|_1\\
&\leq \mu\left\|w_T\right\|\left\|\widehat{\beta}_T^{OLS,T}-\widehat{\beta}_T^{(c)}\right\|+\left(\mu^{\varepsilon_{OLS}}-\mu w^-_{\widetilde{S}^c}\right)\left\|\widehat{\beta}^{(c)}_{\widetilde{S}^c}\right\|_1\\
&\quad+\left(\mu^{\varepsilon_{OLS}}_{\widetilde{S}}-\mu w^-_{\widetilde{S}\setminus T}\right)\left\|\widehat{\beta}^{(c)}_{\widetilde{S}\setminus T}\right\|_1\\
&\leq \mu w_T^+\sqrt{|T|}\left\|\widehat{\beta}_T^{OLS,T}-\widehat{\beta}_T^{(c)}\right\|+\left(\mu^{\varepsilon_{OLS}}-\mu w^-_{\widetilde{S}^c}\right)\left\|\widehat{\beta}^{(c)}_{\widetilde{S}^c}\right\|_1\\
&\quad+\left(\mu^{\varepsilon_{OLS}}_{\widetilde{S}}-\mu w^-_{\widetilde{S}\setminus T}\right)\left\|\widehat{\beta}^{(c)}_{\widetilde{S}\setminus T}\right\|_1 \qquad \text{(A.2.6)}
\end{aligned}$$

Let $\mu \geq \max\left(\mu^{\varepsilon_{OLS}}/w^-_{\widetilde{S}^c}, \mu^{\varepsilon_{OLS}}_{\widetilde{S}}/w^-_{\widetilde{S}\setminus T}\right)$. Setting

$$\begin{aligned} L_1 &= \mu w^+_T / \left(\mu w^-_{\widetilde{S}^c} - \mu^{\varepsilon_{OLS}}\right) \\ L_2 &= \mu w^+_T / \left(\mu w^-_{\widetilde{S}\setminus T} - \mu^{\varepsilon_{OLS}}_{\widetilde{S}}\right), \end{aligned}$$

we have then that

$$\begin{aligned} \sqrt{|T|}\left\|\widehat{\beta}^{OLS,T}_T - \widehat{\beta}^{(c)}_T\right\| &\geq \left\|\widehat{\beta}^{(c)}_{\widetilde{S}^c}\right\|_1 / L_1 + \left\|\widehat{\beta}^{(c)}_{\widetilde{S}\setminus T}\right\|_1 / L_2 \\ \max(L_1, L_2)\sqrt{|T|}\left\|\widehat{\beta}^{OLS,T}_T - \widehat{\beta}^{(c)}_T\right\| &\geq \left\|\widehat{\beta}^{(c)}_{T^c}\right\|_1, \end{aligned}$$

so by definition

$$\left\|X(\widehat{\beta}^{OLS,T} - \widehat{\beta}^{(c)})\right\|_n \geq \phi(\max(L_1, L_2), T)\left\|\widehat{\beta}^{OLS,T}_T - \widehat{\beta}^{(c)}_T\right\|.$$

Similarly, since $T \subset \widetilde{S}$,

$$\begin{aligned} \sqrt{|T|}\left\|\widehat{\beta}^{OLS,T}_T - \widehat{\beta}^{(c)}_T\right\| &\geq \left\|\widehat{\beta}^{(c)}_{\widetilde{S}^c}\right\|_1 / L_1 \\ \sqrt{|\widetilde{S}|}\left\|\widehat{\beta}^{OLS,T}_{\widetilde{S}} - \widehat{\beta}^{(c)}_{\widetilde{S}}\right\| &\geq \left\|\widehat{\beta}^{(c)}_{\widetilde{S}^c}\right\|_1 / L_1 \\ L_1\sqrt{|\widetilde{S}|}\left\|\widehat{\beta}^{OLS,T}_{\widetilde{S}} - \widehat{\beta}^{(c)}_{\widetilde{S}}\right\| &\geq \left\|\widehat{\beta}^{(c)}_{\widetilde{S}^c}\right\|_1, \end{aligned}$$

so

$$\left\|X(\widehat{\beta}^{OLS,T} - \widehat{\beta}^{(c)})\right\|_n \geq \phi(L_1, \widetilde{S})\left\|\widehat{\beta}^{OLS,T}_{\widetilde{S}} - \widehat{\beta}^{(c)}_{\widetilde{S}}\right\| \leq \phi(L_1, \widetilde{S})\left\|\widehat{\beta}^{OLS,T}_T - \widehat{\beta}^{(c)}_T\right\|.$$

Putting these together, we have that

$$\frac{\mu w_T^+ \sqrt{|T|}\left\|X(\widehat{\beta}^{OLS,T}-\widehat{\beta}^{(c)})\right\|_n}{\max\left(\phi(L_1,\widetilde{S}),\phi(\max(L_1,L_2),T)\right)} \geq \frac{1}{2}\left\|X(\widehat{\beta}^{OLS,T}-\widehat{\beta}^{(c)})\right\|_n^2 + \left(\mu w_{\widetilde{S}^c}^- - \mu^{\varepsilon_{OLS}}\right)\left\|\widehat{\beta}^{(c)}_{\widetilde{S}^c}\right\|_1 + \left(\mu w_{\widetilde{S}\setminus T}^- - \mu^{\varepsilon_{OLS}}_{\widetilde{S}}\right)\left\|\widehat{\beta}^{(c)}_{\widetilde{S}\setminus T}\right\|_1. \tag{A.2.7}$$

Hence, for our choice of μ,

$$\left\|X(\widehat{\beta}^{OLS,T}-\widehat{\beta}^{(c)})\right\|_n \leq 2\mu w_T^+\sqrt{|T|}/\max\left(\phi(L_1,\widetilde{S}),\phi(\max(L_1,L_2),T)\right) \tag{A.2.8}$$

$$\left\|\widehat{\beta}_T^{OLS,T}-\widehat{\beta}_T^{(c)}\right\| \leq 2\mu w_T^+\sqrt{|T|}/\max\left(\phi^2(L_1,\widetilde{S}),\phi^2(\max(L_1,L_2),T)\right) \tag{A.2.9}$$

$$\left\|\widehat{\beta}_{T^c}^{OLS,T}-\widehat{\beta}_{T^c}^{(c)}\right\|_1 \leq \max(L_1,L_2)2\mu w_T^+|T|/\max\left(\phi^2(L_1,\widetilde{S}),\phi^2(\max(L_1,L_2),T)\right) \tag{A.2.10}$$

$$\left\|\widehat{\beta}^{OLS,T}-\widehat{\beta}\right\| \leq 2\frac{\left(1+\max(L_1,L_2)\sqrt{|T|}\right)\mu w_T^+\sqrt{|T|}}{\max\left(\phi^2(L_1,\widetilde{S}),\phi^2(\max(L_1,L_2),T)\right)}. \tag{A.2.11}$$

The third inequality can be quite poor for large $|T|$ due to the $\max(L_1,L_2)\sqrt{|T|}$ term. On the other hand, suppose that

$$\phi(L_1,\widetilde{S},|\widetilde{S}|+|T|)>0 \quad \text{or} \quad \phi(\max(L_1,L_2),T,2|T|)>0.$$

Then for any $M\supset T$, $|M\setminus T|\leq|T|$, such that $\min|\widehat{\beta}^{(c)}_M|\geq\max|\widehat{\beta}^{(c)}_{M^c}|$, we have that

$$\left\|\widehat{\beta}_M^{OLS,T}-\widehat{\beta}_M^{(c)}\right\| \leq 2\mu w_T^+\sqrt{|T|}/\max\left(\phi^2(L_1,\widetilde{S},|\widetilde{S}|+|T|),\phi^2(\max(L_1,L_2),T,2|T|)\right).$$

From Lemma A.2.1, however, we have that

$$\left\|\widehat{\beta}^{(c)}_{M^c}\right\| \leq \left\|\widehat{\beta}^{(c)}_{T^c}\right\|_1/\sqrt{|T|},$$

so it follows that

$$\left\|\widehat{\beta}^{OLS,T} - \widehat{\beta}^{(c)}\right\| \leq 2\mu w_T^+ \sqrt{|T|} \frac{1 + \max(L_1, L_2)}{\max\left(\phi^2(L_1, \widetilde{S}, |\widetilde{S}| + |T|), \phi^2(\max(L_1, L_2), T, 2|T|)\right)}.$$

□

Remark A.2.1. Note that in all of these theorems, a somewhat better bound and conditions can be obtained by using the Cauchy-Strauss bound to replace w_T^+ with $\left\|w_T/\sqrt{T}\right\|$ and $w_{\widetilde{S}}^+$ with $\left\|w_{\widetilde{S}}/\sqrt{\widetilde{S}}\right\|$ in both the inequalities and the calculation of L_1 and L_2. This can improve things in the case where a small number of groups in the signal β are significantly smaller in terms of group-wise L2 norm than the rest, and the number of groups is large. However, in this paper, we use the former bound for simplicity, as the latter bound leads to conditions on sparsity-weighted harmonic means of group-wise L2 norms that are more difficult to interpret.

The effect of Lemma A.2.5 can be examined by varying T. Setting T to equal $\widetilde{S}$, we see that as the weights on the out of group variables increase relative to the ones on the relevant groups, we can obtain a fast convergence to the ordinary least squares estimate restricted to variables in the same group as the true ones, assuming that this exists. We do this by selecting a tuning parameter that scales in a manner inversely proportional to the weights on the out of group variables. In other words, by paying the price of ultimately just doing a least squares regression on $X_{\widetilde{S}}$, we remove the influence of the remaining variables very easily. This case is especially useful in the AIAO case, or if the group sizes are quite small.

Meanwhile, setting T to equal S implies that by choosing a higher tuning parameter than before, we can attempt to take advantage of the within group sparsity as well. At best, we can obtain a result similar to conducting a Lasso restricted to the relevant groups $\widetilde{S}$, a substantial improvement if $|\widetilde{S}| \ll p$. We pay a price in this case in terms of the ratio $w_S^+/w_{\widetilde{S}\setminus S}^-$, which can be quite significant if impact of the relevant groups varies greatly.

The conditions required for these convergences offer two possibilities. One is to bound $\phi(L_1, \widetilde{S})$ away from zero, which becomes increasingly easy as the weights ratio between relevant and irrelevant groups increase, since L_1 decreases. However, this requires that $\widetilde{S}$ be not too large, and especially be smaller than n, or else the loss of identifiability means the condition is automatically failed. The second condition of bounding $\phi(\max(L_1, L_2), T)$ allows reasonable success in the case of large $\widetilde{S}$, potentially larger than n, something distinct from the Group Lasso. Its fulfilment is more

complex – if the weights are the same within each group, $L_2 \geq 1$. Indeed, L_2 can be quite large if the contribution of relevant groups are quite variable, making this condition harder to satisfy and so Lemma A.2.5 harder to apply. We can compensate however by increasing T to incorporate elements of $\widetilde{S} \setminus S$ that are likely to have small values of w. In other words, we can still have good results, if we are willing to accept that we will likely incorrectly select some irrelevant covariates in groups that appear collectively very relevant from the initial calculation.

Our main result then emerges by combining this theorem with the convergence results from subsection A.2.2.1.

Proof of Theorem 5.2.1. Fix a $T \subseteq \widetilde{S}$ for which the conditions are satisfied, and choose any $t > 0$ so that $3\exp(-t/2)/\sqrt{\pi} \leq \eta$. Assume for now that $T \neq \widetilde{S}$. Writing $P = I - X_T(X_T'X_T)^{-1}X_T'$,

$$\begin{aligned}
\mu^{\varepsilon_{OLS}} &= \max |X'P\varepsilon|/n = \max |(PX)'\varepsilon|/n, \\
\mu_{\widetilde{S}}^{\varepsilon_{OLS}} &= \max |(PX_{\widetilde{S}})'\varepsilon|/n, \\
\lambda^{\varepsilon} &= \max |X'\varepsilon|/n.
\end{aligned}$$

But P is a projection matrix, so for all $X_{\cdot,j}$, $\|PX_{\cdot,j}\| \leq \|X_{\cdot,j}\|$. Therefore by Lemma A.2.3, under Assumption B1, each of the following fails to occur with probability less than $\exp(-t/2)/\sqrt{\pi}$:

$$\begin{aligned}
\mu^{\varepsilon_{OLS}} &\leq \sigma\sqrt{\frac{t + 2\log(p)}{n}} = O\left(\sigma\sqrt{\frac{\log(p)}{n}}\right) \\
\mu_{\widetilde{S}}^{\varepsilon_{OLS}} &\leq \sigma\sqrt{\frac{t + 2\log|\widetilde{S}|}{n}} = O\left(\sigma\sqrt{\frac{\log|\widetilde{S}|}{n}}\right)
\end{aligned}$$

Further, by Lemma A.2.4 taking η there to be less than $\exp(-t/2)/\sqrt{\pi}$, Conditions A1-2 and B1 implies then that choosing $\lambda = 2\lambda^{\varepsilon}$, there exists some $C' > 0$, such that

with probability exceeding $1 - \exp(-t/2)/\sqrt{\pi}$,

$$
\begin{aligned}
w_T^+ &\leq \max_{G \in \mathcal{G}_{\cap S}} \left(\|\beta_G\| / \sqrt{|G|} - C'\sigma \sqrt{\frac{|S| \log(p)}{n|G|}} \right)^{-1} \\
w_{\widetilde{S} \setminus T}^+ &\geq \min_{G \in \mathcal{G}_{\cap (\widetilde{S} \setminus T)}} \left(\|\beta_G\| / \sqrt{|G|} + C'\sigma \sqrt{\frac{|S| \log(p)}{n|G|}} \right)^{-1} \\
w_{\widetilde{S}^c}^- &\geq \min_{G \in \mathcal{G}^c_{\cap S}} \left(C'\sigma \sqrt{\frac{|S| \log(p)}{n|G|}} \right)^{-1}.
\end{aligned}
$$

With Condition B2, for all $C'' > 0$, there exists n_0 so that for all $n > n_0$, with the same probability,

$$
\begin{aligned}
w_T^+ &\leq \max_{G \in \mathcal{G}_{\cap S}} \frac{\sqrt{|G|}}{\|\beta_G\| - C'' \|\beta_G\| \sqrt{|G|^{-\gamma_1} n^{-\gamma_2}}} \\
w_{\widetilde{S} \setminus T}^+ &\geq \min_{G \in \mathcal{G}_{\cap (\widetilde{S} \setminus T)}} \frac{\sqrt{|G|}}{\|\beta_G\| + C'' \|\beta_G\| \sqrt{|G|^{-\gamma_1} n^{-\gamma_2}}} \\
w_{\widetilde{S}^c}^- &\geq \frac{\min_{G \in \mathcal{G}^c_{\cap S}} \sqrt{|G|^{1+\gamma_1} n^{\gamma_2}}}{C'' \min_{H \in \mathcal{G}_{\cap S}} \|\beta_H\|}.
\end{aligned}
$$

Assume then for the remainder of this proof that this is the case, together with the previous bounds on $\mu^{\varepsilon_{OLS}}$ and $\mu_{\widetilde{S}}^{\varepsilon_{OLS}}$. We see that this is true with probability greater than $1 - 3\exp(-t/2)/\sqrt{\pi} \geq 1 - \eta$.

Choosing

$$
\mu = 2 \max(\mu^{\varepsilon_{OLS}} / w_{\widetilde{S}^c}^-, \mu_{\widetilde{S}}^{\varepsilon_{OLS}} / w_{\widetilde{S} \setminus T}^-),
$$

gives, by the definitions in Lemma A.2.5, that it is simultaneously the case that for all $n \geq n_0$,

$$
\begin{aligned}
L_1 &= \mu w_T^+ / \left(\mu w_{\widetilde{S}^c}^- - \mu^{\varepsilon_{OLS}} \right) \\
&\leq 2\mu w_T^+ / \mu w_{\widetilde{S}^c}^- = 2 w_T^+ / w_{\widetilde{S}^c}^- \\
&\leq \max_{G \in \mathcal{G}_{\cap S}, H \in \mathcal{G}_{\cap S}^c} \frac{2\sqrt{|G|}}{1 - C'' \sqrt{|G|^{-\gamma_1} n^{-\gamma_2}}} \frac{C''}{\sqrt{|H|^{1+\gamma_1} n^{\gamma_2}}} \\
L_2 &= \mu w_T^+ / \left(\mu w_{\widetilde{S} \setminus T}^- - \mu_{\widetilde{S}}^{\varepsilon_{OLS}} \right) \\
&\leq 2\mu w_T^+ / \mu w_{\widetilde{S} \setminus T}^- = 2 w_T^+ / w_{\widetilde{S} \setminus T}^- \\
&\leq \max_{G \in \mathcal{G}_{\cap S}, H \in \mathcal{G}_{\cap (\widetilde{S} \setminus T)}} 2 \frac{\|\beta_H\| \sqrt{|G|}}{\|\beta_G\| \sqrt{|H|}} \frac{1 + C'' \sqrt{|H|^{-\gamma_1} n^{-\gamma_2}}}{1 - C'' \sqrt{|G|^{-\gamma_1} n^{-\gamma_2}}}.
\end{aligned}
$$

Hence, for any value of $\delta > 0$, as $|G|, n \to \infty$, eventually

$$
L_1 \leq \max_{G \in \mathcal{G}_{\cap S}, H \in \mathcal{G}_{\cap S}^c} \delta \sqrt{\frac{|G|}{|H|^{1+\gamma_1} n^{\gamma_2}}}.
$$

Similarly, for all $\delta > 0$, with $|G|, |H|, n$ sufficiently large, it is the case that

$$
L_2 \leq \max_{G \in \mathcal{G}_{\cap S}, H \in \mathcal{G}_{\cap (\widetilde{S} \setminus T)}} 2(1+\delta) \frac{\|\beta_H\| \sqrt{|G|}}{\|\beta_G\| \sqrt{|H|}}.
$$

Therefore under condition C1(a) or condition C1(b), we have that there exists C such that either $\phi(L_1, \widetilde{S}) > C$ or $\phi(\max(L_1, L_2), T) > C$. Hence by Lemma A.2.5,

$$\begin{aligned}
\left\|X\left(\widehat{\beta}^{OLS,T}-\widehat{\beta}^{(c)}\right)\right\|_n &\leq 2\mu w_T^+\sqrt{|T|}/C \\
&\leq 2\sigma\sqrt{\frac{|T|}{n}}\max\left(2\sqrt{\log(p)}\frac{w_T^+}{w_{\widetilde{S}^c}^-},2\sqrt{\log|\widetilde{S}|}\frac{w_T^+}{w_{\widetilde{S}\setminus T}^-}\right)/C \\
&= O\left(\sigma\sqrt{\frac{|T|}{n}}\max\left(\sqrt{\log(p)}\max_{G\in\mathcal{G}_{\cap S},H\in\mathcal{G}_{\cap S}^c}\frac{\sqrt{|G|}}{\sqrt{|H|^{1+\gamma_1}n^{\gamma_2}}},\right.\right. \\
&\qquad \left.\left.\sqrt{\log|\widetilde{S}|}\max_{G\in\mathcal{G}_{\cap S},H\in\mathcal{G}_{\cap(\widetilde{S}\setminus T)}}\frac{\|\beta_H\|\sqrt{|G|}}{\|\beta_G\|\sqrt{|H|}}\right)\right) \\
\left\|\widehat{\beta}_T^{OLS,T}-\widehat{\beta}_T^{(c)}\right\| &= O\left(\sigma\sqrt{\frac{|T|}{n}}\max\left(\sqrt{\log(p)}\max_{G\in\mathcal{G}_{\cap S},H\in\mathcal{G}_{\cap S}^c}\frac{\sqrt{|G|}}{\sqrt{|H|^{1+\gamma_1}n^{\gamma_2}}},\right.\right. \\
&\qquad \left.\left.\sqrt{\log|\widetilde{S}|}\max_{G\in\mathcal{G}_{\cap S},H\in\mathcal{G}_{\cap(\widetilde{S}\setminus T)}}\frac{\|\beta_H\|\sqrt{|G|}}{\|\beta_G\|\sqrt{|H|}}\right)\right).
\end{aligned}$$

Similarly, under condition C2(a) or C2(b), we have that there exists C such that either

$$\phi(L_1,\widetilde{S},|\widetilde{S}|+|T|)>C \quad \text{or} \quad \phi(\max(L_1,L_2),T,2|T|)>C.$$

Then, again by Lemma A.2.5,

$$\begin{aligned}
\left\|\widehat{\beta}_T^{OLS,T}-\widehat{\beta}_T^{(c)}\right\| = O\left(\sigma\sqrt{\frac{|T|}{n}}\max\left(\sqrt{\log(p)}\max_{G\in\mathcal{G}_{\cap S},H\in\mathcal{G}_{\cap S}^c}\frac{\sqrt{|G|}}{\sqrt{|H|^{1+\gamma_1}n^{\gamma_2}}},\right.\right. \\
\left.\left.\sqrt{\log|\widetilde{S}|}\max_{G\in\mathcal{G}_{\cap S},H\in\mathcal{G}_{\cap(\widetilde{S}\setminus T)}}\frac{\|\beta_H\|\sqrt{|G|}}{\|\beta_G\|\sqrt{|H|}}\right)(1+\max(L_1,L_2))\right).
\end{aligned}$$

Now, under condition C1, the smallest singular value of $X_{\cdot,T}/\sqrt{n}$ remains bounded away from zero. Therefore

$$\left\|X(\widehat{\beta}^{OLS,T}-\widehat{\beta})\right\|_n = O(\sigma\sqrt{|T|/n}),$$

and

$$\left\|\widehat{\beta}^{OLS,T}-\widehat{\beta}\right\| = O(\sigma\sqrt{|T|/n}),$$

so using the triangle rule gives the result.

If $T = \widetilde{S}$, we can proceed with the proof as normal, simply omitting the terms relating to $\widetilde{S} \setminus T$ and condition C1(b) or C2(b).

□

A.2.3 Proof of Corollary 5.2.2 and Corollary 5.2.3

Proof. Simply apply Theorem 5.2.1, observing that given the condition on $|G|, |H|$, $L_1 \to 0$. In the case of Corollary 5.2.3 condition C1(a) is thus implied by A1. We have then that in the second lemma,

$$\left\| X\left(\beta - \widehat{\beta}^{(c)}\right)\right\|_n^2 = O\left(\sigma^2 \frac{|\widetilde{S}|}{n}(1 + \sqrt{\log(p)|G|^{-\gamma_1} n^{-\gamma_2}})^2\right).$$

But with non-overlapping groups with proportional sizes, $p = O(|\mathcal{G}||G|)$, so

$$\begin{aligned}\log(p)|G|^{-\gamma_1} &= O(\log \mathcal{G}|G|^{-\gamma_1} + \log|G||G|^{-\gamma_1}) \\ &= O(\log \mathcal{G}|G|^{-\gamma_1} + 1).\end{aligned}$$

Corollary 5.2.2 follows similarly. □

A.2.4 Proof of Lemma 5.4.1

Proof. As in Lemma A.2.5,

$$\left\|\widehat{\beta}^{(l)}_{\widetilde{S}^c}\right\|_1 \leq 3\sqrt{|S|}\left\|\widehat{\beta}^{(l)}_S - \beta_S\right\| \leq \frac{9\lambda|S|}{\phi^2(3,S)}.$$

Now, suppose that $\delta = \widehat{\beta}^{(l)}_{\widetilde{S}^c}$ is reordered in order of decreasing absolute value. Fix a $U \geq 1$, and consider the blocks $\delta_{1\ldots U}, \delta_{U+1\ldots 2U}, \ldots$.

Assume the conditions of the lemma hold.

Define A as the event that for each group $G \in \mathcal{G}$, the group contains less than α_G elements of the maximal absolute value block $\delta_{1\ldots U}$.

Define B as the event that for each group, the group contains less than α'_G variables from each individual block of size U, provided the block is not entirely zero.

Under event $A \cup B$, we have that for $G \in \mathcal{G}^c_{\cap S}$,

$$
\begin{aligned}
\widetilde{w}_G &\geq \left(\sum_{k=2}^{\lceil p/U \rceil} \sqrt{\sum_{i=1}^{U} I_{\{(k-1)U+i \in G\}} |\delta|_{(k-1)U+1} + \sqrt{\alpha_G} |\delta|_{U+1}} \right)^{-1} \\
&\geq \left(\sum_{k=2}^{\lceil p/U \rceil} \sqrt{\alpha'_G} \left\| \delta_{(k-2)U+1 \ldots U} \right\|_1 / U + \sqrt{\alpha_G} \left\| \delta \right\| / U \right)^{-1} \\
&\geq \left(\left(\sqrt{\alpha'_G} + \sqrt{\alpha_G} \right) \left\| \widehat{\beta}^{(l)}_{\widetilde{S}^c} \right\|_1 / U \right)^{-1} \\
&\geq \left(\frac{9\lambda \left(\sqrt{\alpha'_G} + \sqrt{\alpha_G} \right)}{\phi^2(3, S)} \frac{|S|}{U} \right)^{-1}
\end{aligned}
$$

Under the independence assumption, the probability of each event can be bounded below:

$$
\begin{aligned}
Pr(A^c) &\leq \sum_{G \in \mathcal{G}^c_{\cap S}} \binom{|G|}{\alpha_G} \binom{p - |\widetilde{S}| - \alpha_G}{U - \alpha_G} \Big/ \binom{p - |\widetilde{S}|}{U} \\
&\leq \sum_{G \in \mathcal{G}} \frac{|G|^{\alpha_G}}{\alpha_G!} \frac{U!(p - |\widetilde{S}| - \alpha_G)!}{(U - \alpha_G)!(p - |\widetilde{S}|)!} \\
&\leq \sum_{G \in \mathcal{G}} \frac{|G|^{\alpha_G}}{\alpha_G!} \frac{U^{\alpha_G}}{(p - |\widetilde{S}| - \alpha_G)^{\alpha_G}} \\
&\leq \sum_{G \in \mathcal{G}} \frac{1}{\alpha_G^{\alpha_G}} \left(\frac{e|G|U}{(p - |\widetilde{S}| - \alpha_G)} \right)^{\alpha_G} \qquad \leq \sum_{G \in \mathcal{G}} \frac{1}{\alpha_G^{\alpha_G}} \left(\frac{e|G|U}{(p - |\widetilde{S}| - |G|)} \right)^{\alpha_G}
\end{aligned}
$$

Write $C = \frac{e|G|U}{(p - |\widetilde{S}| - |G|)}$. Solving for $Pr(A^c) \leq e^{-t}$, we have that

$$
\begin{aligned}
e^{-t}/|\mathcal{G}| &\geq \left(\frac{C}{\alpha_G} \right)^{\alpha_G} \\
(\alpha_G / C)^{\alpha_G / C} &\geq \left(e^t |\mathcal{G}| \right)^{1/C} \\
\alpha_G &\geq \frac{t + \log(|\mathcal{G}|)}{W((t + \log(|\mathcal{G}|))/C)},
\end{aligned}
$$

with W the Lambert W function.

For C sufficiently small, this can be bounded as

$$\alpha_G \geq \frac{t + \log(|\mathcal{G}|)}{\log((t + \log(|\mathcal{G}|))/C) - \log\log((t + \log(|\mathcal{G}|))/C)}$$

,

and for this value of α_G, $Pr(A) > 1 - e^{-t}$.

For event B, we observe that because the Lasso selects a maximum of $\min(p, n)$ covariates, there a maximum of $\lceil \min(p, n)/U \rceil$ non-zero blocks. By a similar argument, choosing

$$\begin{aligned}
\alpha'_G &\geq (t + \log(|\mathcal{G}|) + \log(\lceil \min(p, n)/U \rceil)) \left(\log \left(\frac{t + \log(|\mathcal{G}|) + \log(\lceil \min(p, n)/U \rceil)}{C} \right) \right. \\
&\quad \left. - \log\log \left(\frac{t + \log(|\mathcal{G}|) + \log(\lceil \min(p, n)/U \rceil)}{C} \right) \right)^{-1} \\
&\geq \frac{t + \log(|\mathcal{G}|) + \log(\lceil \min(p, n)/U \rceil)}{W\left((t + \log(|\mathcal{G}|) + \log(\lceil \min(p, n)/U \rceil)) / C \right)},
\end{aligned}$$

implies that event B occurs with probability exceeding $1 - e^{-t}$.

□

A.2.5 Proof of Lemma 5.4.2

Proof. Note that

$$e^{-n} n^n \leq n! \leq n^n,$$

so

$$\binom{|G|}{\lceil |G|/2 \rceil} \leq (2e)^{|G|}.$$

We can proceed by a simple combinatorial argument.

$$\begin{aligned}
Pr(C) &\leq \sum_{G\in\mathcal{G}^c_{\cap S}} \binom{|G|}{\lceil |G|/2\rceil}\binom{p-|S|-\lceil |G|/2\rceil}{n-\lceil |G|/2\rceil}/\binom{p-|S|}{n} \\
&\leq \sum_{G\in\mathcal{G}^c_{\cap S}} (2e)^{|G|}\frac{n!}{(n-\lceil |G|/2\rceil)!}\frac{(p-|S|-\lceil |G|/2\rceil)!}{(p-|S|)!} \\
&\leq \sum_{G\in\mathcal{G}^c_{\cap S}} (2e)^{|G|}\frac{\sqrt{n}^{|G|}}{\sqrt{p-|S|-\lceil |G|/2\rceil}^{|G|}} \\
&\leq \sum_{G\in\mathcal{G}^c_{\cap S}} (2e)^{|G|}\frac{\sqrt{n}^{|G|}}{\sqrt{p-2n\rceil}^{|G|}}.
\end{aligned}$$

□

A.3 Proofs for Chapter 6

A.3.1 Proof of Theorem 6.2.1

Proof. Let $m > m_0$. Choose $\lambda^{(m)} = 2\max\left(\mu^{\varepsilon_{OLS}}/w^{(m)-}_{\widetilde{S}^c}, \mu^{\varepsilon_{OLS}}_{\widetilde{S}}/w^{(m)-}_{\widetilde{S}\setminus T}\right)$, and fix

$$\begin{aligned}
L_1 &= \mu w^{(m)+}_T / \left(\mu w^{(m)-}_{\widetilde{S}^c} - \mu^{\varepsilon_{OLS}}\right) \\
&= 2\mu w^{(m)+}_T / \mu w^{(m)-}_{\widetilde{S}^c} = 2w^{(m)+}_T / w^{(m)-}_{\widetilde{S}^c} \\
&\leq 2(1+\delta)\widetilde{w}^+_{\widetilde{S}} \max_{G\in\mathcal{G}^c_{\cap S}} \left(\frac{\sqrt{|G|}}{\left\|\widehat{\beta}^{(m-1)}_G\right\|}\right) \\
&\leq 2(1+\delta)\widetilde{w}^+_{\widetilde{S}}\delta \\
L_2 &= \mu w^{(m)+}_T / \left(\mu w^{(m)-}_{\widetilde{S}\setminus T} - \mu^{\varepsilon_{OLS}}_{\widetilde{S}}\right) \\
&\leq 2\mu w^{(m)+}_T / \mu w^{(m)-}_{\widetilde{S}\setminus T} = 2w^{(m)+}_T / w^{(m)-}_{\widetilde{S}\setminus T} \\
&\leq 2\left(\frac{1+\delta}{1-\delta}\right)\frac{\widetilde{w}^+_{\widetilde{S}}}{\widetilde{w}^-_{\widetilde{S}}}.
\end{aligned}$$

By Lemma A.2.5,

$$\begin{aligned}\left\|X(\widehat{\beta}^{OLS,T}-\widehat{\beta}^{(m)})\right\|_n &\leq 2\mu w_T^{(m)+}\sqrt{|T|}/\max\left(\phi(L_1,\widetilde{S}),\phi(\max(L_1,L_2),T)\right)\\ &\leq \frac{4\max\left(\mu^{\varepsilon_{OLS}}/w_{\widetilde{S}^c}^{(m)-},\mu_{\widetilde{S}}^{\varepsilon_{OLS}}/w_{\widetilde{S}\setminus T}^{(m)-}\right)w_T^{(m)+}\sqrt{|T|}}{\max\left(\phi(L_1,\widetilde{S}),\phi(\max(L_1,L_2),T)\right)}\\ &\leq \frac{4\max\left(\mu^{\varepsilon_{OLS}}\max_{G\in\mathcal{G}^c_{\cap S}}\left(\frac{\left\|\widehat{\beta}_G^{(m-1)}\right\|}{\sqrt{|G|}}\right),\frac{\mu_{\widetilde{S}}^{\varepsilon_{OLS}}}{w_{\widetilde{S}\setminus T}^{(m)-}}\right)w_T^{(m)+}\sqrt{|T|}}{\max\left(\phi(L_1,\widetilde{S}),\phi(\max(L_1,L_2),T)\right)}.\end{aligned}$$

Since we are computing a weighted Lasso with each stage of our algorithm, an identical argument to that in Lemma A.2.5 shows that by the definition of ϕ_G,

$$\max_{G\in\mathcal{G}^c_{\cap\widetilde{S}}}\left\|\widehat{\beta}_G^{(m)}\right\|\leq\max_{G\in\mathcal{G}}\left\|\widehat{\beta}_G^{OLS,T}-\widehat{\beta}_G^{(m)}\right\|\leq\frac{\left\|X(\widehat{\beta}^{OLS,T}-\widehat{\beta}^{(m)})\right\|_n}{\max\left(\phi_{\mathcal{G}}(L_1,\widetilde{S}),\phi_{\mathcal{G}}(\max(L_1,L_2),T)\right)}.$$

In the case that $\mu^{\varepsilon_{OLS}}/w_{\widetilde{S}^c}^{(m)-}\geq\mu_{\widetilde{S}}^{\varepsilon_{OLS}}/w_{\widetilde{S}\setminus T}^{(m)-}$, then

$$\max_{G\in\mathcal{G}^c_{\cap\widetilde{S}}}\left(\left\|\widehat{\beta}_G^{(m)}\right\|/\sqrt{|G|}\right)\leq\rho^{(m)}\max_{G\in\mathcal{G}^c_{\cap\widetilde{S}}}\left(\left\|\widehat{\beta}_G^{(m-1)}\right\|/\sqrt{|G|}\right),$$

where

$$\begin{aligned}\rho^{(m)}&\leq\frac{\max_{H\in\mathcal{G}^c_{\cap\widetilde{S}}}4\mu^{\varepsilon_{OLS}}\sqrt{|T|}w_T^{(m)+}\sqrt{|H|^{-1}}}{\max\left(\phi(L_1,\widetilde{S}),\phi(\max(L_1,L_2),T)\right)\max\left(\phi_{\mathcal{G}}(L_1,\widetilde{S}),\phi_{\mathcal{G}}(\max(L_1,L_2),T)\right)}\\ &\leq\frac{\max_{H\in\mathcal{G}^c_{\cap\widetilde{S}}}4\mu^{\varepsilon_{OLS}}\sqrt{|T|}(1+\delta)\widetilde{w}_{\widetilde{S}}^{+}\sqrt{|H|^{-1}}}{\max\left(\phi(L_1,\widetilde{S}),\phi(\max(L_1,L_2),T)\right)\max\left(\phi_{\mathcal{G}}(L_1,\widetilde{S}),\phi_{\mathcal{G}}(\max(L_1,L_2),T)\right)}.\end{aligned}$$

We define this final expression as ρ.

Note that under the conditions of the theorem, with high probability,

$$\rho=O\left(\sigma\sqrt{\log(p)/n}\sqrt{|T|}\widetilde{w}_{\widetilde{S}}^{+}\frac{1}{\min_{H\in\mathcal{G}^c_{\cap\widetilde{S}}}\sqrt{|H|}}\right).$$

Under B2, this becomes:

$$\rho = o\left(n^{-\gamma_2/2}\sqrt{\frac{|T|}{|S|}}\max_{\substack{G\in\mathcal{G}_{\cap S},\\ H\in\mathcal{G}^c_{\cap S}}}\sqrt{\frac{|G|^{1-\gamma_1}}{|H|}}\right).$$

Now, while $\mu^{\varepsilon_{OLS}}/w^{(m)-}_{\widetilde{S}^c} \geq \mu^{\varepsilon_{OLS}}_{\widetilde{S}}/w^{(m)-}_{\widetilde{S}\setminus T}$,

$$\begin{aligned}
\max_{G\in\mathcal{G}^c_{\cap\widetilde{S}}}\left(\left\|\widehat{\beta}^{(m)}_G\right\|/\sqrt{|G|}\right) &\leq \rho^{m-m_0}\max_{G\in\mathcal{G}^c_{\cap\widetilde{S}}}\left(\left\|\widehat{\beta}^{(m_0)}_G\right\|/\sqrt{|G|}\right)\\
1/w^{(m)-}_{\widetilde{S}^c} &\leq \rho^{m-m_0}/w^{(m_0)-}_{\widetilde{S}^c}\\
\left\|X(\widehat{\beta}^{OLS,T}-\widehat{\beta}^{(m)})\right\|_n &\leq \frac{4(1+\delta)\widetilde{w}^+_T\sqrt{|T|}}{\max\left(\phi(L_1,\widetilde{S}),\phi(\max(L_1,L_2),T)\right)}\\
&\quad\cdot\mu^{\varepsilon_{OLS}}\rho^{m-m_0}\max_{G\in\mathcal{G}^c_{\cap S}}\left(\frac{\left\|\widehat{\beta}^{(m_0)}_G\right\|}{\sqrt{|G|}}\right).
\end{aligned}$$

If $\rho < 1$, then the above means that for sufficiently large m, it must be the case that $\mu^{\varepsilon_{OLS}}/w^{(m)-}_{\widetilde{S}^c} \leq \mu^{\varepsilon_{OLS}}_{\widetilde{S}}/w^{(m)-}_{\widetilde{S}\setminus T}$. Hence for sufficiently large m,

$$\left\|X(\widehat{\beta}^{OLS,T}-\widehat{\beta}^{(m)})\right\|_n \leq \frac{4\frac{\mu^{\varepsilon_{OLS}}_{\widetilde{S}}}{(1-\delta)\widetilde{w}_{\widetilde{S}}}(1+\delta)\widetilde{w}^+_{\widetilde{S}}\sqrt{|T|}}{\max\left(\phi(L_1,\widetilde{S}),\phi(\max(L_1,L_2),T)\right)},$$

and the remaining result follows from the conditions.

□

A.3.2 Proof of Theorem 6.3.1

Proof. We fulfil the conditions of Theorem 5 of Xiao [2010]. Therefore we have the following convergence in probability as $t\to\infty$:

$$\frac{1}{2}\left\|Y-X\overline{\beta}^{(1)}_t\right\|^2_n + \lambda|\overline{\beta}^{(1)}_t| \xrightarrow{p} \min_\beta \frac{1}{2}E\left\|Y-X\beta\right\|^2 + \lambda|\beta|,$$

where the expectation is over X and Y.

Because the Lasso criterion is continuous and convex, this implies that, writing $\widehat{\beta}^{(l)}_\infty$ as the population version of the Lasso,

$$\overline{\beta}^{(1)}_t \xrightarrow{p} \widehat{\beta}^{(l)}_\infty.$$

If the inverse weights are themselves continuous functions of the initial estimate, then this means that the inverse weights must converge to the inverse weights of the population version.

Now, the same Theorem 5 of Xiao [2010], together with the convexity of the individual Lasso criterion implies that the second stage would also show the following convergence in probability:

$$\overline{\beta}_t^{(2)} \xrightarrow{p} \arg\min_{\beta} \frac{1}{2} E \sum_{t=1}^{\infty} \left\| Y - \frac{X_{t,\cdot}}{\tilde{w}_t} \beta \right\|^2 + \lambda |\beta|,$$

where $\tilde{w}_t$ is the weight resulting from the first computation after t samples, and the division is taken element wise. But because this converges, we have that

$$\overline{\beta}_t^{(2)} / \tilde{w}_t \xrightarrow{p} \left(\arg\min_{\beta} \frac{1}{2} E \left\| Y - \frac{X_{t,\cdot}}{w} \beta \right\|^2 + \lambda |\beta| \right) / w.$$

Result then follows by consideration of Equation (6.1.1).

□