Modelling interactions in high-dimensional data with Backtracking.

Rajen D. Shah
Statistical Laboratory, University of Cambridge
r.shah@statslab.cam.ac.uk

February 5, 2016

Abstract

We study the problem of high-dimensional regression when there may be interacting variables. Approaches using sparsity-inducing penalty functions such as the Lasso [Tibshirani, 1996] can be useful for producing interpretable models. However, when the number variables runs into the thousands, and so even two-way interactions number in the millions, these methods become computationally infeasible. Typically variable screening based on model fits using only main effects must be performed first. One problem with screening is that important variables may be missed if they are only useful for prediction when certain interaction terms are also present in the model.

To tackle this issue, we introduce a new method we call Backtracking. It can be incorporated into many existing high-dimensional methods based on penalty functions, and works by building increasing sets of candidate interactions iteratively. Models fitted on the main effects and interactions selected early on in this process guide the selection of future interactions. By also making use of previous fits for computation, as well as performing calculations is parallel, the overall run-time of the algorithm can be greatly reduced.

The effectiveness of our method when applied to regression and classification problems is demonstrated on simulated and real data sets. In the case of using Backtracking with the Lasso, we also give some theoretical support for our procedure.

Keywords: high-dimensional data, interactions, Lasso, path algorithm

1 Introduction

In recent years, there has been a lot of progress in the field of high-dimensional regression. Much of the development has centred around the Lasso [Tibshirani, 1996], which given a vector of responses $\mathbf{Y} \in \mathbb{R}^n$ and design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, solves

$$(\hat{\mu}, \hat{\boldsymbol{\beta}}) := \underset{(\mu, \boldsymbol{\beta}) \in \mathbb{R} \times \mathbb{R}^p}{\arg\min} \{ \tfrac{1}{2n} \|\mathbf{Y} - \mu \mathbf{1} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \}, \tag{1.1}$$

where $\mathbf{1}$ is an n-vector of ones and the regularisation parameter λ controls the relative contribution of the penalty term to the objective. The many extensions of the Lasso allow most familiar models from classical (low-dimensional) statistics to now be fitted in situations where the number of variables p may be tens of thousands and even greatly exceed the number of observations n (see the monograph Bühlmann and van de Geer [2011a] and references therein).

However, despite the advances, fitting models with interactions remains a challenge. Two issues that arise are:

(i) Since there are $p(p-1)/2$ possible first-order interactions, the main effects can be swamped by the vastly more numerous interaction terms and without proper regularisation, stand little chance of being selected in the final model (see Figure 1b).

(ii) Monitoring the coefficients of all the interaction terms quickly becomes infeasible as p runs into the thousands.

1.1 Related work

For situations where $p < 1000$ or thereabouts and the case of two-way interactions, a lot of work has been done in recent years to address this need. To tackle (i), many of the proposals use penalty functions and constraints designed to enforce that if an interaction term is in the fitted model, one or both main effects are also present [Lin and Zhang, 2006, Zhao et al., 2009, Yuan et al., 2009, Radchenko and James, 2010, Jenatton et al., 2011, Bach et al., 2012b,a, Bien et al., 2013, Lim and Hastie, 2013, Haris et al., 2015]. See also Turlach [2004] and Yuan et al. [2007], which consider modifications of the LAR algorithm Efron et al. [2004] that impose this type of condition.

In the moderate-dimensional setting that these methods are designed for, the computational issue (ii) is just about manageable. However, when p is larger—the situation of interest in this paper—it typically becomes necessary to narrow the search for interactions. Comparatively little work has been done on fitting models with interactions to data of this sort of dimension. An exception is the method of Random Intersection Trees [Shah and Meinshausen, 2014], which does not explicitly restrict the search space of interactions. However this is designed for a classification setting with a binary predictor matrix and does not fit a model but rather tries to find interactions that are marginally informative.

One option is to screen for important variables and only consider interactions involving the selected set. Wu et al. [2010] and others take this approach: the Lasso is first used to select main effects; then interactions between the selected main effects are added to the design matrix, and the Lasso is run once more to give the final model.

The success of this method relies on all main effects involved in interactions being selected in the initial screening stage. However, this may well not happen. Certain interactions may need to be included in the model before some main effects can be selected. To address this issue, Bickel et al. [2010] propose a procedure involving sequential Lasso fits which, for some predefined number K, selects K variables from each fit and then adds all interactions between those variables as candidate variables for the following fit. The process continues until all interactions to be added are already present. However, it is not clear how one should choose K: a large K may result in a large number of spurious interactions being added at each stage, whereas a small K could cause the procedure to terminate before it has had a chance to include important interactions.

Rather than adding interactions in one or more distinct stages, when variables are selected in a greedy fashion, the set of candidate interactions can be updated after each selection. This dynamic updating of interactions available for selection is present in the popular MARS procedure of Friedman [1991]. One potential problem with this approach is that particularly in high-dimensional situations, overly greedy selection can sometimes produce unstable final models and predictive performance can suffer as a consequence.

The iFORT method of Hao and Zhang [2014] applies forward selection to a dynamically updated set of candidate interactions and main effects, for the purposes of variable screening. In this work, we propose a new method we call Backtracking, for incorporating a similar model building strategy

to that of MARS and iFORT into methods based on sparsity-inducing penalty functions. Though greedy forward selection methods often work well, penalty function-based methods such as the Lasso can be more stable (see Efron et al. [2004]) and offer a useful alternative.

1.1.1 Outline of the idea

When used with the Lasso, Backtracking begins by computing the Lasso solution path, decreasing λ from ∞. A second solution path, P_2, is then produced, where the design matrix contains all main effects, and also the interaction between the first two active variables in the initial path. Continuing iteratively, subsequent solution paths $P_3, \ldots, P_T$ are computed where the set of main effects and interactions in the design matrix for the kth path is determined based on the previous path P_{k-1}. Thus if in the third path, a key interaction was included and so variable selection was then more accurate, the selection of interactions for all future paths would benefit. In this way information is used as soon as it is available, rather than at discrete stages as with the method of Bickel et al. [2010]. In addition, if all important interactions have already been included by P_3, we have a solution path unhindered by the addition of further spurious interactions.

It may seem that a drawback of our proposed approach is that the computational cost of producing all T solution paths will usually be unacceptably large. However, computation of the full collection of solution paths is typically very fast. This is because rather than computing each of the solution paths from scratch, for each new solution path P_{k+1}, we first track along the previous path P_k to find where P_{k+1} departs from P_k. This is the origin of the name Backtracking. Typically, checking whether a given trial solution is on a solution path requires much less computation than calculating the solution path itself, and so this Backtracking step is rather quick. Furthermore, when the solution paths do separate, the tail portions of the paths can be computed in parallel.

An R [R Development Core Team, 2005] package for the method is available on the author's website.

1.1.2 Organisation of the paper

The rest of the paper is organised as follows. In Section 2 we describe an example which provides some motivation for our Backtracking method. In Section 3 we develop our method in the context of the Lasso for the linear model. In Section 4, we describe how our method can be extended beyond the case of the Lasso for the linear model. In Section 5 we report the results of some simulation experiments and real data analyses that demonstrate the effectiveness of Backtracking. Finally, in Section 6, we present some theoretical results which aim to give a deeper understanding of the way in which Backtracking works. Proofs are collected in the appendix.

2 Motivation

In this section we introduce a toy example where approaches that select candidate interactions based on selected main effects will tend to perform poorly. The data follow a linear model with interactions,

$$Y_i = \sum_{j=1}^{6} \beta_j X_{ij} + \beta_7 X_{i1} X_{i2} + \beta_8 X_{i3} X_{i4} + \beta_9 X_{i5} X_{i6} + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \quad i = 1, \ldots, n, \qquad (2.1)$$

with the matrix of predictors $\mathbf{X}$ designed such that X_{i5} is uncorrelated with the response, and also with any linear combination of $\{X_{ij} : j \neq 5\}$.

The construction of $\mathbf{X}$ is as follows. First, consider (Z_{i1}, Z_{i2}, Z_{i3}) generated from a mean zero multivariate normal distribution with $\mathrm{Var}(Z_{ij}) = 1$, $j = 1, 2, 3$, $\mathrm{Cov}(Z_{i1}, Z_{i2}) = 0$ and $\mathrm{Cov}(Z_{i1}, Z_{i3}) = \mathrm{Cov}(Z_{i2}, Z_{i3}) = 1/2$. Independently generate R_{i1} and R_{i2} each of which takes only the values $\{-1, 1\}$, each with probability $1/2$. We form the ith row of the design matrix as follows:

$$\begin{aligned} X_{i1} &= R_{i1} \operatorname{sgn}(Z_{i1}) |Z_{i1}|^{1/4}, \\ X_{i2} &= R_{i1} |Z_{i1}|^{3/4}, \\ X_{i3} &= R_{i2} \operatorname{sgn}(Z_{i2}) |Z_{i2}|^{1/4}, \\ X_{i4} &= R_{i2} |Z_{i2}|^{3/4}, \\ X_{i5} &= Z_{i3}. \end{aligned}$$

The remaining X_{ij}, $j = 6, \ldots, p$ are independently generated from a standard normal distribution. Note that the random signs R_{i1} and R_{i2} ensure that X_{i5} is uncorrelated with each of $X_{i1}, \ldots, X_{i4}$. Furthermore, the fact that $X_{i1}X_{i2} = Z_{i1}$ and $X_{i3}X_{i4} = Z_{i2}$, means that when $\beta_5 = -\frac{1}{2}(\beta_7 + \beta_8)$, X_{i5} is uncorrelated with the response.

If we first select important main effects using the Lasso, for example, when p is large it is very unlikely that variable 5 will be selected. Then if we add all two-way interactions between the selected variables and fit the Lasso once more, the interaction between variables 5 and 6 will not be included. Of course, one can again add interactions between selected variables and compute another Lasso fit, and then there is a chance the interaction will be selected. Thus it is very likely that at least three Lasso fits will be needed in order to select the right variables.

Figure 1a shows the result of applying the Lasso to data generated according to (2.1) with 200 independent and identically distributed (i.i.d.) observations, $p = 500$, σ chosen to give a signal-to-noise ratio (SNR) of 4, and

$$\boldsymbol{\beta} = (-1.25, -0.75, 0.75, -0.5, -2, 1.5, 2, 2, 1)^T,$$

the coefficients having been selected so as to avoid clutter in the plots. As expected, we see variable 5 is nowhere to be seen and instead many unwanted variables are selected as λ is decreased. Figure 1b illustrates the effect of including all $p(p-1)/2$ possible interactions in the design matrix. Even in our rather moderate-dimensional situation, we are not able to recover the true signal. Though all the true interaction terms are selected, now neither variable 4 nor variable 5 are present in the solution paths and many false interactions are selected.

Although this example is rather contrived, it illustrates how sometimes the right interactions need to be augmented to the design matrix in order for certain variables to be selected. Even when interactions are only present if the corresponding main effects are too, main effects can be missed by a procedure that does not consider interactions. In fact, we can see the same phenomenon occurring when the design matrix has i.i.d. Gaussian entries (see Section 5.1). In our case here, except purely by chance, variable 5 can only be selected by the Lasso if either the interactions between variables 1 and 2 or 3 and 4 are present in the design matrix. We also see that multiple Lasso fits might be needed to have any chance of selecting the right model.

This raises the question of which tuning parameters to use in the multiple Lasso fits. One option, which we shall refer to as the iterated Lasso, is to select tuning parameters by cross-validation each

time. A drawback of this approach, though, is that the number of interactions to add can be quite large if cross-validation chooses a large active set. This is often the case when the presence of interactions makes some important main effects hard to distinguish from noise variables in the initial Lasso fit. Then cross-validation may choose a low λ in order to try to select those variables, but this would result in many noise variables also being included in the active set.

We take an alternative approach here and include suspected interactions in the design matrix as soon as possible. That is, if we progress along the solution path from $\lambda = \infty$, and two variables enter the model, we immediately add their interaction to the design matrix and start computing the Lasso again. We could now disregard the original path, but there is little to lose, and possibly much to gain, in continuing the original path in parallel with the new one. We can then repeat this process, adding new interactions when necessary, and restarting the Lasso, whilst still continuing all previous paths in parallel. We show in the next section how computation can be made very fast since many of these solution paths will share the same initial portions.

3 Backtracking with the Lasso

In this section we introduce a version of the Backtracking algorithm applied to the Lasso (1.1). First, we present a naive version of the algorithm, which is easy to understand. Later in Section 3.2, we show that this algorithm performs a large number of unnecessary calculations, and we give a far more efficient version.

3.1 A naive algorithm

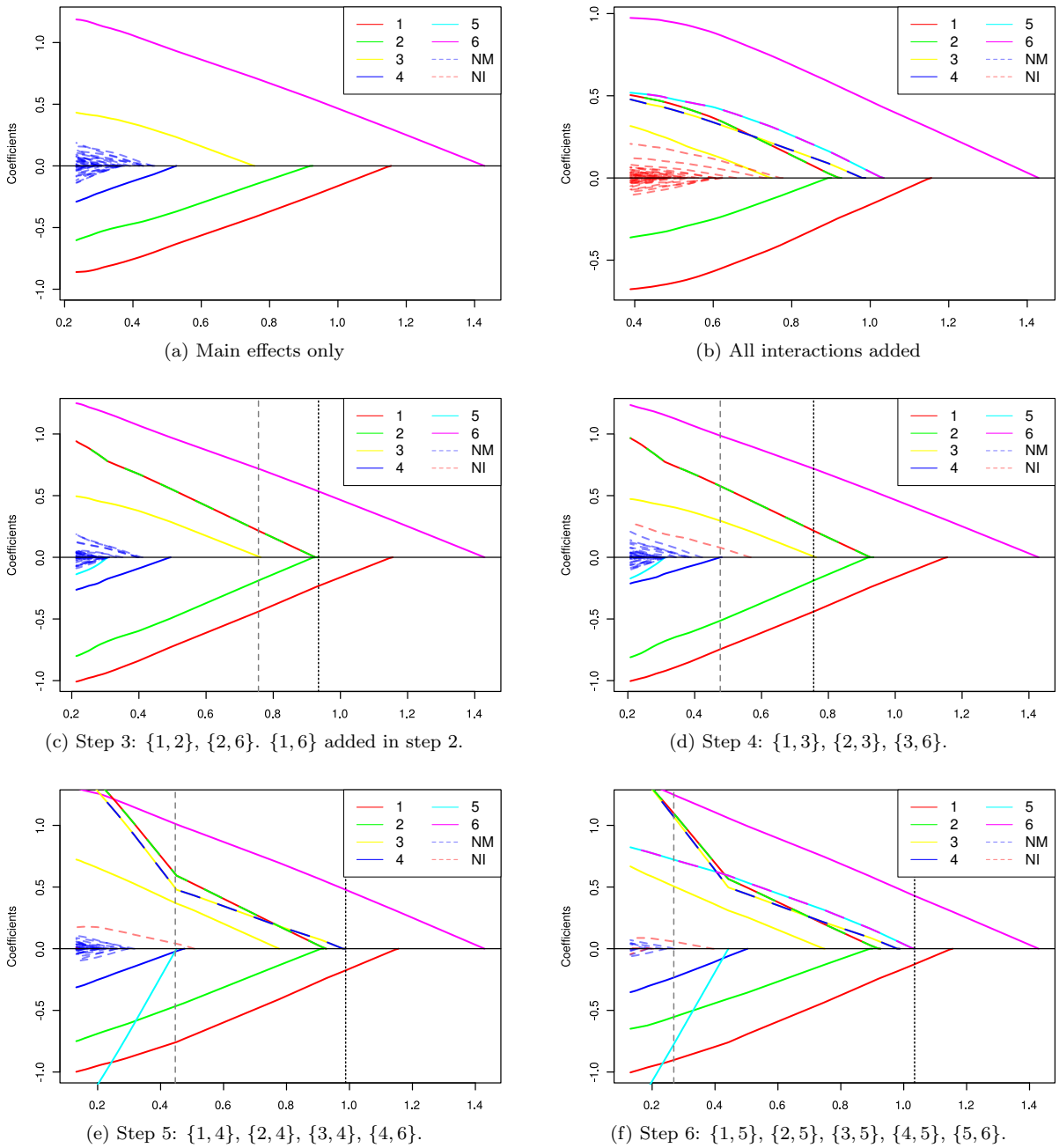
As well as a base regression procedure, the other key ingredient that Backtracking requires is a way of suggesting candidate interactions based on selected main effects, or more generally a way of suggesting higher-order interactions based on lower-order interactions. In order to discuss this and present our algorithm, we first introduce some notation concerning interactions.

Let $\mathbf{X}$ be the original $n \times p$ design matrix, with no interactions. In order to consider interactions in our models, rather than indexing variables by a single number j, we use subsets of $\{1, \ldots, p\}$. Thus by variable $\{1, 2\}$, we mean the interaction between variables 1 and 2, or in our new notation, variables $\{1\}$ and $\{2\}$. When referring to main effects $\{j\}$ however, we will often omit the braces. As we are using the Lasso as the base regression procedure here, interaction $\{1, 2\}$ will be the componentwise product of the first two columns of $\mathbf{X}$. We will write $\mathbf{X}_v \in \mathbb{R}^n$ for variable v.

The choice of whether and how to scale and centre interactions and main effects can be a rather delicate one, where domain knowledge may play a key role. In this work, we will centre all main effects, and scale them to have ℓ_2-norm $\sqrt{n}$. The interactions will be created using these centred and scaled main effects, and they themselves will also be centred and scaled to have ℓ_2-norm $\sqrt{n}$.

For C a set of subsets of $\{1, \ldots, p\}$ we can form a modified design matrix $\mathbf{X}_C$, where the columns of $\mathbf{X}_C$ are given by the variables in C, centred and scaled as described above. Thus C is the set of candidate variables available for selection when design matrix $\mathbf{X}_C$ is used. This subsetting operation will always be taken to have been performed before any further operations on the matrix, so in particular $\mathbf{X}_C^T$ means $(\mathbf{X}_C)^T$.

We will consider all associated vectors and matrices as indexed by variables, so we may speak of component $\{1, 2\}$ of $\boldsymbol{\beta}$, denoted $\beta_{\{1,2\}}$, if $\boldsymbol{\beta}$ were multiplying a design matrix which included $\{1, 2\}$. Further, for any collection of variables A, we will write $\boldsymbol{\beta}_A$ for the subvector whose components are



(a) Main effects only

(b) All interactions added

(c) Step 3: $\{1,2\}$, $\{2,6\}$. $\{1,6\}$ added in step 2.

(d) Step 4: $\{1,3\}$, $\{2,3\}$, $\{3,6\}$.

(e) Step 5: $\{1,4\}$, $\{2,4\}$, $\{3,4\}$, $\{4,6\}$.

(f) Step 6: $\{1,5\}$, $\{2,5\}$, $\{3,5\}$, $\{4,5\}$, $\{5,6\}$.

Figure 1: For data generated as described in Section 2, the coefficient paths against λ of the Lasso with main effects only, (a); the Lasso with all interactions added, (b); and Backtracking with $k = 3, \ldots, 6$, ((c)–(d)); when applied to the example in Section 2. Below the Backtracking solution paths we give $C_k \setminus C_{k-1}$: the interactions which have been added in the current step. The solid red, green, yellow, blue, cyan and magenta lines trace the coefficients of variables $1, \ldots, 6$ respectively, with the alternately coloured lines representing the corresponding interactions. The dotted blue and red coefficient paths indicate noise main effect ('NM') and interaction ('NI') terms respectively. Vertical dotted black and dashed grey lines give the values of λ_k^{start} and λ_k^{add} respectively.

those indexed by A. To represent an arbitrary variable which may be an interaction, we shall often use v or u and reserve j to index main effects.

We will often need to express the dependence of the Lasso solution $\hat{\boldsymbol{\beta}}$ (1.1) on the tuning parameter λ and the design matrix used. We shall write $\hat{\boldsymbol{\beta}}(\lambda, C)$ when $\mathbf{X}_C$ is the design matrix. We will denote the set of active components of a solution $\hat{\boldsymbol{\beta}}$ by $\mathcal{A}(\hat{\boldsymbol{\beta}}) = \{v : \hat{\beta}_v \neq 0\}$.

We now introduce a function $\mathcal{I}$ that given a set of variables A, suggests a set of interactions to add to the design matrix. The choice of $\mathcal{I}$ we use here is as follows:

$$\mathcal{I}(A) = \{v \subseteq \{1, \ldots, p\} : \text{ for all } u \subsetneq v,\, u \neq \emptyset,\, u \in A\}.$$

In other words, $\mathcal{I}(A)$ is the set of variables not in A, all of whose corresponding lower order interactions are present in A. To ease notation, when A contains only main effects $j_1, \ldots, j_s$, we will write $\mathcal{I}(j_1, \ldots, j_s) = \mathcal{I}(A)$. For example, $\mathcal{I}(1,2) = \{\{1,2\}\}$, and $\mathcal{I}(1,2,3) = \{\{1,2\},\{2,3\},\{1,3\}\}$. Note $\{1,2,3\} \notin \mathcal{I}(1,2,3)$ as the lower order interaction $\{1,2\}$ of $\{1,2,3\}$ is not in $\{\{1\},\{2\},\{3\}\}$, for example. Other choices for $\mathcal{I}$ can be made, and we discuss some further possibilities in Section 4.

Backtracking relies on a path algorithm for computing the Lasso on a grid of λ values $\lambda_1 > \cdots \lambda_L$. Several algorithms are available and coordinate descent methods [Friedman et al., 2010] appears to work well in practice.

We are now in a position to introduce a naive version of our Backtracking algorithm applied to the Lasso (Algorithm 1). We will assume that the response $\mathbf{Y}$ is centred in addition to the design matrix, so no intercept term is necessary.

Algorithm 1 A naive version of Backtracking with the Lasso

Set T to be the (given) maximum number of candidate interaction sets to generate. Let the initial candidate set consist of just main effects: $C_1 = \{\{1\}, \ldots, \{p\}\}$. Set the index for the candidate sets $k = 1$. Let $\lambda_1^{\text{start}} = \lambda_1$, the largest λ value on the grid. In the steps which follow, we maintain a record of the set of variables which have been non-zero at any point in the algorithm up to the current point (an "ever active set", A).

1. Compute the solution path of the Lasso with candidate set C_k from λ_k^{start} onwards until the ever active set A has $\mathcal{I}(A) \nsubseteq C_k$ (if the smallest λ value on the grid is reached then go to 5). Let the λ value where this occurs be λ_k^{add}. We will refer to this solution path as P_k.

2. Set $C_{k+1} = C_k \cup \mathcal{I}(A)$ so the next candidate set contains all interactions between variables in the ever active set.

3. Set $\lambda_{k+1}^{\text{start}} = \lambda_1$.

4. Increment k. If $k > T$ go to 5, otherwise go back to 1.

5. For each k complete the solution path P_k by continuing it until $\lambda = \lambda_L$. Computing these final pieces of the solution paths can be done in parallel.

The algorithm computes Lasso solution paths whose corresponding design matrices include interactions chosen based on previous paths. The quantity λ_k^{add} records the value of λ at which interaction terms were added to the set of candidates C_k. Here λ_k^{start} is a redundant quantity and can be replaced everywhere with λ_1 to give the same algorithm. We include it at this stage though

to aid with the presentation of an improved version of the algorithm where λ_k^{start} in general takes values other than λ_1. We note that the final step of completing the solution paths can be carried out as the initial paths are being created, rather than once all initial paths have been created. Though here the algorithm can include three-way or even higher order interactions, it is straightforward to restrict the possible interactions to be added to first-order interactions, for example.

3.2 An improved algorithm

The process of performing multiple Lasso fits is computationally cumbersome, and an immediate gain in efficiency can be realised by noticing that the final collection of solution paths is in fact a tree of solutions: many of the solution paths computed will share the same initial portions.

To discuss this, we first recall the KKT conditions for the Lasso dictate that $\hat{\boldsymbol{\beta}}$ is a solution to (1.1) when the design matrix is $\mathbf{X}_C$ if and only if

$$\tfrac{1}{n}\mathbf{X}_v^T(\mathbf{Y} - \mathbf{X}_C\hat{\boldsymbol{\beta}}) = \lambda \text{sgn}(\hat{\beta}_v) \quad \text{for } \hat{\beta}_v \neq 0 \tag{3.1}$$

$$\tfrac{1}{n}|\mathbf{X}_v^T(\mathbf{Y} - \mathbf{X}_C\hat{\boldsymbol{\beta}})| \leq \lambda \quad \text{for } \hat{\beta}_v = 0. \tag{3.2}$$

Note the $\hat{\mu}\mathbf{X}_v^T\mathbf{1}$ term vanishes as the columns of $\mathbf{X}_C$ are centred.

We see that if for some λ

$$\tfrac{1}{n}\|\mathbf{X}_{C_{k+1}\setminus C_k}^T(\mathbf{Y} - \mathbf{X}_{C_k}\hat{\boldsymbol{\beta}}(\lambda, C_k))\|_\infty \leq \lambda, \tag{3.3}$$

then

$$\hat{\boldsymbol{\beta}}_{C_{k+1}\setminus C_k}(\lambda, C_{k+1}) = \mathbf{0}, \qquad \hat{\boldsymbol{\beta}}_{C_k}(\lambda, C_{k+1}) = \hat{\boldsymbol{\beta}}(\lambda, C_k).$$

Thus given solution path P_k, we can attempt to find the smallest λ_l such that (3.3) holds. Up to that point then, path P_{k+1} will coincide with P_k and so those Lasso solutions need not be re-computed. Note that verifying (3.3) is a computationally simple task requiring only $O(|C_{k+1}\setminus C_k|n)$ operations.

Our final Backtracking algorithm therefore replaces step 3 of Algorithm 1 with the following:

3a. Find the smallest $\lambda_1 \geq \lambda_l \geq \lambda_k^{\text{add}}$ such that (3.3) holds with $\lambda = \lambda_l$ and set this to be $\lambda_{k+1}^{\text{start}}$. If no such λ_l exists, set $\lambda_{k+1}^{\text{start}}$ to be λ_1.

Figures 1c–1f show steps 3–6 (i.e. $k = 3, \ldots, 6$) of Backtracking applied to the example described in Section 2. Note that Figure 1a is in fact step 1. Step 2 is not shown as the plot looks identical to that in Figure 1a. We see that when $k = 6$, we have a solution path where all the true variable and interaction terms are active before any noise variables enter the coefficient plots.

We can further speed up the algorithm by first checking if P_k coincides with P_{k+1} at λ_k^{add}. If not, we can perform a bisection search to find any point where P_k and P_{k+1} agree, but after which they disagree. This avoids checking (3.3) for every λ_l up to λ_k^{add}. We will work with the simpler version of Backtracking here using step 3a, but use this faster version in our implementation.

4 Further applications of Backtracking

Our Backtracking algorithm has been presented in the context of the Lasso for the linear model. However, the real power of the idea is that it can be incorporated into any method that produces a

path of increasingly complex sparse solutions by solving a family of convex optimisation problems parametrised by a tuning parameter. For the Backtracking step, the KKT conditions for these optimisation problems provide a way of checking whether a given trial solution is an optimum. As in the case of the Lasso, checking whether the KKT conditions are satisfied typically requires much less computational effort than computing a solution from scratch. Below we briefly sketch some applications of Backtracking to a few of the many possible methods with which it can be used.

4.1 Multinomial regression

An example, which we apply to real data in Section 5.2, is multinomial regression with a group Lasso [Yuan and Lin, 2006] penalty. Consider n observations of a categorical response that takes J levels, and p associated covariates. Let $\mathbf{Y}$ be the indicator response matrix, with ijth entry equal to 1 if the ith observation takes the jth level, and 0 otherwise. We model

$$\mathbb{P}(Y_{ij} = 1) := \Pi_{ij}(\boldsymbol{\mu}^*, \boldsymbol{\beta}^*; \mathbf{X}_{S^*}) := \frac{\exp\left(\mu_j^* + \left(\mathbf{X}_{S^*}\boldsymbol{\beta}_j^*\right)_i\right)}{\sum_{j'=1}^{J} \exp\left(\mu_{j'}^* + \left(\mathbf{X}_{S^*}\boldsymbol{\beta}_{j'}^*\right)_i\right)}.$$

Here μ^* is a vector of intercept terms and $\boldsymbol{\beta}^*$ is a $|S^*| \times J$ matrix of coefficients; $\boldsymbol{\beta}_j^*$ denotes the jth column of $\boldsymbol{\beta}^*$. This model is over-parametrised, but regularisation still allows us produce estimates of $\boldsymbol{\mu}^*$ and $\boldsymbol{\beta}^*$ and hence also of $\boldsymbol{\Pi}$ (see Friedman et al. [2010]). When our design matrix is $\mathbf{X}_C$, these estimates are given by $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\beta}}) := \underset{\boldsymbol{\mu},\boldsymbol{\beta}}{\arg\min}\, Q(\boldsymbol{\mu}, \boldsymbol{\beta}; \lambda)$ where

$$Q(\boldsymbol{\mu}, \boldsymbol{\beta}; \lambda) \;:=\; \tfrac{1}{n}\sum_{j=1}^{J} \mathbf{Y}_j^T(\mu_j\mathbf{1} + \mathbf{X}_C\boldsymbol{\beta}_j) - \tfrac{1}{n}\mathbf{1}^T \log\left(\sum_{j=1}^{J} \exp(\mu_j\mathbf{1} + \mathbf{X}_C\boldsymbol{\beta}_j\right) + \lambda \sum_{v \in C} \|(\boldsymbol{\beta}^T)_v\|_2.$$

The functions log and exp are to be understood as applied componentwise and the rows of $\boldsymbol{\beta}$ are indexed by elements of C. To derive the Backtracking step for this situation, we turn to the KKT conditions which characterise the minima of Q:

$$\begin{aligned}
\tfrac{1}{n}\{\mathbf{Y}^T - \boldsymbol{\Pi}^T(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\beta}}; \mathbf{X}_C)\}\mathbf{1} &= \mathbf{0}, \\
\tfrac{1}{n}\{\mathbf{Y}^T - \boldsymbol{\Pi}^T(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\beta}}; \mathbf{X}_C)\}\mathbf{X}_v &= -\lambda \frac{(\hat{\boldsymbol{\beta}}^T)_v}{\|(\hat{\boldsymbol{\beta}}^T)_v\|_2} \quad \text{for } (\hat{\boldsymbol{\beta}}^T)_v \neq \mathbf{0}, \\
\tfrac{1}{n}\|\{\mathbf{Y}^T - \boldsymbol{\Pi}^T(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\beta}}; \mathbf{X}_C)\}\mathbf{X}_v\|_2 &\leq \lambda \quad \text{for } (\hat{\boldsymbol{\beta}}^T)_v = \mathbf{0}.
\end{aligned}$$

Thus, analogously to (3.3), for $D \supsetneq C$, $(\hat{\boldsymbol{\beta}}^T(\lambda, D))_{D\setminus C} = \mathbf{0}$ and $(\hat{\boldsymbol{\beta}}^T(\lambda, D))_C = \hat{\boldsymbol{\beta}}^T(\lambda, C)$ if and only if

$$\max_{v \in D\setminus C} \tfrac{1}{n}\|\{\mathbf{Y}^T - \boldsymbol{\Pi}^T(\hat{\boldsymbol{\mu}}(\lambda, C), \hat{\boldsymbol{\beta}}(\lambda, C); \mathbf{X}_C)\}\mathbf{X}_v\|_2 \leq \lambda.$$

4.2 Structural sparsity

Although in our Backtracking algorithm, interaction terms are only added as candidates for selection when all their lower order interactions and main effects are active, this hierarchy in the selection of candidates does not necessarily follow through to the final model: one can have first-order

interactions present in the final model without one or more of their main effects, for example. One way to enforce the hierarchy constraint in the final model is to use a base procedure which obeys the constraint itself. Examples of such base procedures are provided by the Composite Absolute Penalties (CAP) family [Zhao et al., 2009].

Consider the linear regression setup with interactions. For simplicity we only describe Backtracking with first-order interactions. Let C be the candidate set and let $I = C \setminus C_1$ be the (first-order) interaction terms in C. In order to present the penalty, we borrow some notation from Combinatorics. Let $C_1^{(r)}$ denote the set of r-subsets of C_1. For $A \subseteq C_1^{(r)}$ and $r \geq 1$, define

$$\partial_l(A) = \{v \in C_1^{(r-1)} : v \subset u \text{ for some } u \in A\}$$
$$\partial_u(A) = \{v \in C_1^{(r+1)} : v \subset u \text{ for some } u \in A\}$$

These are known as the *lower shadow* and *upper shadow* respectively [Bollobás, 1986].

Our objective function Q is given by

$$Q(\mu, \boldsymbol{\beta}) = \tfrac{1}{2n}\|\mathbf{Y} - \mu\mathbf{1} - \mathbf{X}_C\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}_{C_1 \setminus \partial_l(I)}\|_1 + \lambda \sum_{v \in \partial_l(I)} \|\boldsymbol{\beta}_{\{v\} \cup (\partial_u(\{v\}) \cap I)}\|_\gamma + \lambda\|\boldsymbol{\beta}_I\|_1,$$

where $\gamma > 1$. For example, if $C = \{\{1\}, \ldots, \{4\}, \{1,2\}, \{2,3\}\}$, then omitting the factor of λ, the penalty terms in Q are

$$|\beta_4| + \|(\beta_1, \beta_{\{1,2\}})^T\|_\gamma + \|(\beta_2, \beta_{\{1,2\}}, \beta_{\{2,3\}})^T\|_\gamma + \|(\beta_3, \beta_{\{2,3\}})^T\|_\gamma + |\beta_{\{1,2\}}| + |\beta_{\{2,3\}}|.$$

The form of this penalty forces interactions to enter the active set only after or with their corresponding main effects.

The KKT conditions for this optimisation take a more complicated form than those for the Lasso. Nevertheless, checking they hold for a trial solution is an easier task than computing a solution.

4.3 Nonlinear models

If a high-dimensional additive modelling method [Ravikumar et al., 2009, Meier et al., 2009] is used as the base procedure, it is possible to fit nonlinear models with interactions. Here each variable is a collection of basis functions, and to add an interaction between variables, one adds the tensor product of the two collections of basis functions, penalizing the new interaction basis functions appropriately. Structural sparsity approaches can also be used here. The VANISH method of Radchenko and James [2010] uses a CAP-type penalty in nonlinear regression, and this can be used as a base procedure in a similar way to that sketched above.

4.4 Introducing more candidates

In our description of the Backtracking algorithm, we only introduce an interaction term when *all* of its lower order interactions and main effects are active. Another possibility, in the spirit of MARS [Friedman, 1991], is to add interaction terms when *any* of their lower order interactions or main effects are active. As at the kth step of Backtracking, there will be roughly kp extra candidates, an approach that can enforce the hierarchical constraint may be necessary to allow main effects to be selected from amongst the more numerous interaction candidates. The key point to note is that if

Scenario	S_2^*
1	$\emptyset$
2	$\emptyset$
3	$\{\{1,2\},\{3,4\},\{5,6\}\}$
4	$\{\{1,2\},\{1,3\},\ldots,\{1,6\}\}$
5	$\mathcal{I}(1,2,3) \cup \mathcal{I}(4,5,6)$

Table 1: Simulation settings.

the algorithm is terminated after T steps, we are having to deal with roughly at most Tp variables rather than $O(p^2)$, the latter coming from including all first-order interactions.

Another option proposed by a referee is to augment the initial set of candidates with interactions selected through a simple marginal screening step. If only pairwise interactions are considered here, then this would require $O(p^2 n)$ operations. Though this would be infeasible for very large p, for moderate p this would allow important interactions whose corresponding main effects are not strong to be selected.

5 Numerical results

5.1 Simulations

In this section, we consider five numerical studies designed to demonstrate the effectiveness of Backtracking with the Lasso and also highlight some of the drawbacks of using the Lasso with main effects only, when interactions are present. In each of the five scenarios, we generated 200 design matrices with $n = 250$ observations and $p = 1000$ covariates. The rows of the design matrices were sampled independently from $N_p(\mathbf{0}, \mathbf{\Sigma})$ distributions. The covariance matrix $\mathbf{\Sigma}$ was chosen to be the identity in all scenarios except scenario 2, where

$$\Sigma_{ij} = 0.75^{-||i-j|-p/2|+p/2}.$$

Thus in this case, the correlation between the components decays exponentially with the distance between them in $\mathbb{Z}/p\mathbb{Z}$.

We created the responses according to the linear model with interactions and set the intercept to 0:

$$\mathbf{Y} = \mathbf{X}_{S^*}\boldsymbol{\beta}^*_{S^*} + \boldsymbol{\varepsilon}, \quad \varepsilon_i \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2). \tag{5.1}$$

The error variance σ^2 was chosen to achieve a signal-to-noise ratio (SNR) of either 2 or 3. The set of main effects in S^*, S_1^*, was $1, \ldots, 10$. The subset of variables involved in interactions was $1, \ldots, 6$. The set of first-order interactions in S^* chosen in the different scenarios, S_2^*, is displayed in Table 1, and we took $S^* = S_1^* \cup S_2^*$ so S^* contained no higher order interactions. In each simulation run, $\boldsymbol{\beta}^*_{S_1^*}$ was fixed and given by

$$(2, -1.5, 1.25, -1, 1, -1, 1, 1, 1, 1)^T.$$

Each component of $\boldsymbol{\beta}^*_{S_2^*}$ was chosen to be $\sqrt{\|\boldsymbol{\beta}^*_{S_1^*}\|_2^2 / |S_1^*|}$. Thus the squared magnitude of the interactions was equal to average of the squared magnitudes of the main effects.

In all of the scenarios, we applied four methods: the Lasso using only the main effects; iterated Lasso fits; marginal screening for interactions followed by the Lasso; and the Lasso with Backtracking. Note that due to the size of p in these examples, most of the methods for finding interactions in lower-dimensional data discussed in Section 1, are computationally impractical here.

For the iterated Lasso fits, we repeated the following process. Given a design matrix, first fit the Lasso. Then apply 5-fold cross-validation to give a λ value and associated active set. Finally add all interactions between variables in this active set to the design matrix, ready for the next iteration. For computational feasibility, the procedure was terminated when the number of variables in the design matrix exceeded $p + 250 \times 249/2$.

With the marginal screening approach, we selected the $2p$ interactions with the largest marginal correlation with the response and added them to the design matrix. Then a regular Lasso was performed on the augmented matrix of predictors.

Additionally, in scenarios 3–5, we applied the Lasso with all main effects and only the true interactions. This theoretical Oracle approach provided a gold standard against which to test the performance of Backtracking.

We used the procedures mentioned to yield active sets on which we applied OLS to give a final estimator. To select the tuning parameters of the methods we used cross-validation randomly selection 5 folds but repeating this a total of 5 times to reduce the variance of the cross-validation scores. Thus for each λ value we obtained an estimate of the expected prediction error that was an average over the observed prediction errors on 25 (overlapping) validation sets of size $n/5 = 50$. Note that for both Backtracking and the iterated Lasso, this form of cross-validation chose not just a λ value but also a path rank. When using Backtracking, the size of the active set was restricted to 50 and the size of C_k to $p + 50 \times 49/2 = 1225$, so T was at most 50.

In scenarios 1 and 2, the results of the methods were almost indistinguishable except that the screening approach performed far worse in scenario 1 where it tended to select several false interactions which in turn hampered the selection of main effects and resulted in a much larger prediction error.

The results of scenarios 3–5, where the signal contains interactions, are more interesting and given in Table 2. For each scenario, method and SNR level, we report 5 statistics. 'L_2-sq' is the expected squared distance of the signal $\mathbf{f}^*$ and our prediction functions $\hat{\mathbf{f}}$ based on training data $(\mathbf{Y}_{\text{train}}, \mathbf{X}_{\text{train}})$, evaluated at a random independent test observation $\mathbf{x}_{\text{new}}$:

$$\mathbb{E}_{\mathbf{x}_{\text{new}}, \mathbf{Y}_{\text{train}}, \mathbf{X}_{\text{train}}} (\mathbf{f}^*\{\mathbf{x}_{\text{new}}) - \hat{\mathbf{f}}(\mathbf{x}_{\text{new}}; \mathbf{Y}_{\text{train}}, \mathbf{X}_{\text{train}})\}^2.$$

'FP Main' and 'FP Inter' are the numbers of noise main effects and noise interaction terms respectively, incorrectly included in the final active set. 'FN Main' and 'FN Inter' are the numbers of true main effects and interaction terms respectively, incorrectly excluded from the final active set.

For all the statistics presented, lower numbers are to be preferred. However, the higher number of false selections incurred by both Backtracking and the Oracle procedure compared to using the main effects only or iterated Lasso fits, is due to the model selection criterion being the expected prediction error. It should not be taken as an indication that the latter procedures are performing better in these cases.

Backtracking performs best out of the four methods compared here. Note that under all of the settings, iterated Lasso fits incorrectly selects more interaction terms than Backtracking. We see that the more careful way in which Backtracking adds candidate interactions, helps here. Unsurprisingly, fitting the Lasso on just the main effects performs rather poorly in terms of predictive

performance. However, it also fails to select important main effects; Backtracking and Iterates have much lower main effect false negatives. The screening approach appears to perform worst here. This is partly because it is not making use of the fact that in all of the examples considered, the main effects involved in interactions are also informative. However, its poor performance is also due the fact that too many false interactions are added to the design matrix after the screening stage. Reducing the number added may help to improve results, but choosing the number of interactions to include via cross-validation, for example, would be computationally costly, unless a Backtracking-type strategy of the sort introduced in this paper were used. We also note that for very large p, marginal screening of interactions would be infeasible due to the quadratic scaling in complexity with p.

Scenario	Statistic	SNR = 2					SNR = 3				
		Main	Iterate	Screening	Backtracking	Oracle	Main	Iterate	Screening	Backtracking	Oracle
3	L_2-sq	6.95	1.40	12.87	1.21	0.82	5.67	0.27	9.24	0.27	0.18
	FP Main	3.18	2.43	0.01	2.89	3.19	1.91	0.65	0.00	0.73	0.79
	FN Main	1.26	0.38	7.24	0.24	0.14	0.52	0.05	5.14	0.04	0.01
	FP Inter	0.00	0.93	11.05	0.45	0.00	0.00	0.27	13.57	0.12	0.00
	FN Inter	3.00	0.18	2.06	0.14	0.01	3.00	0.03	1.39	0.04	0.00
4	L_2-sq	12.05	3.25	17.68	2.72	1.68	10.44	0.63	15.19	0.41	0.31
	FP Main	2.22	3.88	0.02	5.34	7.05	2.58	1.80	0.04	2.08	2.21
	FN Main	3.12	0.90	8.13	0.61	0.26	1.77	0.11	6.94	0.04	0.00
	FP Inter	0.00	2.50	12.33	0.77	0.00	0.00	1.77	17.90	0.28	0.00
	FN Inter	5.00	0.66	4.07	0.51	0.08	5.00	0.08	3.39	0.03	0.00
5	L_2-sq	14.12	5.08	19.96	4.52	2.14	12.84	1.56	16.99	1.17	0.44
	FP Main	3.07	4.75	0.02	5.87	8.57	3.43	3.01	0.05	3.23	3.77
	FN Main	3.20	1.26	8.26	0.98	0.33	2.35	0.25	7.00	0.19	0.02
	FP Inter	0.00	3.28	17.97	0.87	0.00	0.00	3.05	21.92	0.55	0.00
	FN Inter	6.00	1.34	5.00	1.23	0.14	6.00	0.39	4.14	0.30	0.00

Table 2: Simulation results.

5.2 Real data analyses

In this section, we look at the performance of Backtracking using two base procedures, the Lasso for the linear model and the Lasso for multinomial regression, on a regression and a classification data set. As competing methods, we consider simply using the base procedures ('Main'), iterated Lasso fits ('Iterated'), Lasso following marginal screening for interactions ('Screening'), Random Forests [Breiman, 2001], hierNet [Bien et al., 2013] and MARS [Friedman, 1991] (implemented using Hastie et al. [2013]). Note that we do not view the latter two methods as competitors of Backtracking, as they are designed for use on lower dimensional datasets than Backtracking is capable of handling. However, it is still interesting to see how the methods perform on data of dimension that is perhaps approaching the upper end of what is easily manageable for methods such as hierNet and MARS, but at the lower end of what one might use Backtracking on.

Below we describe the datasets used which are both from the UCI machine learning repository [Asuncion and Newman, 2007].

5.2.1 Communities and Crime

This dataset available at `http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized` contains crime statistics for the year 1995 obtained from FBI data, and national census data from 1990, for various towns and communities around the USA. We took violent crimes per capita as our response: violent crime being defined as murder, rape, robbery, or assault. The data set contains two different estimates of the populations of the communities: those from the 1990 census and those from the FBI database in 1995. The latter was used to calculate our desired response using the number of cases of violent crimes. However, in several cases, the FBI population data seemed suspect and we discarded all observations where the maximum of the ratios of the two available population estimates differed by more than 1.25. In addition, we removed all observations that were missing a response and several variables for which the majority of values were missing. This resulted in a dataset with $n = 1903$ observations and $p = 101$ covariates. The response was scaled to have empirical variance 1.

5.2.2 ISOLET

This data set consists of $p = 617$ features based on the speech waveforms generated from utterances of each letter of the English alphabet. The task is to learn a classifier which can determine the letter spoken based on these features. The dataset is available from `http://archive.ics.uci.edu/ml/datasets/ISOLET`; see Fanty and Cole [1991] for more background on the data. We consider classification on the notoriously challenging E-set consisting of the letters 'B', 'C', 'D', 'E', 'G', 'P', 'T', 'V' and 'Z' (pronounced 'zee'). As there were 150 subjects and each spoke each letter twice, we have $n = 2700$ observations spread equally among 9 classes. The dimension of this data is such that MARS and hierNet could not be applied.

5.3 Methods and results

For the Communities and crime data set, we used the Lasso for the linear model as the base regression procedure for Backtracking and Iterates. Since the per capita violent crime response was always non-negative, the positive part of the fitted values was taken. For Main, Backtracking, Iterates, Screening and hierNet, we employed 5-fold cross-validation with squared error loss to select tuning parameters. For MARS we used the default settings for pruning the final fits using generalised cross-validation. With Random Forests, we used the default settings on both data sets. For the classification example, penalised multinomial regression was used (see Section 4.1) as the base procedure for Backtracking and Iterates, and the deviance was used as the loss function for 5-fold cross-validation. In all of the methods except Random Forests, we only included first-order interactions. When using Backtracking, we also restricted the size of C_k to $p+50\times 49/2 = p+1225$.

To evaluate the procedures, we randomly selected 2/3 for training and the remaining 1/3 was used for testing. This was repeated 200 times for each of the data sets. Note that we have specifically chosen data sets with n large as well as p large. This is to ensure that comparisons between the performances of the methods can be made with more accuracy. For the regression example, out-of-sample squared prediction error was used as a measure of error; for the classification example, we used out-of-sample misclassification error with 0–1 loss. The results are given in Table 3.

Random Forests has the lowest prediction error on the regression dataset, with Backtracking not far behind, whilst Backtracking wins in the classification task, and in fact achieves strictly lower misclassification error than all the other methods on 90% of all test samples. Note that a

Method	Error	
	Communities and crime	ISOLET
Main	0.414 (6.5×10^{-3})	0.0641 (4.7×10^{-4})
Iterate	0.384 $(5.9 \times 10^{-3}$	0.0641 (4.7×10^{-4})
Screening	0.390 (7.8×10^{-3})	-
Backtracking	0.365 (3.7×10^{-3})	0.0563 (4.5×10^{-4})
Random Forest	0.356 (2.4×10^{-3})	0.0837 (6.0×10^{-4})
hierNet	0.373 (4.7×10^{-3})	-
MARS	5580.586 (3.1×10^{3})	-

Table 3: Real data analyses results. Average error rates over 200 training–testing splits are given, with standard deviations of the results divided by $\sqrt{200}$ in parentheses.

direct comparison with Random Forests is perhaps unfair, as the latter is a black-box procedure whereas Backtracking is aiming for a more interpretable model.

MARS performs very poorly indeed on the regression dataset. The enormous prediction error is caused by the fact that whenever observations corresponding to either New York or Los Angeles were in the test set, MARS predicted their responses to be far larger than they were. However, even with these observations removed, the instability of MARS meant that it was unable to give much better predictions than an intercept-only model.

HierNet performs well on this dataset, though it is worth noting that we had to scale the interactions to have the same ℓ_2-norm as the main effects to get such good results (the default scaling produced error rates worse than that of an intercept-only model). Backtracking does better here. One reason for this is that the because the main effects are reasonably strong in this case, a low amount of penalisation works well. However, because with hierNet, the penalty on the interactions is coupled with the penalty on the main effects, the final model tended to include close to two hundred interaction terms. The Screening approach similarly suffers from including too many interactions and performs only a little better than a main effects only fit.

The way that Backtracking creates several solution paths with varying numbers of interaction terms means that it is possible to fit main effects and a few interactions using a low penalty without this low penalisation opening the door to many other interaction terms. The iterated Lasso approach also has this advantage, but as the number of interactions are increased in discrete stages, it can miss a candidate set with the right number of interactions that may be picked up by the more continuous model building process used by Backtracking. This occurs in a rather extreme way with the ISOLET dataset where, since in the first stage of the iterated Lasso, cross-validation selected far too many variables (> 250), the second and subsequent steps could not be performed. This is why the results are identical to using the main effects alone.

6 Theoretical properties

Our goal in this section is to understand under what circumstances Backtracking with the Lasso can arrive at a set of candidates, C^*, that contains all of the true interactions, and only a few false interactions. On the event on which this occurs, we can then apply many of the existing results on the Lasso, to show that the solution path $\hat{\boldsymbol{\beta}}(\lambda, C^*)$ has certain properties. As an example, in Section 6.2 we give sufficient conditions for the existence of a λ^* such that $\{v : \hat{\beta}_v(\lambda^*, C^*) \neq 0\}$

equals the true set of variables.

We work with the normal linear model with interactions,

$$\mathbf{Y} = \mu^* \mathbf{1} + \mathbf{X}_{S^*} \boldsymbol{\beta}^*_{S^*} + \boldsymbol{\varepsilon}, \tag{6.1}$$

where $\varepsilon_i \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2)$, and to ensure identifiability, $\mathbf{X}_{S^*}$ has full column rank. We will assume that $S^* = S_1^* \cup S_2^*$, where S_1^* and S_2^* are main effects and two-way interactions respectively. Let the interacting main effects be I^*; formally, I^* is the smallest set of main effects such that $\mathcal{I}(I^*) \supseteq S_2^*$. Assume $I^* \subseteq S_1^*$ so interactions only involve important main effects. Let $s_l = |S_l^*|$, $l = 1, 2$ and set $s = s_1 + s_2$. Define $C^* = C_1 \cup \mathcal{I}(S_1^*)$. Note that C^* contains S^* but not additional interactions from any variables from $C_1 \setminus S_1^*$.

Although the Backtracking algorithm was presented for a base path algorithm that computed solutions at only discrete values, for the following results, we need to imagine an idealised algorithm which computes the entire path of solutions. In addition, we will assume that we only allow first-order interactions in the Backtracking algorithm, and that $T \geq s_1$.

We first consider the special case where the design matrix is derived from a random matrix with i.i.d. multivariate normal rows, before describing a result for fixed design.

6.1 Random normal design

Let the random matrix $\mathbf{Z}$ have independent rows distributed as $N_p(\mathbf{0}, \boldsymbol{\Sigma})$. Suppose that $\mathbf{X}_{C_1}$, the matrix of main effects, is formed by scaling and centring $\mathbf{Z}$. We consider an asymptotic regime where $\mathbf{X}$, $\mathbf{f}^*$, S^*, σ^2 and p can all change as $n \to \infty$, though we will suppress their dependence on n in the notation. Furthermore, for sets of indices $S, M \subseteq \{1, \ldots, p\}$, let $\boldsymbol{\Sigma}_{S,M} \in \mathbb{R}^{|S| \times |M|}$ denote the submatrix of $\boldsymbol{\Sigma}$ formed from those rows and columns of $\boldsymbol{\Sigma}$ indexed by S and M respectively. For any positive semi-definite matrix $\mathbf{A}$, we will let $c_{\min}(\mathbf{A})$ denote its minimal eigenvalue. For sequences a_n, b_n, by $a_n \succ b_n$ we mean $b_n = o(a_n)$.

We make the following assumptions.

A1. $c_{\min}(\boldsymbol{\Sigma}_{S_1^*, S_1^*}) \geq c_* > 0$.

A2. $\sup_{\boldsymbol{\tau} \in \mathbb{R}^{s_1} : \|\boldsymbol{\tau}\|_\infty \leq 1} \|\boldsymbol{\Sigma}_{N, S_1^*} \boldsymbol{\Sigma}^{-1}_{S_1^*, S_1^*} \boldsymbol{\tau}\|_\infty \leq \delta < 1$.

A3. $s_1^4 \log(p)/n \to 0$ and $s_1^8 \log(s_1)^2/n \to 0$.

A4. For $j \in I^*$,

$$\min_{j \in I^*} |\beta_j^*| \succ \frac{s_1(\sigma\sqrt{\log p} + \sqrt{s_1 + \log p})}{\sqrt{n}} + \frac{s_1\sqrt{\log(s_1)}}{n^{1/3}}.$$

A5. $\|\boldsymbol{\beta}^*_{S_2^*}\|_1$ is bounded as $n \to \infty$.

A1 is a standard assumption in high-dimensional regression and is, for example, implied by the compatibility constant of Bühlmann and van de Geer [2011b] being bounded away from zero. A2 is closely related to irrepresentable conditions (see Meinshausen and Bühlmann [2006], Zhao and Yu [2006], Zou [2006], Bühlmann and van de Geer [2011b], Wainwright [2009]), which are used for proving variable selection consistency of the Lasso. Note that although here the signal may contain interactions our irrepresentable-type condition only involves main effects.

A3 places restrictions on the rates at which s_1 and p can increase with n. The first condition involving $\log(p)$ is somewhat natural as $s_1^2 \log(p)/n \to 0$ would typically be required in order to show ℓ_1 estimation consistency of $\boldsymbol{\beta}$ where only s_1 main effects are present; here our effective number of variables is $s_1 \leq s \leq s_1^2$. The second condition restricts the size of s_1 more stringently but is nevertheless weaker than equivalent conditions in Hao and Zhang [2014].

A4 is a minimal signal strength condition. The term involving σ is the usual bound on the signal strength required in results on variable selection consistency with the Lasso when there are s_1^2 non-zero variables. Due to the presences of interactions, the terms not involving σ place additional restrictions on the sizes of non-zero components of $\boldsymbol{\beta}^*$ even when $\sigma = 0$. A5 ensures that the model is not too heavily misspecified in the initial stages of the algorithm, where we are regressing on only main effects.

Theorem 1. *Assuming A1–A5, the probability that there exists a k^* such that $C^* \supseteq C_{k^*} \supseteq S^*$ tends to 1 as $n \to \infty$.*

6.2 Fixed design

The result for a random normal design above is based on a corresponding result for fixed design which we present here. In order for Backtracking not to add any interactions involving noise variables, to begin with, one pair of interacting signal variables must enter the solution path before any noise variables. Other interacting signal variables need only become active after the interaction between this first pair has become active. Thus we need that there is some ordering of the interacting variables where each variable only requires interactions between those variables earlier in the order to be present before it can become active. Variables early on in the order must have the ability to be selected when there is serious model misspecification as few interaction terms will be available for selection. Variables later in the order only need to have the ability to be selected when the model is approximately correct.

Note that a signal variable having a coefficient large in absolute value does not necessarily ensure that it becomes active before any noise variable. Indeed, in our example in Section 2, variable 5 did not enter the solution path at all when only main effects were present, but had the largest coefficient. Write $\mathbf{f}^*$ for $\mathbf{X}_{S^*}\boldsymbol{\beta}_{S^*}$, and for a set S such that $\mathbf{X}_S$ has full column rank, define

$$\boldsymbol{\beta}^S := (\mathbf{X}_S^T \mathbf{X}_S)^{-1} \mathbf{X}_S^T \mathbf{f}^*.$$

Intuitively what should matter are the sizes of the appropriate coefficients of $\boldsymbol{\beta}^S$ for suitable choices of S. In the next section, we give a sufficient condition based on $\boldsymbol{\beta}^S$ for a variable $v \in S$ to enter the solution path before any variable outside S.

6.2.1 The entry condition

Let $\mathbf{P}^S = \mathbf{X}_S(\mathbf{X}_S^T \mathbf{X}_S)^{-1}\mathbf{X}_S^T$ denote orthogonal projection on to the space spanned by the columns of $\mathbf{X}_S$. Further, for any two candidate sets S, M that are sets of subsets of $\{1, \ldots, p\}$, define

$$\hat{\boldsymbol{\Sigma}}_{S,M} = \tfrac{1}{n} \mathbf{X}_S^T \mathbf{X}_M.$$

Now given a set of candidates, C, let $v \in S \subset C$ and write $M = C \setminus S$. For $\eta > 0$, we shall say that the $\text{Ent}(v, S, C; \eta)$ condition holds if, $\mathbf{X}_S$ has full column rank, and the following holds,

$$\sup_{\boldsymbol{\tau}_S \in \mathbb{R}^{|S|}: \|\boldsymbol{\tau}_S\|_\infty \leq 1} \|\hat{\boldsymbol{\Sigma}}_{M,S} \hat{\boldsymbol{\Sigma}}_{S,S}^{-1} \boldsymbol{\tau}_S\|_\infty < 1, \tag{6.2}$$

$$|\beta_v^S| > \max_{u \in M} \left\{ \frac{\frac{1}{n} \left| \mathbf{X}_u^T (\mathbf{I} - \mathbf{P}^S) \mathbf{f}^* \right| + 2\eta}{1 - \|\hat{\boldsymbol{\Sigma}}_{S,S}^{-1} \hat{\boldsymbol{\Sigma}}_{S,\{u\}}\|_1} + \eta \right\} \|(\hat{\boldsymbol{\Sigma}}_{S,S}^{-1})_v\|_1. \tag{6.3}$$

In Lemma 4 given in the appendix, we show that this condition is sufficient for variable v to enter the active set before any variable in M, when the set of candidates is C and $\|\mathbf{X}_C^T \boldsymbol{\varepsilon}\|_\infty \leq \eta$. In addition, we show that v will remain in the active set at least until some variable from M enters the active set.

The second part of the entry condition (6.3) asserts that coefficient v of the regression of $\mathbf{f}^*$ on $\mathbf{X}_S$ must exceed a certain quantity that we now examine in more detail. The $\frac{1}{n} \mathbf{X}_u^T (\mathbf{I} - \mathbf{P}^S) \mathbf{f}^*$ term is the sample covariance between $\mathbf{X}_u$, which is one of the columns of $\mathbf{X}_M$, and the residual from regressing $\mathbf{f}^*$ on $\mathbf{X}_S$. Note that the more of S^* that S contains, the closer this will be to 0.

To understand the $\|(\hat{\boldsymbol{\Sigma}}_{S,S}^{-1})_v\|_1$ term, without loss of generality take v as $\{1\}$ and write $\mathbf{b} = \hat{\boldsymbol{\Sigma}}_{S\setminus\{v\},\{v\}}$ and $\mathbf{D} = \hat{\boldsymbol{\Sigma}}_{S\setminus\{v\},S\setminus\{v\}}$. For any square matrix $\hat{\boldsymbol{\Sigma}}$, let $c_{\min}(\hat{\boldsymbol{\Sigma}})$ denote its minimal eigenvalue. Using the formula for the inverse of a block matrix and writing s for $|S|$, we have

$$\begin{aligned} \|(\hat{\boldsymbol{\Sigma}}_{S,S}^{-1})_v\|_1 &= \left\| \begin{pmatrix} 1 + \mathbf{b}^T (\mathbf{D} - \mathbf{b}\mathbf{b}^T)^{-1} \mathbf{b} \\ -(\mathbf{D} - \mathbf{b}\mathbf{b}^T)^{-1} \mathbf{b} \end{pmatrix} \right\|_1 \\ &\leq 1 + \frac{\|\mathbf{b}\|_2^2 + \sqrt{s-1}\|\mathbf{b}\|_2}{c_{\min}(\hat{\boldsymbol{\Sigma}}_{S,S})}. \end{aligned}$$

In the final line we have used the Cauchy–Schwarz inequality and the fact that if $\mathbf{w}^*$ is a unit eigenvector of $\mathbf{D} - \mathbf{b}\mathbf{b}^T$ with minimal eigenvalue, then

$$c_{\min}(\mathbf{D} - \mathbf{b}\mathbf{b}^T) = \left\| \hat{\boldsymbol{\Sigma}}_{S,S} \begin{pmatrix} -\mathbf{b}^T \mathbf{w}^* \\ \mathbf{w}^* \end{pmatrix} \right\|_2 \geq c_{\min}(\hat{\boldsymbol{\Sigma}}_{S,S}) \sqrt{1 + |\mathbf{b}^T \mathbf{w}^*|^2} \geq c_{\min}(\hat{\boldsymbol{\Sigma}}_{S,S}).$$

Thus when variable v is not too correlated with the other variables in S, and so $\|\mathbf{b}\|_2$ is small, $\|(\hat{\boldsymbol{\Sigma}}_{S,S}^{-1})_v\|_1$ will not be too large. Even when this is not the case, we still have the bound

$$\|(\hat{\boldsymbol{\Sigma}}_{S,S}^{-1})_v\|_1 \leq \frac{\sqrt{|S|}}{c_{\min}(\hat{\boldsymbol{\Sigma}}_{S,S})}.$$

Turning now to the denominator, $\|\hat{\boldsymbol{\Sigma}}_{S,S}^{-1} \hat{\boldsymbol{\Sigma}}_{S,\{u\}}\|_1$ is the ℓ_1-norm of the coefficient of regression of $\mathbf{X}_u$ on $\mathbf{X}_S$, and the maximum of this quantity over $u \in M$ gives the left-hand side of (6.2). Thus when u is highly correlated with many of the variables in S, $\|\hat{\boldsymbol{\Sigma}}_{S,S}^{-1} \hat{\boldsymbol{\Sigma}}_{S,\{u\}}\|_1$ will be large. On the other hand, in this case one would expect $\|(\mathbf{I} - \mathbf{P}^S)\mathbf{X}_u\|_2$ to be small, and so to some extent the numerator and denominator compensate for each other.

6.2.2 Statement of results

Without loss of generality assume $I^* = \{1, \ldots, |I^*|\}$. Also let $\mathcal{J} = \{\mathcal{I}(A) : A \subseteq S_1^*\}$. Our formal assumption corresponding to the discussion at the beginning of Section 6 is the following.

The entry order condition. There is some ordering of the variables in I^*, which without loss of generality we take to simply be $1, \ldots, |I^*|$, such that for each $j \in I^*$, we have,

$$\begin{gathered}\text{For all } A \in \mathcal{J} \text{ with } \mathcal{I}(1, \ldots, j-1) \subseteq A \subseteq \mathcal{I}(S_1^*) \\ \text{Ent}(j, S_1^* \cup B, C_1 \cup A; \eta) \text{ holds for some } A \cap S_2^* \subseteq B \subseteq A.\end{gathered}$$

Here

$$\eta = \eta(t; n, p, s_1, \sigma) = \sigma \sqrt{\frac{t^2 + 2\log(p + s_1^2)}{n}}.$$

First we discuss the implications for variable 1. The condition ensures that whenever the candidate set is enlarged from C_1 to also include any set of interactions built from S_1^*, variable 1 enters the active set before any variable outside $\mathcal{I}(S_1^*)$, and moreover, it remains in the active set at least until a variable outside $\mathcal{I}(S_1^*)$ enters.

For $j > 2$, we see that the enlarged candidate sets for which we require the entry conditions to hold, are fewer in number. Variable $|I^*|$ only requires the entry condition to hold for candidate sets that at least include $\mathcal{I}(1, \ldots, |I^*|-1)$ and thus include almost all of S^*. What this means is that we require some 'strong' interacting variables, for which when $\mathbf{f}^*$ is regressed onto a variety of sets of variables containing them (some of which contain only a few of the true interaction variables), always have large coefficients. Given the existence of such strong variables, other interacting variables need only have large coefficients when $\mathbf{f}^*$ is regressed onto sets containing them that also include many true interaction terms. Note that the equivalent result for the success of the strategy that simply adds interactions between selected main effects would essentially require all main effect involved in interactions to satisfy the conditions imposed on the variables 1 and 2 here. Going back to the example in Section 2, variable 5 has $|\beta_5^S| \approx 0$ for all $S \subseteq \{1, \ldots, 6\}$, but $|\beta_5^S| > 0$ once $\{1, 2\} \in S$ or $\{3, 4\} \in S$.

Theorem 2. *Assume the entry order condition holds. With probability at least* $1 - \exp(-t^2/2)$*, there exists a* k^* *such that* $C^* \supseteq C_{k^*} \supseteq S^*$.

The following corollary establishes variable selection consistency under some additional conditions.

Corollary 3. *Assume the entry order condition holds. Writing* $N = C^* \setminus S^*$*, further assume*

$$\|\hat{\boldsymbol{\Sigma}}_{N,S^*} \hat{\boldsymbol{\Sigma}}_{S^*,S^*}^{-1} \text{sgn}(\boldsymbol{\beta}_{S^*}^*)\|_\infty < 1;$$

and that for all $v \in S^*$*,*

$$|\beta_v^*| > \frac{\eta \left| \text{sgn}(\boldsymbol{\beta}_{S^*}^*)^T (\hat{\boldsymbol{\Sigma}}_{S^*,S^*}^{-1})_v \right|}{1 - \|\hat{\boldsymbol{\Sigma}}_{N,S^*} \hat{\boldsymbol{\Sigma}}_{S^*,S^*}^{-1} \text{sgn}(\boldsymbol{\beta}_{S^*}^*)\|_\infty} + \xi,$$

where

$$\xi = \xi(t; n, s, \sigma, c_{\min}(\hat{\boldsymbol{\Sigma}}_{S^*,S^*})) = \sigma \sqrt{\frac{t^2 + 2\log(s)}{n c_{\min}(\hat{\boldsymbol{\Sigma}}_{S^*,S^*})}}.$$

Then with probability at least $1 - 3\exp(-t^2/2)$*, there exist* k^* *and* λ^* *such that*

$$\mathcal{A}(\tilde{\boldsymbol{\beta}}(\lambda^*, C_{k^*})) = S^*.$$

Note that if we were to simply apply the Lasso to the set of candidates $C^{\text{all}} := C_1 \cup \mathcal{I}(C_1)$ (i.e. all possible main effects and their first-order interactions), we would require an irrepresentable condition of the form

$$\|\hat{\boldsymbol{\Sigma}}_{N^{\text{all}},S^*}\hat{\boldsymbol{\Sigma}}_{S^*,S^*}^{-1}\text{sgn}(\boldsymbol{\beta}^*_{S^*})\|_\infty < 1,$$

where $N^{\text{all}} = C^{\text{all}} \setminus S^*$. Thus we would need $O(p^2)$ inequalities to hold, rather than our $O(p)$. Of course, we had to introduce many additional assumptions to reach this stage and no set of assumptions is uniformly stronger or weaker than the other. However, our proposed method is computationally feasible.

7 Discussion

While several methods now exist for fitting interactions in moderate-dimensional situations where p is in the order of hundreds, the problem of fitting interactions when the data is of truly high dimension has received less attention.

Typically, the search for interactions must be restricted by first fitting a model using only main effects, and then including interactions between those selected main effects, as well as the original main effects, as candidates in a final fit. This approach has the drawbacks that important main effects may not be selected in the initial stage as they require certain interactions to be present in order for them to be useful for prediction. In addition, the initial model may contain too many main effects when, without the relevant interactions, the model selection procedure cannot find a good sparse approximation to the true model.

The Backtracking method proposed in this paper allows interactions to be added in a more natural gradual fashion, so there is a better chance of having a model which contains the right interactions. The method is computationally efficient, and our numerical results demonstrate its effectiveness for both variable selection and prediction.

From a theoretical point of view we have shown that when used with the Lasso, rather than requiring all main effects involved in interactions to be highly correlated with the signal, Backtracking only needs there to exist some ordering of these variables where those early on in the order are important for predicting the response by themselves. Variables later in the order only need to be helpful for predicting the response when interactions between variables early on in the order are present.

Though in this paper, we have largely focussed on Backtracking used with the Lasso, the method is very general and can be used with many procedures that involve sparsity-inducing penalty functions. These methods tend to be some of the most useful for dealing with high-dimensional data, as they can produce stable, interpretable models. Combined with Backtracking, the methods become much more flexible, and it would be very interesting to explore to what extent using non-linear base procedures could yield interpretable models with predictive power comparable to black-box procedures such as Random Forests [Breiman, 2001]. In addition, we believe integrating Backtracking with some of the penalty-based methods for fitting interactions to moderate-dimensional data, will prove to be a fruitful direction for future research.

Acknowledgements

I am very grateful to Richard Samworth, for many helpful comments and suggestions.

8 Appendix

We first prove Theorem 2 and Corollory 3. The results obtained here are used for the proof of Theorem 1 which follows.

8.1 Proofs of Theorem 2 and Corollory 3

In this subsection we use many ideas from Section B of Wainwright [2009] and Section 6 of Bühlmann and van de Geer [2011b].

Lemma 4. *Let $S \subseteq C$ be such that X_S has full column rank and let $M = C \setminus S$. On the event*

$$\Omega_{C,\eta} := \{\tfrac{1}{n}\|\mathbf{X}_C^T \boldsymbol{\varepsilon}\|_\infty \leq \eta\},$$

the following hold:

(i) If

$$\lambda > \max_{u \in M} \left\{ \frac{\frac{1}{n}|\mathbf{X}_u^T(\mathbf{I} - \mathbf{P}^S)\mathbf{f}^*| + 2\eta}{1 - \|\hat{\boldsymbol{\Sigma}}_{S,S}^{-1}\hat{\boldsymbol{\Sigma}}_{S,\{u\}}\|_1} \right\}, \tag{8.1}$$

then the Lasso solution is unique and $\hat{\boldsymbol{\beta}}_M(\lambda, C) = \mathbf{0}$.

(ii) If λ is such that for some Lasso solution $\hat{\boldsymbol{\beta}}_M(\lambda, C) = \mathbf{0}$, and for $v \in S$,

$$|\beta_v^S| > \|(\hat{\boldsymbol{\Sigma}}_{S,S}^{-1})_v\|_1(\lambda + \eta),$$

then for all Lasso solutions, $\hat{\beta}_v(\lambda, C) \neq 0$.

(iii) Let

$$\lambda^{\text{ent}} = \sup\{\lambda : \lambda \geq 0 \text{ and for some Lasso solution } \hat{\boldsymbol{\beta}}_M(\lambda, C) \neq \mathbf{0}\},$$

where we take $\sup \emptyset = 0$. If for $v \in S$,

$$|\beta_v^S| > \max_{u \in M} \left\{ \frac{\frac{1}{n}|\mathbf{X}_u^T(\mathbf{I} - \mathbf{P}^S)\mathbf{f}^*| + 2\eta}{1 - \|\hat{\boldsymbol{\Sigma}}_{S,S}^{-1}\hat{\boldsymbol{\Sigma}}_{S,\{u\}}\|_1} + \eta \right\} \|(\hat{\boldsymbol{\Sigma}}_{S,S}^{-1})_v\|_1,$$

there exists a $\lambda > \lambda^{\text{ent}}$ such that the solution $\hat{\boldsymbol{\beta}}(\lambda, C)$ is unique, and for all $\lambda' \in (\lambda^{\text{ent}}, \lambda]$ and all Lasso solutions $\hat{\boldsymbol{\beta}}(\lambda', C)$, we have $\hat{\beta}_v(\lambda', C) \neq 0$.

Proof. We begin by proving (i). Suppressing the dependence of $\hat{\boldsymbol{\beta}}$ on λ and C, we can write the KKT conditions ((3.1), (3.2)) as

$$\frac{1}{n}\mathbf{X}_C^T(\mathbf{Y} - \mathbf{X}_C\hat{\boldsymbol{\beta}}) = \lambda\hat{\boldsymbol{\tau}},$$

where $\hat{\boldsymbol{\tau}}$ is an element of the subdifferential $\partial\|\hat{\boldsymbol{\beta}}\|_1$ and thus satisfies

$$\|\hat{\boldsymbol{\tau}}\|_\infty \leq 1, \tag{8.2}$$

$$\hat{\beta}_v \neq 0 \Rightarrow \hat{\tau}_v = \text{sgn}(\hat{\beta}_v). \tag{8.3}$$

By decomposing $\mathbf{Y}$ as $\mathbf{P}^S\mathbf{f}^* + (\mathbf{I}-\mathbf{P}^S)\mathbf{f}^* + \varepsilon$, $\mathbf{X}_C$ as $(\mathbf{X}_S\,\mathbf{X}_M)$, and noting that $\mathbf{X}_S^T(\mathbf{I}-\mathbf{P}^S)=\mathbf{0}$, we can rewrite the KKT conditions in the following way:

$$\tfrac{1}{n}\mathbf{X}_S^T(\mathbf{P}^S\mathbf{f}^* - \mathbf{X}_S\hat{\boldsymbol{\beta}}_S) + \tfrac{1}{n}\mathbf{X}_S^T\boldsymbol{\varepsilon} - \hat{\boldsymbol{\Sigma}}_{S,M}\hat{\boldsymbol{\beta}}_{J^*} = \lambda\hat{\boldsymbol{\tau}}_S, \tag{8.4}$$

$$\tfrac{1}{n}\mathbf{X}_M^T(\mathbf{P}^S\mathbf{f}^* - \mathbf{X}_S\hat{\boldsymbol{\beta}}_S) + \tfrac{1}{n}\mathbf{X}_M^T\{(\mathbf{I}-\mathbf{P}^S)\mathbf{f}^* + \boldsymbol{\varepsilon}\} - \hat{\boldsymbol{\Sigma}}_{M,M}\hat{\boldsymbol{\beta}}_M = \lambda\hat{\boldsymbol{\tau}}_M. \tag{8.5}$$

Now let $\breve{\boldsymbol{\beta}}_S$ be a solution to the restricted Lasso problem,

$$(\hat{\mu}, \breve{\boldsymbol{\beta}}_S) = \operatorname*{arg\,min}_{\mu,\boldsymbol{\beta}_S} \left\{ \tfrac{1}{2n}\|\mathbf{Y} - \mu\mathbf{1} - \mathbf{X}_S\boldsymbol{\beta}_S\|^2 + \lambda\|\boldsymbol{\beta}_S\|_1 \right\}.$$

The KKT conditions give that $\breve{\boldsymbol{\beta}}_S$ satisfies

$$\frac{1}{n}\mathbf{X}_S^T(\mathbf{Y} - \mathbf{X}_S\breve{\boldsymbol{\beta}}_S) = \lambda\breve{\boldsymbol{\tau}}_S, \tag{8.6}$$

where $\breve{\boldsymbol{\tau}}_S \in \partial\|\breve{\boldsymbol{\beta}}_S\|_1$. We now claim that

$$(\hat{\boldsymbol{\beta}}_S, \hat{\boldsymbol{\beta}}_M) = (\breve{\boldsymbol{\beta}}_S, \mathbf{0}) \tag{8.7}$$

$$(\hat{\boldsymbol{\tau}}_S, \hat{\boldsymbol{\tau}}_M) = \left(\breve{\boldsymbol{\tau}}_S,\, \hat{\boldsymbol{\Sigma}}_{M,S}\hat{\boldsymbol{\Sigma}}_{S,S}^{-1}(\breve{\boldsymbol{\tau}}_S - \tfrac{1}{n}\lambda^{-1}\mathbf{X}_S^T\boldsymbol{\varepsilon}) + \tfrac{1}{n}\lambda^{-1}\mathbf{X}_M^T\{(\mathbf{I}-\mathbf{P}^S)\mathbf{f}^* + \boldsymbol{\varepsilon}\}\right) \tag{8.8}$$

is the unique solution to (8.4), (8.5), (8.2) and (8.3). Indeed, as $\breve{\boldsymbol{\beta}}_S$ solves the reduced Lasso problem, we must have that (8.4) and (8.3) are satisfied. Multiplying (8.4) by $\mathbf{X}_S\hat{\boldsymbol{\Sigma}}_{S,S}^{-1}$, setting $\hat{\boldsymbol{\beta}}_M = \mathbf{0}$ and rearranging gives us that

$$\mathbf{P}^S\mathbf{f}^* - \mathbf{X}_S\hat{\boldsymbol{\beta}}_S = \mathbf{X}_S\hat{\boldsymbol{\Sigma}}_{S,S}^{-1}(\lambda\hat{\boldsymbol{\tau}}_S - \tfrac{1}{n}\mathbf{X}_S^T\boldsymbol{\varepsilon}), \tag{8.9}$$

and substituting this into (8.5) shows that our choice of $\hat{\boldsymbol{\tau}}_M$ satisfies (8.5). It remains to check that we have $\|\hat{\boldsymbol{\tau}}_M\|_\infty \leq 1$. In fact, we shall show that $\|\hat{\boldsymbol{\tau}}_M\|_\infty < 1$. Since we are on $\Omega_{C,\eta}$ and $\|\breve{\boldsymbol{\tau}}_S\|_\infty \leq 1$, for $u \in M$ we have

$$\begin{aligned}
\lambda|\hat{\boldsymbol{\tau}}_u| &\leq \|\hat{\boldsymbol{\Sigma}}_{S,S}^{-1}\hat{\boldsymbol{\Sigma}}_{S,\{u\}}\|_1 \left(\lambda\|\breve{\boldsymbol{\tau}}_S\|_\infty + \|\tfrac{1}{n}\mathbf{X}_S^T\boldsymbol{\varepsilon}\|_\infty\right) + \tfrac{1}{n}\left|\mathbf{X}_u^T(\mathbf{I}-\mathbf{P}^S)\mathbf{f}^*\right| + \tfrac{1}{n}\left|\mathbf{X}_u^T\boldsymbol{\varepsilon}\right| \\
&< \lambda\|\hat{\boldsymbol{\Sigma}}_{S,S}^{-1}\hat{\boldsymbol{\Sigma}}_{S,\{u\}}\|_1 + \tfrac{1}{n}\left|\mathbf{X}_u^T(\mathbf{I}-\mathbf{P}^S)\mathbf{f}^*\right| + 2\eta \\
&< \lambda,
\end{aligned}$$

where the final inequality follows from (8.1). We have shown that there exists a solution, $\hat{\boldsymbol{\beta}}$, to the Lasso optimisation problem with $\hat{\boldsymbol{\beta}}_M = 0$. The uniqueness of this solution follows from noting that $\|\hat{\boldsymbol{\tau}}_M\|_\infty < 1$, $\mathbf{X}_S$ has full column rank and appealing to Lemma 1 of Wainwright [2009].

For (ii), note that from (8.4), provided $\hat{\boldsymbol{\beta}}_M = 0$, we have that

$$\hat{\boldsymbol{\beta}}_S = \boldsymbol{\beta}^S - \hat{\boldsymbol{\Sigma}}_{S,S}^{-1}(\lambda\hat{\boldsymbol{\tau}}_S - \tfrac{1}{n}\mathbf{X}_S^T\boldsymbol{\varepsilon}).$$

But by assumption

$$|\beta_v^S| > \|(\hat{\boldsymbol{\Sigma}}_{S,S}^{-1})_v\|_1(\lambda+\eta) \geq \left|(\hat{\boldsymbol{\Sigma}}_{S,S}^{-1})_v^T(\lambda\hat{\boldsymbol{\tau}}_S - \tfrac{1}{n}\mathbf{X}_S^T\boldsymbol{\varepsilon})\right|,$$

whence $\hat{\beta}_v \neq 0$.

(iii) follows easily from (i) and (ii). □

8.1.1 Proof of Theorem 2

In all that follows, we work on the event $\Omega_{C^*,\eta}$ defined in Lemma 4. Using standard bounds for the tails of Gaussian random variables and the union bound, it is easy to show that $\mathbb{P}(\Omega_1 \cap \Omega_{C^*,\eta}) \geq 1 - \exp(-t^2/2)$. Let $N = \{1, \ldots, p\} \setminus S_1^*$.

Let $\tilde{T}$ be the number of steps taken by the algorithm: this would typically be T, but may be smaller if a perfect fit is reached or if $p < T$ for example. Let C_k be the largest member of $\{C_1, \ldots, C_{\tilde{T}}\}$ satisfying $C_k \subseteq C^*$. Such a C_k exists since $C_1 \subseteq C^*$.

Now suppose for a contradiction that $C_k \not\supseteq S^*$. Let j be such that

$$\mathcal{I}(1, \ldots, j-1) \subseteq C_k,$$

with j maximal. Since $\mathcal{I}(1) = \emptyset$, such a j exists. Let $A = C_k \setminus C_1$. Note that $A \in \mathcal{J}$ and

$$\mathcal{I}(1, \ldots, j-1) \subseteq A \subseteq C^* \setminus C_1 = \mathcal{I}(S_1^*).$$

By the entry order condition, we know that j will enter the active set before any variable in N, and before a perfect fit is reached. Thus $k+1 \leq \tilde{T}$ and C_{k+1} contains only additional interactions not involving any variables from N, so $C_{k+1} \subseteq C^*$. □

8.1.2 Proof of Corollary 3

Let $\Omega_{C^*,\eta}$ be defined as in Lemma 4. Also define the events

$$\Omega_1 = \{\tfrac{1}{n}\|\mathbf{X}_N^T(\mathbf{I} - \mathbf{P}^{S^*})\boldsymbol{\varepsilon}\|_\infty \leq \eta\},$$
$$\Omega_2 = \{\tfrac{1}{n}\|\hat{\boldsymbol{\Sigma}}_{S^*,S^*}^{-1}\mathbf{X}_{S^*}^T\boldsymbol{\varepsilon}\|_\infty \leq \xi\}$$

In all that follows, we work on the event $\Omega_1 \cap \Omega_2 \cap \Omega_{C^*,\eta}$. As $\mathbf{I} - \mathbf{P}^{S^*}$ is a projection,

$$\mathbb{P}(\tfrac{1}{n}|\mathbf{X}_v{}^T(\mathbf{I} - \mathbf{P}^{S^*})\boldsymbol{\varepsilon}| \leq \eta) \geq \mathbb{P}(\tfrac{1}{n}|\mathbf{X}_v{}^T\boldsymbol{\varepsilon}| \leq \eta).$$

Further, $\frac{1}{n}\hat{\boldsymbol{\Sigma}}_{S^*,S^*}^{-1}\mathbf{X}_{S^*}^T\boldsymbol{\varepsilon} \sim N_{|S^*|}(\mathbf{0}, \frac{1}{n}\sigma^2\hat{\boldsymbol{\Sigma}}_{S^*,S^*}^{-1})$. Thus

$$\mathbb{P}(\Omega_3) \geq |S^*|\mathbb{P}(|Z| \leq \xi)$$

where $Z \sim N(0, \sigma^2/(nc_{\min}(\hat{\boldsymbol{\Sigma}}_{S^*,S^*})))$. Note that

$$\mathbb{P}(\Omega_1 \cap \Omega_2 \cap \Omega_{C^*,\eta}) \geq 1 - \mathbb{P}(\Omega_{C^*,\eta}^c) - \mathbb{P}(\Omega_1^c) - \mathbb{P}(\Omega_2^c).$$

Using this, it is straightforward to show that $\mathbb{P}(\Omega_1 \cap \Omega_2 \cap \Omega_{C^*,\eta}) \geq 1 - 3\exp(-t^2/2)$.

Since we are on $\Omega_{C^*,\eta}$, we can assume the existence of a k^* from Theorem 2. We now follow the proof of Lemma 4 taking $S = S^*$ and $M = C_{k^*} \setminus S^* \subseteq N$. The KKT conditions become

$$\hat{\boldsymbol{\Sigma}}_{S^*,S^*}(\boldsymbol{\beta}_{S^*}^* - \hat{\boldsymbol{\beta}}_{S^*}) + \tfrac{1}{n}\mathbf{X}_{S^*}^T\boldsymbol{\varepsilon} - \hat{\boldsymbol{\Sigma}}_{S^*,M}\hat{\boldsymbol{\beta}}_M = \lambda\hat{\boldsymbol{\tau}}_{S^*}, \tag{8.10}$$
$$\hat{\boldsymbol{\Sigma}}_{M,S^*}(\boldsymbol{\beta}_{S^*}^* - \hat{\boldsymbol{\beta}}_{S^*}) + \tfrac{1}{n}\mathbf{X}_M^T\boldsymbol{\varepsilon} - \hat{\boldsymbol{\Sigma}}_{M,M}\hat{\boldsymbol{\beta}}_M = \lambda\hat{\boldsymbol{\tau}}_M, \tag{8.11}$$

with $\hat{\boldsymbol{\tau}}$ also satisfying (8.2) and (8.3) as before. Now let λ be such that

$$\frac{\eta}{1 - \|\hat{\boldsymbol{\Sigma}}_{M,S^*}\hat{\boldsymbol{\Sigma}}_{S^*,S^*}^{-1}\mathrm{sgn}(\boldsymbol{\beta}_{S^*}^*)\|_\infty} < \lambda < \min_{v \in S^*}\left\{\left|\mathrm{sgn}(\boldsymbol{\beta}_{S^*}^*)^T(\hat{\boldsymbol{\Sigma}}_{S^*,S^*}^{-1})_v\right|^{-1}(|\beta_v^*| - \xi)\right\}.$$

It is straightforward to check that

$$(\hat{\boldsymbol{\beta}}_{S^*}, \hat{\boldsymbol{\beta}}_M) = (\boldsymbol{\beta}^*_{S^*} - \lambda \hat{\boldsymbol{\Sigma}}^{-1}_{S^*,S^*} \mathrm{sgn}(\boldsymbol{\beta}^*_{S^*}) + \tfrac{1}{n} \hat{\boldsymbol{\Sigma}}^{-1}_{S^*,S^*} \mathbf{X}^T_{S^*} \boldsymbol{\varepsilon},\, \mathbf{0})$$
$$(\hat{\boldsymbol{\tau}}_{S^*}, \hat{\boldsymbol{\tau}}_M) = \Big(\mathrm{sgn}(\boldsymbol{\beta}^*_{S^*}),\, \hat{\boldsymbol{\Sigma}}_{M,S^*} \hat{\boldsymbol{\Sigma}}^{-1}_{S^*,S^*} \mathrm{sgn}(\boldsymbol{\beta}^*_{S^*}) + \tfrac{1}{n} \lambda^{-1} \mathbf{X}^T_M (\mathbf{I} - \mathbf{P}^{S^*}) \boldsymbol{\varepsilon} \Big)$$

is the unique solution to (8.10), (8.11), (8.2) and (8.3). □

8.2 Proof of Theorem 1

In the following, we make use of notation defined in Section 6.2. In addition, for convenience we write $S = S^*_1$, $M = S \cup J^*$. Also, we will write main effects variables $\{j\}$ as simply j. First we collect together various results concerning $\hat{\boldsymbol{\Sigma}}_{C^*,C^*}$.

Lemma 5. *Consider the setup of Theorem 1. Let $\mathbb{E}_n$ and Var_n denote empirical expectation and variance with respect to $\mathbf{Z}$ so that, for example $\mathbb{E}_n z_j = \sum_{i=1}^n Z_{ij}/n$.*

(i) Let $\mathbf{D}$ be the diagonal matrix indexed by C^* used to scale transformations of $\mathbf{Z}$ in order to create $\mathbf{X}_{C^*}$ i.e. with entries such that $D^2_{jj} = \mathrm{Var}_n(z_j)$ and $D^2_{vv} = \mathrm{Var}_n(z_j - \mathbb{E}_n z_j)(z_k - \mathbb{E}_n z_k)$ when $v = \{j,k\}$. Then

$$\max_{j \in C_1} |D^2_{jj} - 1| = O_P(\sqrt{\log(p)/n}) \tag{8.12}$$
$$\max_{\{j,k\} \in M} |D^2_{\{j,k\},\{j,k\}} - 1 - \Sigma^2_{jk}| = O_P(\sqrt{\log(s_1)} n^{-1/4}) \tag{8.13}$$

(ii)

$$\tfrac{1}{n} \|\mathbf{X}^T_{J^*} \mathbf{X}_S\|_\infty = O_P(\sqrt{\log(s_1)} n^{-1/3}) \tag{8.14}$$
$$c_{min}(\hat{\boldsymbol{\Sigma}}_{S,S}) \geq c_* - s_1 O_P(\sqrt{\log(s_1)/n}) \tag{8.15}$$
$$c_{min}(\hat{\boldsymbol{\Sigma}}_{M,M}) \geq c^2_* + s^2_1 O_P(\sqrt{\log(s_1)} n^{-1/4}) \tag{8.16}$$
$$c_{max}(\hat{\boldsymbol{\Sigma}}_{J^*,J^*}) \leq 2c^{*2} + s^2_1 O_P(\sqrt{\log(s_1)} n^{-1/4}). \tag{8.17}$$

Proof. We use bounds on the tails of products of normal random variables from Hao and Zhang [2014] (equation B.9). We have

$$\begin{aligned} \max_{j,k} |\mathrm{Cov}_n(z_j, z_k) - \Sigma_{jk}| &= \max_{j,k} |\mathbb{E}_n(z_j z_k) - \mathbb{E}_n z_j \mathbb{E}_n z_k - \Sigma_{jk}| \\ &= O_P(\sqrt{\log(p)/n}). \end{aligned}$$

Also,

$$\begin{aligned} &\max_{j,k,l,m \in S} |\mathrm{Cov}_n\big((z_j - \mathbb{E}_n z_j)(z_k - \mathbb{E}_n z_k),\, (z_l - \mathbb{E}_n z_l)(z_m - \mathbb{E}_n z_m)\big) - \Sigma_{jl}\Sigma_{km} - \Sigma_{jm}\Sigma_{kl}| \\ &= \max_{j,k,l,m \in S} |\mathbb{E}_n(z_j z_k z_l z_m) - \mathbb{E}_n(z_j z_k)\mathbb{E}_n(z_l z_k) - \Sigma_{jl}\Sigma_{km} - \Sigma_{jm}\Sigma_{kl}| + O_P(\sqrt{\log(s_1)/n}) \\ &= O_P(\sqrt{\log(s_1)} n^{-1/4}). \end{aligned}$$

Now we consider (ii). We have

$$\begin{aligned}\tfrac{1}{n}\|\mathbf{X}_{J^*}^T\mathbf{X}_S\|_\infty &\le \max_{v\in J^*} D_{vv}^{-1}\max_{k\in S} D_{kk}^{-1}\max_{j,k,l\in S}|\mathrm{Cov}_n\big((z_j-\mathbb{E}_n z_j)(z_k-\mathbb{E}_n z_k),\, z_l\big)| \\ &\le O_P(\sqrt{\log(s_1)}n^{-1/3}),\end{aligned}$$

the rate being driven by the size of $\mathbb{E}_n(z_j z_k z_l)$. Also

$$\begin{aligned}c_{\min}(\hat{\mathbf{\Sigma}}_{S,S}) &= \min_{\boldsymbol{\tau}\in\mathbb{R}^{s_1}:\|\boldsymbol{\tau}\|_2=1}\boldsymbol{\tau}\{\mathbf{\Sigma}_{S,S}-(\mathbf{\Sigma}_{S,S}-\hat{\mathbf{\Sigma}}_{S,S})\}\boldsymbol{\tau} \\ &\ge c_{\min}(\mathbf{\Sigma}_{S,S}) - \max_{\boldsymbol{\tau}\in\mathbb{R}^{s_1}:\|\boldsymbol{\tau}\|_2=1}\|\boldsymbol{\tau}\|_1^2\|\mathbf{\Sigma}_{S,S}-\hat{\mathbf{\Sigma}}_{S,S}\|_\infty \\ &= c_* - s_1 O_P(\sqrt{\log(s_1)/n}).\end{aligned}$$

Now let $\tilde{\mathbf{\Sigma}}$ be a matrix with entries indexed by M with

$$\tilde{\Sigma}_{uv} = \Sigma_{jl}\Sigma_{km} + \Sigma_{jm}\Sigma_{kl}$$

when $u=\{j,k\}$ and $v=\{l,m\}$. Lemma A.4 of Hao and Zhang [2014] shows that $c_{\min}(\tilde{\mathbf{\Sigma}}) \ge 2c_{\min}(\mathbf{\Sigma}_{S,S})^2$ and $c_{\max}(\tilde{\mathbf{\Sigma}}) \le 2c_{\max}(\mathbf{\Sigma}_{S,S})^2$. Thus we have

$$\begin{aligned}c_{\min}(\hat{\mathbf{\Sigma}}_{M,M}) &= \min_{\boldsymbol{\tau}\in\mathbb{R}^{|M|}:\|\mathbf{D}_{M,M}\boldsymbol{\tau}\|_2=1}\boldsymbol{\tau}\mathbf{D}_{M,M}\hat{\mathbf{\Sigma}}_{M,M}\mathbf{D}_{M,M}\boldsymbol{\tau} \\ &\ge \|\mathbf{D}_{M,M}\|_\infty^{-1}c_{\min}(\mathbf{D}_{M,M}\hat{\mathbf{\Sigma}}_{M,M}\mathbf{D}_{M,M}) \\ &\ge \{1+O_P(\sqrt{\log(s_1)}n^{-1/4})\}[c_*^2 - s_1^2\{\|\tilde{\mathbf{\Sigma}}-\mathbf{D}_{M,M}\hat{\mathbf{\Sigma}}_{M,M}\mathbf{D}_{M,M})\|_\infty + O_P(\sqrt{\log(s_1)}n^{-1/3})\}] \\ &\ge c_*^2 + s_1^2 O_P(\sqrt{\log(s_1)}n^{-1/4}).\end{aligned}$$

Similarly

$$\begin{aligned}c_{\max}(\hat{\mathbf{\Sigma}}_{J^*,J^*}) &= \max_{\boldsymbol{\tau}\in\mathbb{R}^{|J^*|}:\|\mathbf{D}_{J^*,J^*}\boldsymbol{\tau}\|_2=1}\boldsymbol{\tau}\mathbf{D}_{J^*,J^*}\hat{\mathbf{\Sigma}}_{J^*,J^*}\mathbf{D}_{J^*,J^*}\boldsymbol{\tau} \\ &\le \{1-O_P(\sqrt{\log(s_1)}n^{-1/4})\}c_{\max}(\mathbf{D}_{J^*,J^*}\hat{\mathbf{\Sigma}}_{J^*,J^*}\mathbf{D}_{J^*,J^*}) \\ &\le \{1-O_P(\sqrt{\log(s_1)}n^{-1/4})\}\{2c^{*2}+s_1^2\|\tilde{\mathbf{\Sigma}}-\mathbf{D}_{J^*,J^*}\hat{\mathbf{\Sigma}}_{J^*,J^*}\mathbf{D}_{J^*,J^*})\|_\infty\} \\ &\le 2c^{*2} + s_1^2 O_P(\sqrt{\log(s_1)}n^{-1/4}).\end{aligned}$$

□

Lemma 6. *Working with the assumptions of Theorem 1, we have*

$$\max_{A\in\mathcal{J}}\|\boldsymbol{\beta}_S^{S\cup A}-\boldsymbol{\beta}_S^*\|_\infty \le O_P(s_1\sqrt{\log(s_1)}n^{-1/3}).$$

Proof. For $A\in\mathcal{J}$ let $\mathbf{\Delta}^A\in\mathbb{R}^{|S\cup A|}$ with $\mathbf{\Delta}_S^A = \boldsymbol{\beta}_S^{S\cup A}-\boldsymbol{\beta}_S^*$ and $\mathbf{\Delta}_A^A = \boldsymbol{\beta}_A^{S\cup A}$. Define $\mathbf{g}^* = \mathbf{X}_{S_2^*}\boldsymbol{\beta}_{S_2^*}^*$. Note that

$$\mathbf{f}^* = \mathbf{X}_S\boldsymbol{\beta}_S^* + \mathbf{g}^*,$$

so

$$\mathbf{\Delta}^A = (\mathbf{X}_{S\cup A}^T\mathbf{X}_{S\cup A})^{-1}\mathbf{X}_{S\cup A}\mathbf{g}^*.$$

First we bound $\|\boldsymbol{\Delta}_A^A\|_2^2$ in terms of $\|\mathbf{g}^*\|_2^2$. We have that

$$\|\mathbf{X}_{S\cup A}\boldsymbol{\Delta}^A\|_2^2 = \|\mathbf{X}_S\boldsymbol{\Delta}_S^A\|_2^2 + 2{\boldsymbol{\Delta}_S^A}^T\mathbf{X}_S^T\mathbf{X}_A\boldsymbol{\Delta}_A^A + \|\mathbf{X}_A\boldsymbol{\Delta}_A^A\|_2^2 \leq \|\mathbf{g}^*\|_2^2.$$

Thus

$$c_{\min}(\tfrac{1}{n}\mathbf{X}_S^T\mathbf{X}_S)\|\boldsymbol{\Delta}_S^A\|_\infty^2 + 2\sqrt{|A|}\|\tfrac{1}{n}\mathbf{X}_S^T\mathbf{X}_A\|_\infty\|\boldsymbol{\Delta}_A^A\|_2\|\boldsymbol{\Delta}_S^A\|_\infty + c_{\min}(\tfrac{1}{n}\mathbf{X}_A^T\mathbf{X}_A)\|\boldsymbol{\Delta}_A^A\|_2^2 - \tfrac{1}{n}\|\mathbf{g}^*\|_2^2 \leq 0.$$

Thinking of this as a quadratic in $\|\boldsymbol{\Delta}_S^A\|_\infty$ and considering the discriminant yields

$$\|\boldsymbol{\Delta}_A^A\|_2^2 \leq \frac{\frac{1}{n}c_{\min}(\frac{1}{n}\mathbf{X}_S^T\mathbf{X}_S)\|\mathbf{g}^*\|_2^2}{c_{\min}(\frac{1}{n}\mathbf{X}_S^T\mathbf{X}_S)c_{\min}(\frac{1}{n}\mathbf{X}_A^T\mathbf{X}_A) - \|\frac{1}{n}\mathbf{X}_S^T\mathbf{X}_A\|_\infty^2|A|}.$$

Thus by Lemma 5 (ii) and condition A2, $\max_{A\in\mathcal{J}}\|\boldsymbol{\Delta}_A^A\|_2 = \frac{1}{\sqrt{n}}\|\mathbf{g}^*\|_2 O_P(1)$.

But

$$\frac{1}{\sqrt{n}}\|\mathbf{g}^*\|_2 \leq \sqrt{c_{\max}(\hat{\boldsymbol{\Sigma}}_{J^*,J^*})}\|\boldsymbol{\beta}^*_{S_2^*}\|_1 = O_P(1)$$

by Lemma 5 (ii) and A5, so $\max_{A\in\mathcal{J}}\|\boldsymbol{\Delta}_A^A\|_2 = O_P(1)$.

Next observe that

$$\|\mathbf{X}_{S\cup A}\boldsymbol{\Delta}^A - \mathbf{g}^*\|_2^2 \leq \|\mathbf{X}_A\boldsymbol{\Delta}_A^A - \mathbf{g}^*\|_2^2,$$

so

$$\begin{aligned}
\|\boldsymbol{\Delta}_S^A\|_\infty^2 c_{\min}(\tfrac{1}{n}\mathbf{X}_S^T\mathbf{X}_S) &\leq \tfrac{1}{n}\|\mathbf{X}_S\boldsymbol{\Delta}_S^A\|_2^2 \\
&\leq 2\tfrac{1}{n}{\boldsymbol{\Delta}_S^A}^T\mathbf{X}_S^T(\mathbf{X}_A\boldsymbol{\Delta}_A^A - \mathbf{g}^*) \\
&\leq 2\sqrt{|A|}\|\boldsymbol{\Delta}_S^A\|_\infty\|\tfrac{1}{n}\mathbf{X}_S^T\mathbf{X}_A\|_\infty\|\boldsymbol{\Delta}_A^A\|_2 + 2\|\boldsymbol{\Delta}_S^A\|_\infty\|\tfrac{1}{n}\mathbf{X}_S^T\mathbf{g}^*\|_1.
\end{aligned}$$

Therefore

$$\|\boldsymbol{\Delta}_S^A\|_\infty \leq 2\{c_{\min}(\tfrac{1}{n}\mathbf{X}_S^T\mathbf{X}_S)\}^{-1}(\sqrt{|A|}\|\tfrac{1}{n}\mathbf{X}_S^T\mathbf{X}_A\|_\infty\|\boldsymbol{\Delta}_A^A\|_2 + \|\tfrac{1}{n}\mathbf{X}_S^T\mathbf{g}^*\|_1),$$

so

$$\max_{A\in\mathcal{J}}\|\boldsymbol{\Delta}_S^A\|_\infty \leq 2\{c_{\min}(\tfrac{1}{n}\mathbf{X}_S^T\mathbf{X}_S)\}^{-1}(\sqrt{|J^*|}\|\tfrac{1}{n}\mathbf{X}_S^T\mathbf{X}_{J^*}\|_\infty O_P(1) + \|\tfrac{1}{n}\mathbf{X}_S^T\mathbf{g}^*\|_1).$$

Now

$$\begin{aligned}
\|\tfrac{1}{n}\mathbf{X}_S^T\mathbf{g}^*\|_1 &\leq s_1\|\tfrac{1}{n}\mathbf{X}_S^T\mathbf{X}_{S_2^*}\|_\infty\|\boldsymbol{\beta}^*_{S_2^*}\|_1 \\
&\leq O_P(s_1\sqrt{\log(s_1)}n^{-1/3}).
\end{aligned}$$

Thus

$$\max_{A\in\mathcal{J}}\|\boldsymbol{\Delta}_S^A\|_\infty \leq O_P(s_1\sqrt{\log(s_1)}n^{-1/3}).$$

□

8.2.1 Proof of Theorem 1

In view of Theorem 2 and its proof, it is enough to show that with probability tending to 1, we have

$$\max_{A\in\mathcal{J}}\sup_{\boldsymbol{\tau}\in\mathbb{R}^{s_1}}\|\hat{\boldsymbol{\Sigma}}_{N,S\cup A}\hat{\boldsymbol{\Sigma}}^{-1}_{S\cup A,S\cup A}\boldsymbol{\tau}\|_\infty<1, \tag{8.18}$$

$$\min_{j\in I^*}\min_{A\in\mathcal{J}}|\beta_j^{S\cup A}|>\max_{A\in\mathcal{J}}\max_{j\in N}\left\{\frac{\frac{1}{n}|\mathbf{X}_j^T(\mathbf{I}-\mathbf{P}^{S\cup A})\mathbf{f}^*|+2\frac{1}{n}\|\mathbf{X}_{C^*}^T\boldsymbol{\varepsilon}\|_\infty}{1-\|\hat{\boldsymbol{\Sigma}}^{-1}_{S\cup A,S\cup A}\hat{\boldsymbol{\Sigma}}_{S\cup A,j}\|_1}+\frac{1}{n}\|\mathbf{X}_{C^*}^T\boldsymbol{\varepsilon}\|_\infty\right\}\frac{\sqrt{|M|}}{c_{\min}(\hat{\boldsymbol{\Sigma}}_{M,M})}. \tag{8.19}$$

First note that for $j\in N$, $\mathbf{Z}_j=\mathbf{Z}_S\boldsymbol{\Sigma}^{-1}_{S,S}\boldsymbol{\Sigma}_{S,j}+\mathbf{E}_j$ where $\mathbf{E}_j$ is independent of $\mathbf{Z}_S$ and $\mathbf{E}_j\sim N_n(\mathbf{0},(1-\boldsymbol{\Sigma}_{j,S}\boldsymbol{\Sigma}^{-1}_{S,S}\boldsymbol{\Sigma}_{S,j})\mathbf{I})$. Thus

$$\mathbf{X}_jD_{jj}=\mathbf{X}_S\mathbf{D}_{S,S}\boldsymbol{\Sigma}^{-1}_{S,S}\boldsymbol{\Sigma}_{S,j}+\mathbf{E}_j-\mathbf{1}\bar{E}_j,$$

and

$$\max_{A\in\mathcal{J}}\|(\mathbf{X}_{S\cup A}^T\mathbf{X}_{S\cup A})^{-1}\mathbf{X}_{S\cup A}^T\mathbf{X}_j\|_1\le D_{kk}^{-1}\|\mathbf{D}_{S,S}\boldsymbol{\Sigma}^{-1}_{S,S}\boldsymbol{\Sigma}_{S,j}\|_1+\max_{A\in\mathcal{J}}\|\hat{\boldsymbol{\Sigma}}^{-1}_{S\cup A,S\cup A}\tfrac{1}{n}\mathbf{X}_{S\cup A}^T\mathbf{E}_j\|_1.$$

Now the second term above is at most

$$\max_{A\in\mathcal{J}}\max_{\boldsymbol{\tau}\in\mathbb{R}^{|S\cup A|}:\|\boldsymbol{\tau}\|_2\le1}\|\hat{\boldsymbol{\Sigma}}^{-1}_{S\cup A,S\cup A}\boldsymbol{\tau}\|_1\|\tfrac{1}{n}\mathbf{X}_M^T\mathbf{E}_j\|_2.$$

But

$$\begin{aligned}\max_{A\in\mathcal{J}}\max_{\boldsymbol{\tau}\in\mathbb{R}^{|S\cup A|}:\|\boldsymbol{\tau}\|_\infty\le1}\|\hat{\boldsymbol{\Sigma}}^{-1}_{S\cup A,S\cup A}\boldsymbol{\tau}\|_1&\le\frac{\sqrt{|M|}}{c_{\min}(\hat{\boldsymbol{\Sigma}}_{M,M})}\\&\le\frac{\sqrt{|M|}}{c_*^2+s_1^2O_P(\sqrt{\log(s_1)}n^{-1/4})}.\end{aligned}$$

Also since for $v\in M$ and $j\in N$, $\mathbf{X}_v^T\mathbf{E}_j/n\sim N(0,1)$ we have

$$\max_{j\in N}\|\tfrac{1}{n}\mathbf{X}_M^T\mathbf{E}_j\|_2^2\le|M|O_P(\log(p)/n).$$

Therefore

$$\max_{A\in\mathcal{J}}\sup_{\boldsymbol{\tau}\in\mathbb{R}^{s_1}}\|\hat{\boldsymbol{\Sigma}}_{N,S\cup A}\hat{\boldsymbol{\Sigma}}^{-1}_{S\cup A,S\cup A}\boldsymbol{\tau}\|_\infty\le(1+o_P(1))\delta+\frac{s_1^2o_P(1)}{c_*^2+o_P(1)}.$$

This shows that (8.18) is satisfied with probability tending to 1.

Next

$$\max_{j\in N}\max_{A\in\mathcal{J}}\frac{1}{n}|\mathbf{X}_j^T(\mathbf{I}-\mathbf{P}^{S\cup A})\mathbf{f}^*|=\max_{j\in N}\max_{A\in\mathcal{J}}\frac{D_{jj}^{-1}}{n}|\mathbf{E}_j^T(\mathbf{I}-\mathbf{P}^{S\cup A})\mathbf{X}_A\boldsymbol{\beta}_A^*|.$$

Since $\mathbf{E}_j^T(\mathbf{I}-\mathbf{P}^{S\cup A})\mathbf{X}_A\boldsymbol{\beta}_A^*/n\sim N(0,\|(\mathbf{I}-\mathbf{P}^{S\cup A})\mathbf{X}_A\boldsymbol{\beta}_A^*\|_2^2/n^2)$ we have

$$\max_{j\in N}\max_{A\in\mathcal{J}}\frac{1}{n}|\mathbf{X}_j^T(\mathbf{I}-\mathbf{P}^{S\cup A})\mathbf{f}^*|\le\sqrt{\frac{\log(2^{s_1}p)}{n}}\frac{1}{\sqrt{n}}\|\mathbf{X}_{S_2^*}\boldsymbol{\beta}^*_{S_2^*}\|_2O_P(1).$$

By (8.17) we have

$$\frac{1}{\sqrt{n}}\|\mathbf{X}_{S_2^*}\boldsymbol{\beta}^*_{S_2^*}\|_2 \leq \{2c^{*2} + s_1^2\sqrt{\log(s_1)}n^{-1/4}O_P(1)\}\|\boldsymbol{\beta}_{S_2^*}\|_1.$$

Now using Lemma 6 we see that the difference between the LHS and RHS of (8.19) is at least

$$\min_{j\in I^*}|\beta_j^*| - O_P(s_1\sqrt{\log(s_1)}n^{-1/3}) - \left(\frac{(\sqrt{s_1+\log p}+\sigma\sqrt{\log p})/\sqrt{n}}{1-\delta+o_P(1)} + \sigma\sqrt{\frac{\log(p)}{n}}\right)s_1O_P(1).$$

Thus A4 ensures that (8.19) holds with probability tending to 1. □

References

A. Asuncion and D.J. Newman. UCI Machine Learning Repository, 2007. URL `http://archive.ics.uci.edu/ml`.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27:450–468, 2012a.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4:1–106, 2012b.

P.J. Bickel, Y. Ritov, and A.B. Tsybakov. Hierarchical selection of variables in sparse high-dimensional regression. *IMS Collections*, 6:56–69, 2010.

J. Bien, J. Taylor, and R. Tibshirani. A lasso for hierarchical interactions. *Annals of Statistics*, 41 (3):1111–1141, 2013.

B. Bollobás. *Combinatorics*. Cambridge University Press, 1986.

L. Breiman. Random Forests. *Machine Learning*, 45:5–32, 2001.

P. Bühlmann and S. van de Geer. *Statistics for high-dimensional data*. Springer, 2011a.

P. Bühlmann and S.A. van de Geer. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2011b.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least Angle Regression. *Annals of Statistics*, 32:407–451, 2004.

M. Fanty and R. Cole. Spoken letter recognition. In R.P. Lippman, J. Moody, and D.S. Touretzky, editors, *Advances in Neural Information Processing Systems*, volume 13, San Mateo, CA, 1991. Morgan Kaufmann.

J. Friedman. Multivariate adaptive regression splines. *Annals of Statistics*, 19:1–67, 1991.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33:1–22, 2010.

Ning Hao and Hao Helen Zhang. Interaction screening for ultrahigh-dimensional data. *Journal of the American Statistical Association*, 109(507):1285–1301, 2014.

Asad Haris, Daniela Witten, and Noah Simon. Convex modeling of interactions with strong heredity. *Journal of Computational and Graphical Statistics*, (just-accepted):1–35, 2015.

Trevor Hastie, Robert Tibshirani, Friedrich Leisch, Kurt Hornik, and Brian D. Ripley. *mda: Mixture and flexible discriminant analysis*, 2013. URL `http://CRAN.R-project.org/package=mda`. R package version 0.4-4.

Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis Bach. Proximal Methods for Hierarchical Sparse Coding. *Journal of Machine Learning Research*, 12:2297–2334, 2011.

Michael Lim and Trevor Hastie. Learning interactions through hierarchical group-lasso regularization. *arXiv preprint arXiv:1308.2719*, 2013.

Y. Lin and H.H. Zhang. Component selection and smoothing in multivariate nonparametric regression. *Annals of Statistics*, 35:2272–2297, 2006.

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

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

R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

P. Radchenko and G. James. Variable selection using adaptive nonlinear interaction structures in high dimensions. *Journal of the American Statistical Association*, 105:1541–1553, 2010.

P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman. Sparse additive models. *Journal of the Royal Statistical Society, Series B*, 71:1009–1030, 2009.

Rajen Dinesh Shah and Nicolai Meinshausen. Random intersection trees. *The Journal of Machine Learning Research*, 15(1):629–654, 2014.

R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

B. Turlach. Discussion of 'Least angle regression'. *Annals of Statistics*, 32:481–490, 2004.

M.J. Wainwright. Sharp thresholds for high-dimensional and noisy recovery of sparsity. *IEEE Transactions on Information Theory*, 55:2183–2202, 2009.

J. Wu, B. Devlin, S. Ringquist, M. Trucco, and K. Roeder. Screen and clean: a tool for identifying interactions in genome-wide association studies. *Genetic Epidemiology*, 34:275–285, 2010.

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

M. Yuan, V. R. Joseph, and Y. Lin. An efficient variable selection approach for analyzing designed experiments. *Technometrics*, 49:430–439, 2007.

M. Yuan, R. Joseph, and H. Zou. Structured variable selection and estimation. *Annals of Applied Statistics*, 3:1738–1757, 2009.

P. Zhao and B. Yu. On Model Selection Consistency of Lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

P. Zhao, G. Rocha, and B. Yu. The composite absolute families penalty for grouped and hierarchical variable selection. *Annals of Statistics*, 37:3648–3497, 2009.

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