SLOPE – Adaptive Variable Selection via Convex Optimization

Małgorzata Bogdan[a] Ewout van den Berg[b] Chiara Sabatti[c,d] Weijie Su[d]
Emmanuel J. Candès[d,e*]

May 2014[†]

[a] Department of Mathematics and Computer Science, Wrocław University of Technology, Poland
[b] IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, U.S.A.
[c] Department of Health Research and Policy, Stanford University, Stanford CA 94305, U.S.A.
[d] Department of Statistics, Stanford University, Stanford, CA 94305, U.S.A.
[e] Department of Mathematics, Stanford University, Stanford, CA 94305, U.S.A.

Abstract

We introduce a new estimator for the vector of coefficients β in the linear model $y = X\beta + z$, where X has dimensions $n \times p$ with p possibly larger than n. SLOPE, short for Sorted L-One Penalized Estimation, is the solution to

$$\min_{b \in \mathbb{R}^p} \quad \tfrac{1}{2}\|y - Xb\|_{\ell_2}^2 + \lambda_1 |b|_{(1)} + \lambda_2 |b|_{(2)} + \ldots + \lambda_p |b|_{(p)},$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$ and $|b|_{(1)} \geq |b|_{(2)} \geq \ldots \geq |b|_{(p)}$ are the decreasing absolute values of the entries of b. This is a convex program and we demonstrate a solution algorithm whose computational complexity is roughly comparable to that of classical ℓ_1 procedures such as the lasso. Here, the regularizer is a sorted ℓ_1 norm, which penalizes the regression coefficients according to their rank: the higher the rank—i. e. the stronger the signal—the larger the penalty. This is similar to the Benjamini-Hochberg procedure (BH) [9], which compares more significant p-values with more stringent thresholds. One notable choice of the sequence $\{\lambda_i\}$ is given by the BH critical values $\lambda_{\mathrm{BH}}(i) = z(1 - i \cdot q/2p)$, where $q \in (0, 1)$ and $z(\alpha)$ is the quantile of a standard normal distribution. SLOPE aims to provide finite sample guarantees on the selected model; of special interest is the false discovery rate (FDR), defined as the expected proportion of irrelevant regressors among all selected predictors. Under orthogonal designs, SLOPE with λ_{BH} provably controls FDR at level q. Moreover, it also appears to have appreciable inferential properties under more general designs X while having substantial power, as demonstrated in a series of experiments running on both simulated and real data.

Keywords. Sparse regression, variable selection, false discovery rate, lasso, sorted ℓ_1 penalized estimation (SLOPE).

Introduction

Analyzing and extracting information from datasets where the number of observations n is smaller than the number of variables p is one of the challenges of the present "big-data" world. In response,

[*] Corresponding author

[†] An earlier version of the paper appeared on arXiv.org in October 2013: arXiv:1310.1969v2

the statistics literature of the past two decades documents the development of a variety of methodological approaches to address this challenge. A frequently discussed problem is that of linking, through a linear model, a response variable y to a set of predictors $\{X_j\}$ taken from a very large family of possible explanatory variables. In this context, the lasso [42] and the Dantzig selector [19], for example, are computationally attractive procedures offering some theoretical guarantees, and with consequent wide-spread application. In spite of this, there are some scientific problems where the outcome of these procedures is not entirely satisfying, as they do not come with a machinery allowing us to make inferential statements on the validity of selected models in finite samples. To illustrate this, we resort to an example.

Consider a study where a geneticist has collected information about n individuals by having identified and measured all p possible genetics variants in a genomic region. The geneticist wishes to discover which variants cause a certain biological phenomenon, as an increase in blood cholesterol level. If we ponder a minute on how the results of this first study shall be followed up, we realize that (1) measuring cholesterol levels in a new individual is cheaper and faster than scoring his or her genetic variants, so that predicting y in future samples given the value of the relevant covariates is not an important goal. Instead, correctly identifying functional variants is relevant: on the one hand, (2) a genetic polymorphism correctly implicated in the determination of cholesterol levels points to a specific gene and to a biological pathway that might not be previously known to be related to blood lipid levels and, therefore, promotes an increase in our understanding of biological mechanisms, as well as providing targets for drug development. On the other hand, (3) the erroneous discovery of an association between a genetic variant and cholesterol levels will translate in considerable waste of time and money, which will be spent in trying to verify this association with direct manipulation experiments. Moreover, it is worth emphasizing that (4) some of the genetic variants in the study have a biological effect while others do not—there is a ground truth that statisticians can aim to discover. Finally, to be able to share his/her results with the scientific community in a convincing manner, (5) the researcher needs to be able to attach some finite sample confidence statements to her findings. In a more abstract language, our geneticist would need a tool that privileges correct model selection over minimization of prediction error, and would allow for inferential statements to be made on the validity of her selections. This paper presents a new methodology that attempts to address some of these needs.

We imagine that the n-dimensional response vector y is truly generated by a linear model of the form

$$y = X\beta + z,$$

with X an $n \times p$ design matrix, β a p-dimensional vector of regression coefficients and z the $n \times 1$ vector of random errors. We assume that all relevant variables (those with $\beta_i \neq 0$) are measured in addition to a large number of irrelevant ones. As any statistician knows, these are in themselves quite restrictive assumptions, but they are a widely accepted starting point. To formalize our goal, namely, the selection of important variables accompanied by a finite sample confidence statement, we seek a procedure that controls the expected proportion of irrelevant variables among the selected. In a scientific context where selecting a variable corresponds to making a discovery, we aim at controlling the False Discovery Rate (FDR). The FDR is of course a well-recognized measure of global error in multiple testing and effective procedures to control it are available: indeed, the Benjamini and Hochberg procedure (BH) [9] inspired the present proposal. It goes without saying that the connection between multiple testing and model selection has been made before (see e.g. [5], [22], [1], [2], [16]) and others in the recent literature have tackled the challenges encountered by our geneticists: we will discuss the differences between our approach and others in later sections as appropriate. The procedure we introduce in this paper is, however, entirely new. Variable selection is achieved by solving a convex problem not previously considered in the statistical literature, and which marries the advantages of ℓ_1 penalization with the adaptivity inherent in strategies like BH.

The reminder of the paper is organized as follows. Section 1 introduces SLOPE, our novel penalization strategy, motivating its construction in the context of orthogonal designs, and placing it in the context of current knowledge of effective model selection strategies. Section 2 describes the algorithm we developed and implemented to find SLOPE estimates. Section 3 showcases the application of our novel procedure in a variety of settings: we illustrate how it effectively solves a multiple testing problem with positively correlated test statistics; we discuss how regularizing parameters should be chosen in non-orthogonal designs; and we apply it to a genetic dataset not unlike that of our idealized example. Section 4 concludes the paper with a discussion comparing our methodology to other recently introduced proposals as well as outlining open problems.

1 Sorted L-One Penalized Estimation (SLOPE)

1.1 Adaptive penalization and multiple testing in orthogonal designs

To build intuition behind SLOPE, which encompasses our proposal for model selection in situations where $p > n$, we begin by considering the case of orthogonal designs and i.i.d. Gaussian errors with known standard deviation, as this makes the connection between model selection and multiple testing natural. Since the design is orthogonal, $X'X = I_p$, and the regression $y = X\beta + z$ with $z \sim \mathcal{N}(0, \sigma^2 I_n)$ can be recast as

$$\tilde{y} = X'y = X'X\beta + X'z = \beta + X'z \sim \mathcal{N}(\beta, \sigma^2 I_p). \tag{1.1}$$

In some sense, the problem of selecting the correct model reduces to the problem of testing the p hypotheses $H_{0,j} : \beta_j = 0$ versus two sided alternatives $H_{1,j} : \beta_i \neq 0$. When p is large, a multiple comparison correction strategy is called for and we consider two popular procedures.

- *Bonferroni's method.* To control the familywise error rate[1] (FWER) at level $\alpha \in [0, 1]$, one can apply Bonferroni's method, and reject $H_{0,j}$ if $|\tilde{y}_j|/\sigma > \Phi^{-1}(1 - \alpha/2p)$, where $\Phi^{-1}(\alpha)$ is the αth quantile of the standard normal distribution. Hence, Bonferroni's method defines a comparison threshold that only depends on the number p of covariates and on the noise level.

- *Benjamini-Hochberg procedure.* To control the FDR at level $q \in [0, 1]$, BH begins by sorting the entries of $\tilde{y}$ in decreasing order of magnitude, $|\tilde{y}|_{(1)} \geq |\tilde{y}|_{(2)} \geq \ldots \geq |\tilde{y}|_{(p)}$, which yields corresponding ordered hypotheses $H_{(1)}, \ldots, H_{(p)}$. (Note that here, as in the rest of the paper, (1) indicates the largest element of a set, instead of the smallest. This breaking with common convention allows us to keep (1) as the index for the most 'interesting' hypothesis). Then reject all hypotheses $H_{(i)}$ for which $i \leq i_{\mathrm{BH}}$, where i_{BH} is defined by

$$i_{\mathrm{BH}} = \max\{i : |\tilde{y}|_{(i)}/\sigma > \Phi^{-1}(1 - q_i)\}, \quad q_i = i \cdot q/2p \tag{1.2}$$

(with the convention that $i_{\mathrm{BH}} = 0$ if the set above is empty).[2] Letting V (resp. R) be the total number of false rejections (resp. total number of rejections), Benjamini and Hochberg

[1] Recall that the FWER is the probability of at least one false rejection.

[2] To minimize clutter we are being somewhat sloppy in the definition of BH although we have not introduced any error since our test statistics have continuous distributions.

[9] showed that

$$\mathrm{FDR} = \mathbb{E}\left[\frac{V}{R \vee 1}\right] = q\frac{p_0}{p}, \tag{1.3}$$

where p_0 is the number of true null hypotheses, $p_0 := |\{i : \beta_i = 0\}| = p - \|\beta\|_{\ell_0}$.

In contrast to Bonferroni's method, BH is an adaptive procedure in the sense that the threshold for rejection $|y|_{(i_{\mathrm{BH}})}$ is defined in a data-dependent fashion, and is sensitive to the sparsity and magnitude of the true signals. In a setting where there are many large β_j's, the last selected variable needs to pass a far less stringent threshold than it would in a situation where no β_j is truly different from 0. It has been shown [2, 15, 25] that this behavior allows BH to adapt to the unknown signal sparsity, resulting in some asymptotic optimality properties.

We now consider how the lasso [42] would behave in this setting. The solution to

$$\min_{b \in \mathbb{R}^p} \tfrac{1}{2}\|y - Xb\|_{\ell_2}^2 + \lambda \|b\|_{\ell_1} \tag{1.4}$$

in the case of orthogonal designs is given by soft-thresholding. In particular, the lasso estimate $\hat{\beta}_j$ is not zero if and only if $|\tilde{y}_j| > \lambda$. That is, variables are selected using a non-adaptive threshold λ. Mindful of the costs associated with the selection of irrelevant variables, we can control the FWER by setting $\lambda_{\mathrm{Bonf}} = \sigma \cdot \Phi^{-1}(1 - \alpha/2p) \approx \sigma \cdot \sqrt{2 \log p}$. This choice, however, is likely to result in a loss of power, and not to strike the right balance between errors of type I and missed discoveries. Choosing a value of λ substantially smaller than λ_{Bonf} in a non-data dependent fashion, would lead not only to a loss of FWER control, but also of FDR control since the FDR and FWER are identical measures under the global null in which all our variables are irrelevant. Another strategy is to use cross-validation. However, this data-dependent approach for selecting the regularization parameter λ targets the minimization of prediction error, and does not offer guarantees with respect to model selection (see Section 1.3.3). Our idea to achieve adaptivity, thereby increasing power while controlling some form of type-one error is to break the monolithic penalty $\lambda\|\beta\|_{\ell_1}$, which treats every variable in the same manner. Set

$$\lambda_{\mathrm{BH}}(i) \overset{\mathrm{def}}{=} \Phi^{-1}(1 - q_i), \quad q_i = i \cdot q/2p,$$

and consider the following program

$$\min_{b \in \mathbb{R}^p} \tfrac{1}{2}\|y - Xb\|_{\ell_2}^2 + \sigma \cdot \sum_{i=1}^{p} \lambda_{\mathrm{BH}}(i)|b|_{(i)}, \tag{1.5}$$

where $|b|_{(1)} \geq |b|_{(2)} \geq \ldots \geq |b|_{(p)}$ are the order statistics of the absolute values of the coordinates of b: in (1.5) different variables receive different levels of penalization depending on their relative importance. While the similarities of (1.5) with BH are evident, the solution to (1.5) is not a series of scalar thresholding operations, so that the procedures are not, even in this case of orthogonal variables, exactly equivalent. Nevertheless, an important results is this:

Theorem 1.1. *In the linear model with orthogonal design X and $z \sim \mathcal{N}(0, \sigma^2 I_n)$, the procedure (1.5) rejecting hypotheses for which $\hat{\beta}_j \neq 0$, has an FDR obeying*

$$\mathrm{FDR} = \mathbb{E}\left[\frac{V}{R \vee 1}\right] \leq q\frac{p_0}{p}. \tag{1.6}$$

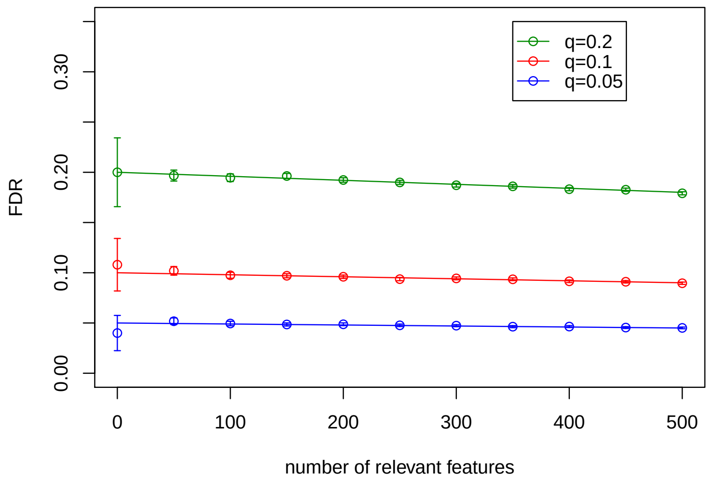


Figure 1: FDR of (1.5) in an orthogonal setting in which $n = p = 5000$. Straight lines correspond to $q \cdot p_0/p$, circles indicate the average FDP across 500 replicates, and bars correspond to ± 2 SE.

Theorem 1.1 is proven in the Appendix. Figure 1 illustrates the FDR achieved by (1.5) in simulations using a $5,000 \times 5,000$ orthogonal design X and nonzero regression coefficients equal to $5\sqrt{2 \log p}$.

We conclude this section with several remarks describing the properties of our procedure under orthogonal designs.

1. While the $\lambda_{\mathrm{BH}}(i)$'s are chosen with reference to BH, (1.5) is neither equivalent to the step-up procedure described above nor to the step-down version.[3]

2. The proposal (1.5) is sandwiched between the step-down and step-up procedures in the sense that it rejects at most as many hypotheses as the step-up procedure and at least as many as the step-down cousin, also known to control the FDR [38].

3. The fact that (1.5) controls FDR is not a trivial consequence of this sandwiching.

The observations above reinforce the fact that (1.5) is different from the procedure known as *FDR thresholding* developed by Abramovich and Benjamini [1] in the context of wavelet estimation and later analyzed in [2]. With $t_{\mathrm{FDR}} = |y|_{(i_{\mathrm{BH}})}$, FDR thresholding sets

$$\hat{\beta}_i = \begin{cases} y_i & |y_i| \geq t_{\mathrm{FDR}} \\ 0 & |y_i| < t_{\mathrm{FDR}}. \end{cases} \tag{1.7}$$

This is a hard-thresholding estimate but with a data-dependent threshold: the threshold decreases as more components are judged to be statistically significant. It has been shown that this simple estimate is asymptotically minimax throughout a range of sparsity classes [2]. Our method is similar in the sense that it also chooses an adaptive threshold reflecting the BH procedure. However, it

[3] The step-down version rejects $H_{(1)}, \ldots, H_{(i-1)}$, where i is the first time at which $|\tilde{y}_i|/\sigma \leq \Phi^{-1}(1 - q_i)$.

does not produce a hard-thresholding estimate. Rather, owing to nature of the sorted ℓ_1 norm, it outputs a sort of soft-thresholding estimate.

More importantly, it is not clear at all how one would extend (1.7) to nonorthogonal designs whereas the formulation 1.5 is not in any way linked to the orthogonal setting (even if the choice of the λ sequence in these more general cases is not trivial), as we are about to discuss.

1.2 SLOPE

While orthogonal designs have helped us define the program (1.5), this penalized estimation strategy is clearly applicable in more general settings. To make this explicit, it is useful to introduce the *sorted ℓ_1 norm*: letting $\lambda \neq 0$ be a nonincreasing sequence of nonnegative scalars,

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0, \tag{1.8}$$

we define the sorted-ℓ_1 norm of a vector $b \in \mathbb{R}^p$ as[4]

$$J_\lambda(b) = \lambda_1 |b|_{(1)} + \lambda_2 |b|_{(2)} + \ldots + \lambda_p |b|_{(p)}. \tag{1.9}$$

Proposition 1.2. *The functional* (1.9) *is a norm provided* (1.8) *holds.*

Proof If is obvious that $J_\lambda(b) = 0$ if only if $b = 0$, and that for any scalar $c \in \mathbb{R}$, $J_\lambda(cb) = |c| J_\lambda(b)$. Thus it remains to prove that $J_\lambda(b)$ is convex.

At first, suppose that $p = 2$. Then we can write J_λ as

$$J_\lambda(b) = (\lambda_1 - \lambda_2)|b|_{(1)} + \lambda_2(|b|_{(1)} + |b|_{(2)}) = (\lambda_1 - \lambda_2)\|b\|_{\ell_\infty} + \lambda_2 \|b\|_{\ell_1},$$

which implies that J_λ is the sum of two norms and is, therefore, convex.

Now for a general p, a simple calculation shows that

$$J_\lambda(b) = \sum_{i=1}^p (\lambda_i - \lambda_{i+1}) f_i(b), \qquad f_i(b) = \sum_{j \leq i} |b|_{(j)}$$

with the convention that $\lambda_{p+1} = 0$. Since each f_i is convex, each term $(\lambda_i - \lambda_{i+1}) f_i$ is convex and, therefore, their sum is also convex (a sum of convex functions is convex). To see why each f_i is convex, write

$$f_i(b) = \sup_{\epsilon \in C_i} f_\epsilon(b), \qquad \begin{array}{l} f_\epsilon(b) = \sum_{j=1}^p \epsilon_j |b_j|, \\ \\ C_i = \{\epsilon : \epsilon_j \in \{0,1\}, \#\{j : \epsilon_j \neq 0\} \leq i\}. \end{array} .$$

Since each $f_\epsilon(b)$ is convex and that the supremum of convex functions is convex, f_i is convex. ■

Now define SLOPE as the solution to

$$\text{minimize} \quad \tfrac{1}{2}\|y - Xb\|^2 + \sigma \sum_{i=1}^p \lambda_i |b|_{(i)}. \tag{1.10}$$

[4] Observe that when all the λ_i's take on an identical positive value, the sorted ℓ_1 norm reduces to the usual ℓ_1 norm (up to a multiplicative factor). Also, when $\lambda_1 > 0$ and $\lambda_2 = \ldots = \lambda_p = 0$, the sorted ℓ_1 norm reduces to the ℓ_∞ norm (again, up to a multiplicative factor).

As a convex program, SLOPE is tractable: as a matter of fact, we shall see in Section 2 that its computational cost is roughly the same as that of the lasso. Just as the sorted ℓ_1 norm is an extension of the ℓ_1 norm, SLOPE can be also viewed as an extension of the lasso. SLOPE's general formulation, however, allows to achieve the adaptivity we discussed earlier. The case of orthogonal regressors suggests one particular choice of a λ sequence and we will discuss others in later sections.

1.3 Relationship to other model selection strategies

Our purpose is to bring the program (1.10) to the attention of the statistical community: this is a computational tractable proposal for which we provide robust algorithms, it is very similar to BH when the design is orthogonal, and has promising properties in terms of FDR control for general designs. We now compare it with other commonly used approaches to model selection.

1.3.1 Methods based on ℓ_0 penalties

Canonical model selection procedures find estimates $\hat{\beta}$ by solving

$$\min_{b \in \mathbb{R}^p} \|y - Xb\|_{\ell_2}^2 + \lambda \|b\|_{\ell_0}, \tag{1.11}$$

where $\|b\|_{\ell_0}$ is the number of nonzero components in b. The idea behind such procedures is to achieve the best possible trade-off between the goodness of fit and the number of variables included in the model. Popular selection procedures such as AIC and C_p [3, 33] are of this form: when the errors are i.i.d. $\mathcal{N}(0, \sigma^2)$, AIC and C_p take $\lambda = 2\sigma^2$. In the high-dimensional regime, such a choice typically leads to including very many irrelevant variables in the model yielding rather poor predictive power in sparse settings (when the true regression coefficient sequence is sparse). In part to remedy this problem, Foster and George [22] developed the risk inflation criterion (RIC): they proposed using a larger value of λ effectively proportional to $2\sigma^2 \log p$, where we recall that p is the total number of variables in the study. Under orthogonal designs, if we associate nonzero fitted coefficients with rejections, this yields FWER control. Unfortunately, RIC is also rather conservative as it is a Bonferroni-style procedure and, therefore, it may not have much power in detecting those variables with nonvanishing regression coefficients unless they are very large.

The above dichotomy has been recognized for some time now and several researchers have proposed more adaptive strategies. One frequently discussed idea in the literature is to let the parameter λ in (1.11) decrease as the number of included variables increases. For instance, when minimizing

$$\|y - Xb\|_{\ell_2}^2 + p(\|b\|_{\ell_0}),$$

penalties with appealing information- and decision-theoretic properties are roughly of the form

$$p(k) = 2\sigma^2 k \log(p/k) \quad \text{or} \quad p(k) = 2\sigma^2 \sum_{1 \le j \le k} \log(p/j). \tag{1.12}$$

Among others, we refer the interested reader to [23, 14] and to [43] for a related approach.

Interestingly, for large p and small k these penalties are close to the FDR related penalty

$$p(k) = \sigma^2 \sum_{1 \le j \le k} \lambda_{\mathrm{BH}}^2(i), \tag{1.13}$$

proposed in [2] in the context of the estimation of the vector of normal means, or regression under the orthogonal design (see the preceding section) and further explored in [8]. Due to an implicit control of the number of false discoveries, similar model selection criteria are appealing in gene mapping studies (see e.g. [26]).

The problem with the selection strategies just described is that, in general, they are computationally intractable. Solving (1.12) would involve a brute-force search essentially requiring to fit least-squares estimates for *all* possible subsets of variables. This is not practical for even moderate values of p—e. g. for $p > 60$.

The decaying sequence of the smoothing parameters in SLOPE goes along the line of the adaptive ℓ_0 penalties proposed in (1.12) in which the 'cost per variable included' decreases as more get selected. However, SLOPE is computationally tractable and can be easily evaluated even for large-dimensional problems.

1.3.2 Adaptive lasso

Perhaps the most popular alternative to the computationally intractable ℓ_0 penalization methods is the lasso. We have already discussed some of the limitations of this approach with respect to FDR control and now wish to explore further the connections between SLOPE and variants of this procedure. It is well known that the lasso estimates of the regression coefficients are biased due to the shrinkage imposed by the ℓ_1 penalty. To increase the accuracy of the estimation of large signals and eliminate some false discoveries the adaptive or reweighted versions of lasso were introduced (see e.g. [44] or [20]). In these procedures the smoothing parameters $\lambda_1, \ldots, \lambda_p$ are adjusted to the unknown signal magnitudes based on some estimates of regression coefficients, perhaps obtained through previous iterations of lasso. The idea is then to consider a weighted penalty $\sum_i w_i|b_i|$, where w_i is inversely proportional to the estimated magnitudes so that large regression coefficients are shrunk less than smaller ones. In some circumstances, such adaptive versions of lasso outperform the regular lasso for selection [44].

The idea behind SLOPE is entirely different. In the adaptive lasso, the penalty tends to decrease as the magnitude of coefficients increases. In our approach, the exact opposite happens. This comes from the fact that we seek to adapt to the unknown signal sparsity while controlling the FDR. As shown in [2], FDR controlling properties can have interesting consequences for estimation. In practice, since the SLOPE sequence $\lambda_1 \geq \ldots \geq \lambda_p$ leading to FDR control is typically rather large, we do not recommend using SLOPE directly for the estimation of regression coefficients. Instead we propose the following two-stage procedure: in the first step, SLOPE is used to identify significant predictors; in the second step, the corresponding regression coefficients are estimated using the least squares method within the identified sparse regression model. Such a two-step procedure can be thought of as an extreme case of reweighting, where those selected variables are not penalized while those that are not receive an infinite penalty. As shown below, these estimates have very good properties when the coefficient sequence β is sparse.

1.3.3 A first illustrative simulation

To concretely illustrate the specific behavior of SLOPE compared to more traditional penalized approaches, we rely on the simulation of a relatively simple data structure. We set $n = p = 5000$ and generate the entries of the design matrix with i.i.d. $\mathcal{N}(0, 1/n)$ entries. The number of true signals k varies between 0 and 50 and their magnitudes are set to $\beta_i = \sqrt{2 \log p} \approx 4.1$, while the variance of the error term is assumed known and equal to 1. This choice of model parameters makes the signal barely distinguishable from the noise.

We fit these observations with three procedures: 1) lasso with parameter $\lambda_{\text{Bonf}} = \sigma \cdot \Phi^{-1}(1 - \alpha/2p)$, which controls FWER weakly; 2) lasso with the smoothing parameter λ_{CV} chosen with 10-fold cross-validation; 3) SLOPE with a sequence $\lambda_1, \ldots, \lambda_p$ defined in Section 3.2.2, expression (3.7). The level α for λ_{Bonf} and q for FDR control in SLOPE are both set to 0.1. To compensate for the fact that lasso with λ_{Bonf} and SLOPE tend to apply a much more stringent penalization than lasso with λ_{CV}—which aims to minimize prediction error—we have "de-biased" their resulting β estimates when comparing the methods on the ground of prediction error: that is, we have used ordinary least squares to estimate the coefficients of the variables selected by lasso–λ_{Bonf} and SLOPE.

We compare the procedures on the basis of three criteria: a) FDR, b) power, c) relative squared error $\|X\hat{\beta} - X\beta\|_{\ell^2}^2 / \|X\beta\|_{\ell_2}^2$. Note that only the first of these measures is meaningful for the case where $k = 0$, and in such case FDR=FWER. Figure 2 reports the results of 500 independent replicates.

As illustrated in Figures 2a and 2b, the three approaches exhibit quite dramatically different properties with respect to model selection: SLOPE and lasso–λ_{Bonf} control FDR, paying a corresponding price in terms of power, while lasso–λ_{CV} offers no FDR control. In more detail, SLOPE controls FDR at level 0.1 for the explored range of k; as k increases, its power goes from 45% to 70%. Lasso–λ_{Bonf} has FDR =0.1 at $k = 0$, and a much lower one for the remaining values of k; this results in a loss of power with respect to SLOPE: irrespective of k, power is less than 45%. Cross-validation chooses a λ that minimizes an estimate of prediction error: this would increase if irrelevant variables are selected, but also if important ones are omitted, or if their coefficients are excessively shrunk. As a result, the λ_{CV} is quite smaller than a penalization parameter chosen with FDR control in mind. This results in greater power than SLOPE, but with a much larger FDR (80% on average).

Figure 2c illustrates the relative mean-square error, which serves as a measure of prediction accuracy. We recall that we used de-biased versions of lasso–λ_{Bonf} and SLOPE, while we left the predictions from lasso–λ_{CV} untouched since these were chosen to minimize prediction error. It is remarkable how, despite the fact that lasso–λ_{CV} has higher power, SLOPE has lower percentage prediction error for all the sparsity levels considered.

2 Algorithms

In this section, we present effective algorithms for computing the solution to SLOPE (1.10), which rely on the numerical evaluation of the proximity operator (prox) to the sorted ℓ_1 norm.

2.1 Proximal gradient algorithms

SLOPE is a convex optimization problem of the form

$$\text{minimize} \quad f(b) = g(b) + h(b), \tag{2.1}$$

where g is smooth and convex, and h is convex but not smooth. In SLOPE, g is the residual sum of squares and, therfore, quadratic while h is the sorted ℓ_1 norm. A general class of algorithms for solving problems of this kind are known as *proximal gradient methods*, see [36, 37] and references

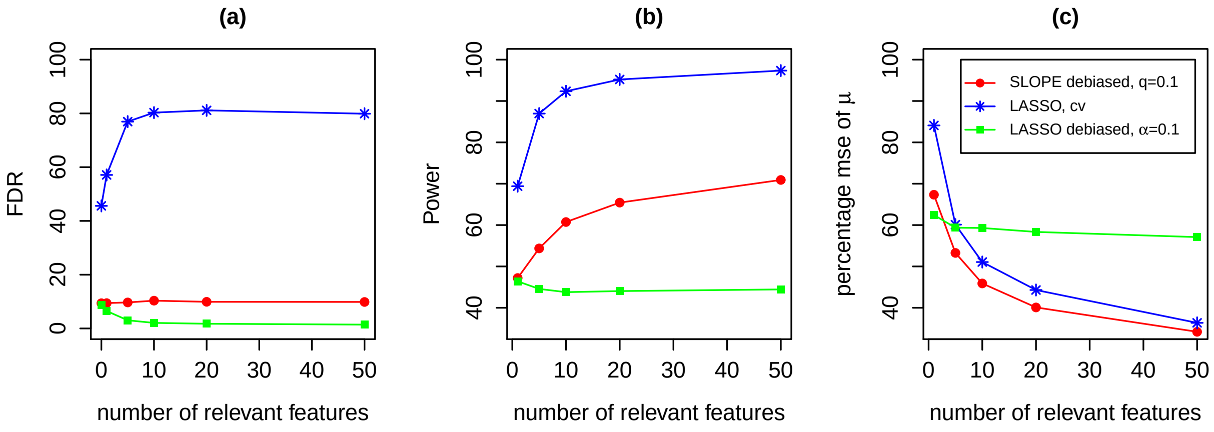


Figure 2: Properties of different procedures as a function of the true number of nonzero regression coefficients: (a) FDR, (b) Power, (c) Relative MSE defined as the average of $100 \cdot \|\hat{\mu} - \mu\|^2_{\ell_2} / \|\mu\|^2_{\ell_2}$, with $\mu = X\beta$, $\hat{\mu} = X\hat{\beta}$. The design matrix entries are i.i.d. $\mathcal{N}(0, 1/n)$, $n = p = 5,000$, all nonzero regression coefficients are equal to $\sqrt{2 \log p} \approx 4.13$, and $\sigma^2 = 1$. Each point in the figures corresponds to the average of 500 replicates.

therein. These are iterative algorithms operating as follows: at each iteration, we hold a guess b of the solution, and compute a local approximation to the smooth term g of the form

$$g(b) + \langle \nabla g(b), x - b \rangle + \frac{1}{2t} \|x - b\|^2_{\ell_2}.$$

This is interpreted as the sum of a Taylor approximation of g and of a proximity term; as we shall see, this term is responsible for searching an update reasonably close to the current guess b, and t can be thought of as a step size. Then the next guess b_+ is the unique solution to

$$\begin{aligned} b_+ &= \text{arg min}_x \quad \left\{ g(b) + \langle \nabla g(b), x - b \rangle + \frac{1}{2t} \|x - b\|^2_{\ell_2} + h(x) \right\} \\ &= \text{arg min}_x \quad \left\{ \frac{1}{2t} \|(b - t\nabla g(b)) - x\|^2_{\ell_2} + h(x) \right\} \end{aligned}$$

(unicity follows from strong convexity). Hence, in the case where h vanishes, this reduces to straightforward gradient descent while in the case where h is an indicator function equal to 0 if b lies in some convex set C and equal to $+\infty$ otherwise, this reduces to the well-known projected gradient method. (In the latter case, minimizing $g(b) + h(b)$ is equivalent to minimizing $g(b)$ subject to $b \in C$.) In the literature, the mapping

$$x(y) = \text{arg min}_x \quad \left\{ \frac{1}{2t} \|y - x\|^2_{\ell_2} + h(x) \right\}$$

is called the proximal mapping or prox for short, and denoted by $x = \text{prox}_{th}(y)$.

The prox of the ℓ_1 norm is given by entry-wise soft-thresholding [37, page 150] so that a proximal gradient method to solve the lasso would take the following form: starting with $b^0 \in \mathbb{R}^p$, inductively define

$$b^{k+1} = \eta_{\lambda t_k}(b^k - t_k X'(Xb^k - y); t_k \lambda),$$

where $\eta_\lambda(y) = \text{sign}(y) \cdot (|y| - \lambda)_+$ and $\{t_k\}$ is a sequence of step sizes. Hence, we can solve the lasso by iterative soft thresholding.

It turns out that one can compute the prox to the sorted ℓ_1 norm in nearly the same amount of time as it takes to apply soft thesholding. In particular, assuming that the entries are sorted, we shall demonstrate a linear-time algorithm. Hence, we may consider a proximal gradient method for SLOPE as in Algorithm 1.

Algorithm 1 Proximal gradient algorithm for SLOPE (1.10)

```
Require: $b^0 \in \mathbb{R}^p$
for $k = 0, 1, \ldots$ do
    $b^{k+1} = \text{prox}_{t_k J_\lambda}(b^k - t_k X'(Xb^k - y))$
end for
```

It is well known that the algorithm converges (in the sense that $f(b^k)$, where f is the objective functional, converges to the optimal value) under some conditions on the sequence of step sizes $\{t_k\}$. Valid choices include step sizes obeying $t_k < 2/\|X\|^2$ and step sizes obtained by backtracking line search, see [7, 6]. Further, one can use duality theory to derive concrete stopping criteria, see Appendix B for details.

Many variants are of course possible and one may entertain accelerated proximal gradient methods in the spirit of FISTA, see [6] and [35, 36]. The scheme below is adapted from [6].

Algorithm 2 Accelerated proximal gradient algorithm for SLOPE (1.10)

```
Require: $b^0 \in \mathbb{R}^p$, and set $a^0 = b^0$ and $\theta_0 = 1$
for $k = 0, 1, \ldots$ do
    $b^{k+1} = \text{prox}_{t_k J_\lambda}(a^k - t_k X'(Xa^k - y))$
    $\theta_{k+1}^{-1} = \frac{1}{2}(1 + \sqrt{1 + 4/\theta_k^2})$
    $a^{k+1} = b^{k+1} + \theta_{k+1}(\theta_k^{-1} - 1)(b^{k+1} - b^k)$
end for
```

The code in our numerical experiments uses a straightforward implementation of the standard FISTA algorithm, along with problem-specific stopping criteria. Standalone Matlab and R implementations of the algorithm are available at `http://www-stat.stanford.edu/~candes/SortedL1`. In addition, the TFOCS package available at `http://cvxr.com` [7] implements Algorithms 1 and 2 as well as many variants; for instance, the Matlab code below prepares the prox and then solves the SLOPE problem,

```
prox = prox_Sl1(lambda);
beta = tfocs( smooth_quad, { X, -y }, prox, beta0, opts );
```

Here `beta0` is an initial guess (which can be omitted) and `opts` are options specifying the methods and parameters one would want to use, please see [7] for details. There is also a one-liner with default options which goes like this:

```
beta  = solver_SLOPE( X, y, lambda);
```

2.2 Fast prox algorithm

Given $y \in \mathbb{R}^n$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$, the prox to the sorted ℓ_1 norm is the unique solution to

$$\operatorname{prox}(y; \lambda) := \operatorname{argmin}_{x \in \mathbb{R}^n} \ \tfrac{1}{2}\|y - x\|_{\ell_2}^2 + \sum_{i=1}^n \lambda_i |x|_{(i)}. \tag{2.2}$$

Without loss of generality we can make the following assumption:

Assumption 2.1. *The vector y obeys $y_1 \geq y_2 \geq \cdots \geq y_n \geq 0$.*

At the solution to (2.2), the sign of each $x_i \neq 0$ will match that of y_i. It therefore suffices to solve the problem for $|y|$ and restore the signs in a post-processing step, if needed. Likewise, note that applying any permutation P to y results in a solution Px. We can thus choose a permutation that sorts the entries in y and apply its inverse to obtain the desired solution.

Proposition 2.2. *Under Assumption 2.1, the solution x to* (2.2) *satisfies $x_1 \geq x_2 \geq \cdots \geq x_n \geq 0$.*

Proof Suppose that $x_i < x_j$ for $i < j$ (and $y_i > y_j$), and form a copy x' of x with entries i and j exchanged. Letting f be the objective functional in (2.2), we have

$$f(x) - f(x') = \tfrac{1}{2}(y_i - x_i)^2 + \tfrac{1}{2}(y_j - x_j)^2 - \tfrac{1}{2}(y_i - x_j)^2 - \tfrac{1}{2}(y_j - x_i)^2.$$

This follows from the fact that the sorted ℓ_1 norm takes on the same value at x and x' and that all the quadratic terms cancel but those for i and j. This gives

$$f(x) - f(x') = x_j y_i - x_i y_i + x_i y_j - x_j y_j = (x_j - x_i)(y_i - y_j) > 0,$$

which shows that the objective x' is strictly smaller, thereby contradicting optimality of x. ■

Under Assumption 2.1 we can reformulate (2.2) as

$$\begin{array}{ll} \text{minimize} & \tfrac{1}{2}\|y - x\|_{\ell_2}^2 + \sum_{i=1}^n \lambda_i x_i \\ \text{subject to} & x_1 \geq x_2 \geq \cdots \geq x_n \geq 0. \end{array} \tag{2.3}$$

In other words, the prox is the solution to a quadratic program (QP). However, we do not suggest performing the prox calculation by calling a standard QP solver, rather we introduce the FastProxSL1 algorithm for computing the prox: for ease of exposition, we introduce Algorithm 3 in its simplest form before presenting a stack implementation (Algorithm 4) running in $O(n)$ flops.

Algorithm 3 FastProxSL1

```
input: Nonnegative and nonincreasing sequences $y$ and $\lambda$.
while $y - \lambda$ is not nonincreasing do
  Identify strictly increasing subsequences, i.e. segments $i : j$ such that
      $$y_i - \lambda_i < y_{i+1} - \lambda_{i+1} < \ldots < y_j - \lambda_j. \qquad (2.4)$$
  Replace the values of $y$ and $\lambda$ over such segments by their average value: for $k \in \{i, i+1, \ldots, j\}$
      $$y_k \leftarrow \frac{1}{j-i+1} \sum_{i \le k \le j} y_k, \qquad \lambda_k \leftarrow \frac{1}{j-i+1} \sum_{i \le k \le j} \lambda_k.$$
end while
output: $x = (y - \lambda)_+$.
```

Algorithm 3, which terminates in at most n steps, is simple to understand: we simply keep on averaging until the monotonicity property holds, at which point the solution is known in closed form. The key point establishing the correctness of the algorithm is that the update does not change the value of the prox. This is formalized below.

Lemma 2.3. *The solution does not change after each update; formally, letting (y^+, λ^+) be the updated value of (y, λ) after one pass in Algorithm 3,*

$$\operatorname{prox}(y; \lambda) = \operatorname{prox}(y^+; \lambda^+).$$

Next, if $(y - \lambda)_+$ is nonincreasing, then it is the solution to (2.2)*, i.e.* $\operatorname{prox}(y; \lambda) = (y - \lambda)_+$.

This lemma, whose proof is in the Appendix, guarantees that the FastProxSL1 algorithm finds the solution to (2.2) in a finite number of steps.

As stated earlier, it is possible to obtain a careful $O(n)$ implementation of FastProxSL1. Below we present a stack-based approach. We use tuple notation $(a, b)_i = (c, d)$ to denote $a_i = c$, $b_i = d$.

	$p = 10^5$	$p = 10^6$	$p = 10^7$
Total prox time (sec.)	9.82e-03	1.11e-01	1.20e+00
Prox time after normalization (sec.)	6.57e-05	4.96e-05	5.21e-05

Table 1: Average runtimes of the stack-based prox implementation with normalization steps (sorting and sign changes) included, respectively excluded.

Algorithm 4 Stack-based algorithm for FastProxSL1.

```
input: Nonnegative and nonincreasing sequences $y$ and $\lambda$.
# Find optimal group levels
$t \leftarrow 0$
for $k = 1$ to $n$ do
    $t \leftarrow t + 1$
    $(i, j, s, w)_t = (k,\ k,\ y_i - \lambda_i,\ (y_i - \lambda_i)_+)$
    while $(t > 1)$ and $(w_{t-1} \leq w_t)$ do
        $(i, j, s, w)_{t-1} \leftarrow (i_{t-1},\ j_t,\ s_{t-1} + s_t, (\frac{j_{t-1}-i_{t-1}+1}{j_t-i_{t-1}+1} \cdot s_{t-1} + \frac{j_t-i_t+1}{j_t-i_{i-1}+1} \cdot s_t)_+)$
        Delete $(i, j, s, w)_t$, $t \leftarrow t - 1$
    end while
end for
# Set entries in $x$ for each block
for $\ell = 1$ to $t$ do
    for $k = i_\ell$ to $j_\ell$ do
        $x_k \leftarrow w_\ell$
    end for
end for
```

For the complexity of the algorithm note that we create a total of n new tuples. Each of these tuple is merged into a previous tuple at most once. Since the merge takes a constant amount of time the algorithm has the desired $O(n)$ complexity.

With this paper, we are making available a C, a Matlab, and an R implementation of the stack-based algorithm at `http://www-stat.stanford.edu/~candes/SortedL1`. The algorithm is also included in the current version of the TFOCS package. To give an idea of the speed, we applied the code to a series of vectors with fixed length and varying sparsity levels. The average runtimes measured on a MacBook Pro equipped with a 2.66 GHz Intel Core i7 are reported in Table 1.

2.3 Connection with isotonic regression

Brad Efron informed us about the connection between the FastProxSL1 algorithm for SLOPE and a simple iterative algorithm for solving istonic problems called the pool adjacent violators algorithm (PAVA) [30, 4]. A simple instance of an isotonic regression problem involves fitting data in a least squares sense in such a way that the fitted values are monotone:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|y - x\|_{\ell_2}^2 \\ \text{subject to} & x_1 \geq x_2 \geq \cdots \geq x_n \end{array} \tag{2.5}$$

here, y is a vector of observations and x is the vector of fitted values, which are here constrained to be nonincreasing. We have chosen this formulation to emphasize the connection with (2.3). Indeed, our QP (2.3) is equivalent to

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\sum_{i=1}^{n}(y_i - \lambda_i - x_i)^2 \\ \text{subject to} & x_1 \geq x_2 \geq \cdots \geq x_n \geq 0 \end{array}$$

so that we see are really solving an isotonic regression problem with data $y_i - \lambda_i$. Algorithm 3 is then a version of PAVA as described in [4], see [13, 27] for related work and connections with active set methods.

In closing, we would also like to note that De Leeuw, Hornik and Mair [32] have contributed a nice R package called `isotone` available at `http://cran.r-project.org/web/packages/isotone/index.html`, which can also be used to compute the prox to the sorted ℓ_1 norm, and thus to help fit SLOPE models.

3 Results

We now turn to illustrate the performance of our SLOPE proposal in three different ways. First, we describe a multiple-testing situation where reducing the problem to a model selection setting and applying SLOPE assures FDR control, and results in a testing procedure with appreciable properties. Secondly, we discuss guiding principles to choose the sequence of λ_i's in general settings, and illustrate the efficacy of the proposals with simulations. Thirdly, we apply SLOPE to a data set collected in genetics investigations.

3.1 An application to multiple testing

In this section we show how SLOPE can be used as an effective multiple comparison controlling procedure in a testing problem with a specific correlation structure. Consider the following situation. Scientists perform $p = 1,000$ experiments in each of 5 randomly selected laboratories, resulting in observations that can be modeled as

$$y_{i,j} = \mu_i + \tau_j + z_{i,j}, \quad 1 \leq i \leq 1000,\ 1 \leq j \leq 5,$$

where the laboratory effects τ_j are i.i.d. $\mathcal{N}(0, \sigma_\tau^2)$ random variables and the errors $z_{i,j}$ are i.i.d. $\mathcal{N}(0, \sigma_z^2)$, with the τ and z sequences independent of each other. It is of interest to test whether $H_j : \mu_j = 0$ versus a two-sided alternative. Averaging the scores over all five labs, results in

$$\bar{y}_i = \mu_i + \bar{\tau} + \bar{z}_i, \quad 1 \leq i \leq 1000,$$

with $\bar{y} \sim \mathcal{N}(0, \Sigma)$ and $\Sigma_{i,i} = \frac{1}{5}(\sigma_\tau^2 + \sigma_z^2) = \sigma^2$ and $\Sigma_{i,j} = \frac{1}{5}\sigma_\tau^2 = \rho$ for $i \neq j$.

The problem has then been reduced to testing if the means of a multivariate Gaussian vector with equicorrelated entries do not vanish. One possible approach is to use each $\bar{y}_i$ as a marginal test, and rely on the Benjamini-Hochberg procedure to control FDR. That is, we order $|\bar{y}|_{(1)} \geq |\bar{y}|_{(2)} \geq \ldots \geq |\bar{y}|_{(p)}$ and apply the step-up procedure with critical values equal to $\sigma \cdot \Phi^{-1}(1 - iq/2p)$. While BH has not been proven to control FDR for families of two-sided tests, according to our simulations it seems to do so when the data are multivariate normal. Therefore, in our simulations we use the original BH rather than the theoretically justified but more conservative adjustment from [10, Theorem 1.3], which would substantially reduce the performance.

Another possible approach is to 'whiten the noise' and express our multiple testing problem in the form of a regression equation

$$\tilde{y} = \Sigma^{-1/2}\bar{y} = \Sigma^{-1/2}\mu + \epsilon, \tag{3.1}$$

where $\epsilon \sim \mathcal{N}(0, I_p)$. Treating $\Sigma^{-1/2}$ as the regression design matrix, our problem is equivalent to classical model selection: identify the non-zero components of the vector μ of regression coefficients.[5] Note that while the matrix Σ is far from being diagonal, $\Sigma^{-1/2}$ is diagonally dominant. For example when $\sigma^2 = 1$ and $\rho = 0.5$ then $\Sigma^{-1/2}_{i,i} = 1.4128$ and $\Sigma^{-1/2}_{i,j} = -0.0014$ for $i \neq j$. Thus, every low-dimensional sub-model obtained by selecting few columns of the design matrix $\Sigma^{-1/2}$ will be very close to orthogonal. In summary, the transformation (3.1) reduces the multiple-testing problem with strongly positively correlated test statistics to a problem of model selection under approximately orthogonal design, which is well suited for the application of SLOPE with the λ_{BH} values.

To compare the performances of these two approaches, we simulate a sequence of sparse multiple testing problems, with $\sigma^2 = 1$, $\rho = 0.5$ and the number k of nonzero μ_i's varying between 0 and 80. To use SLOPE, we center the vector $\tilde{y}$ by subtracting its mean, and center and standardize columns of $\Sigma^{-1/2}$ so that they have zero mean and unit norm. With this normalization, all the nonzero means are set to $\sqrt{2\log p} \approx 3.7$, so as to obtain moderate power. Figure 3 reports the results of these simulations, averaged over 500 independent replicates.

In our setting, SLOPE keeps FDR at the nominal level as long as $k \leq 40$. Then its FDR slowly increases, but for $k \leq 80$ it is still very close to the nominal level. On the other hand, the BH procedure on the marginal tests is too conservative: the FDR is below the nominal level (Figure 3a and 3b), resulting in a loss of power with respect to SLOPE (Figure 3c). Moreover, the False Discovery Proportion (FDP) in the marginal tests with BH correction appears more variable across replicates than that of SLOPE (Figure 3a, 3b and 3d). Figure 4 presents the results in greater detail for $q = 0.1$ and $k = 50$: in approximately 75% of the cases the observed FDP for BH is equal to 0, while in the remaining 25%, it takes values which are distributed over the whole interval (0,1). This behavior is undesirable. On the one hand, FDP = 0 typically equates with little discoveries (and hence power loss). On the other hand, if many FDP = 0 contribute to the average in the FDR, this quantity is kept below the desired level q even if, when there are discoveries, a large number of them are false. Indeed, in approximately 35% of all cases BH on the marginal tests did not make any rejections (i.e., $R = 0$); and conditional on $R > 0$, the mean of FDP is equal to 0.22 with a standard deviation of 0.28, which clearly shows that the observed FDP is typically far away from the nominal value of $q = 0.1$. In other words, while BH controls FDR on average, the scientists would either make no discoveries or have very little confidence on those actually made. In contrast, SLOPE results in a more predictable FDP and substantially larger and more predictable True Positive Proportion (TPP, fraction of correctly identified true signals), see Figure 4.

3.2 Choosing λ in general settings.

In the previous sections we observed that under the orthogonal designs lasso with $\lambda_{\mathrm{Bonf}} = \sigma \cdot \Phi^{-1}(1 - \alpha/2p)$ controls FWER at the level α, while SLOPE with the sequence $\lambda = \lambda_{\mathrm{BH}}$ controls FDR at the level q. We are interested, however, in applying these procedures in more general settings, specifically when $p > n$ and there is some correlation among the explanatory variables, and when the value of σ^2 is not known. We start tackling the first situation. Correlation among

[5] To be explicit, (3.1) is the basic regression model with $X = \Sigma^{-1/2}$ and $\beta = \mu$.

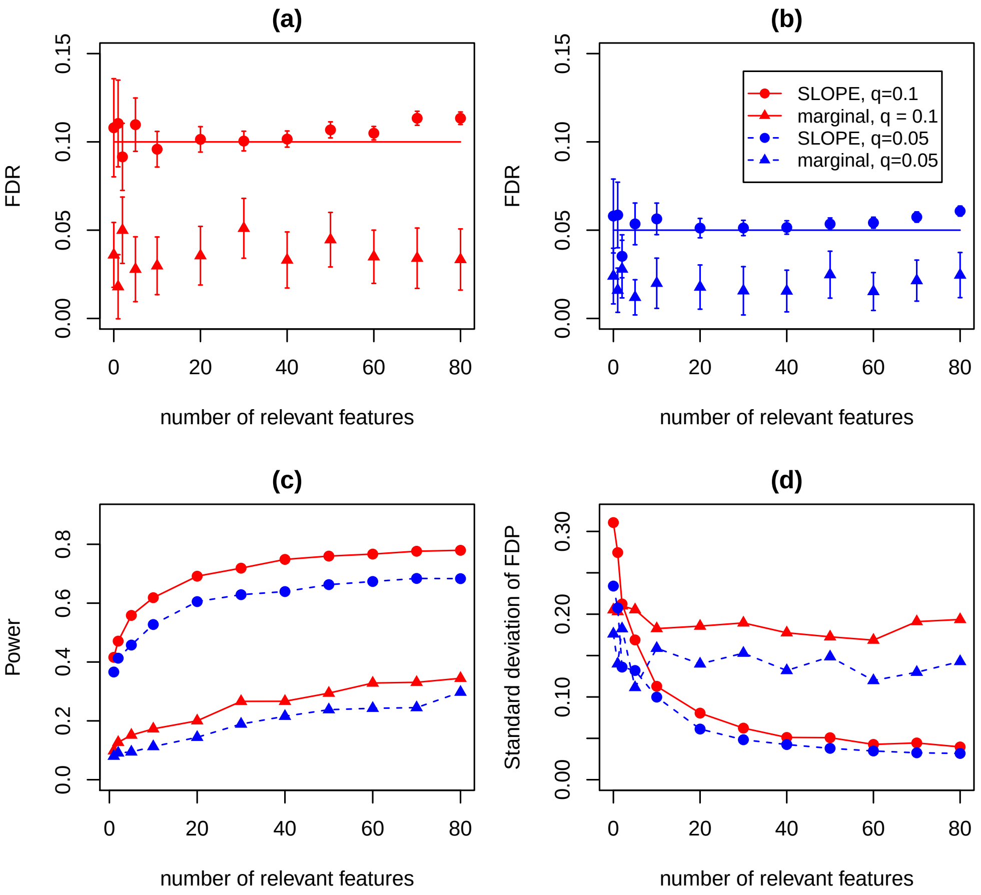


Figure 3: Simulation results for testing multiple means from correlated statistics. (a)-(b) Mean FDP ± 2 SE for SLOPE and marginal tests as a function of k. (c) Power plot. (d) Variability of the false discovery proportions for both methods.

regressors notoriously introduces a series of complications in the statistical analysis of linear models, ranging from the increased computational costs that motivated the early popularity of orthogonal designs, to the conceptual difficulties of distinguishing causal variables among correlated ones. Indeed, recent results on the consistency of ℓ_1 penalization methods typically require some form of partial orthogonality. SLOPE and lasso aim at finite sample properties, but it would not be surprising if departures from orthogonality were to have a serious effect. To explore this, we study the performance of lasso and SLOPE in the case where the entries of the design matrix are generated independently from the $\mathcal{N}(0, 1/n)$ distribution. Specifically, we consider two Gaussian designs with $n = 5000$: one with $p = 2n$ =10,000 and one with $p = n/2 = 2500$. We set the value of non-zero coefficients to $5\sqrt{2 \log p}$ and consider situations where the number of important variables ranges between 0 and 100. Figure 5 reports the results of 500 independent simulations.

While the columns of the design matrix are realizations of independent random variables, their inner products are not equal to zero due to randomness. Our simulations show that even such a small departure from orthogonality can have a substantial impact on the properties of the model

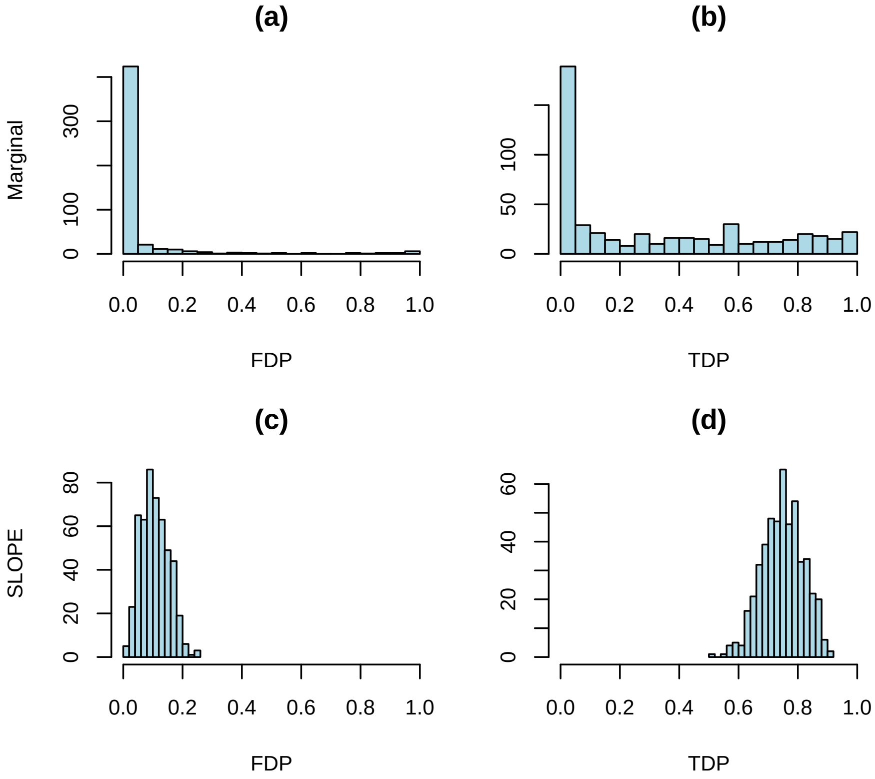


Figure 4: Testing example with $q = 0.1$ and $k = 50$. Top row refers to marginal tests, and bottom row to SLOPE. Histograms of false discovery proportions are in the first column and of true discovery proportions in the second.

selection procedures. In Figure 5 it can be observed that when $k = 0$ (data is pure noise), then both lasso-λ_{Bonf} and SLOPE both control their targeted error rates (FWER and FDR) at the nominal level. However, this control is lost as the number k of nonzero coefficients increases, with a departure that is more severe when the ratio between p/n is larger.

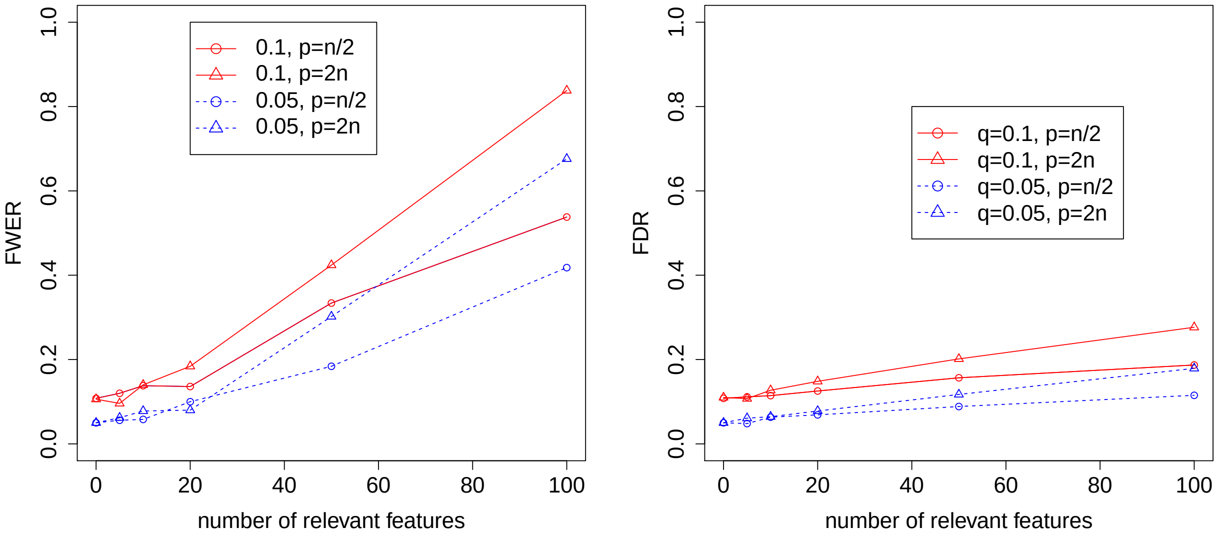
3.2.1 The effect of shrinkage

What is behind this fairly strong effect and is it possible to choose a λ sequence to compensate it? Some useful insights come from studying the solution of the lasso. Assume that the columns of X have unit norm and that $z \sim \mathcal{N}(0, 1)$. Then the optimality conditions for the lasso give

$$\begin{aligned}\hat{\beta} = \eta_\lambda(\hat{\beta} - X'(X\hat{\beta} - y)) &= \eta_\lambda(\hat{\beta} - X'(X\hat{\beta} - X\beta - z)) \\ &= \eta_\lambda(\hat{\beta} - X'X(\hat{\beta} - \beta) + X'z), \end{aligned} \tag{3.2}$$

where η_λ is the soft-thresholding operator, $\eta_\lambda(t) = \text{sgn}(t)(|t| - \lambda)_+$, applied componentwise. Defining $v_i = \langle X_i, \sum_{j \neq i} X_j(\beta_j - \hat{\beta}_j)\rangle$, we can write

$$\hat{\beta}_i = \eta_\lambda(\beta_i + X_i'z + v_i), \tag{3.3}$$



(a) FWER of lasso with λ_{Bonf}.

(b) FDR of SLOPE with λ_{BH}.

Figure 5: Observed FWER for lasso with λ_{Bonf} and FDR for SLOPE with λ_{BH} under Gaussian design and $n = 5,000$.

which expresses the relation between the estimated value of $\hat{\beta}_i$ and its true value β_i. If the variables are orthogonal, the v_i's are identically equal to 0, leading to $\hat{\beta}_i = \eta_\lambda(\beta_i + X_i'z)$. Conditionally on X, $X_i'z \sim \mathcal{N}(0,1)$ and by using Bonferroni's method, one can choose λ such that $\mathbb{P}(\max_i |X_i'z| > \lambda) \leq \alpha$. When X is not orthogonal, however, $v_i \neq 0$, and its size increases with the estimation error for β_j with $j \neq i$. While there is an error associated with the selection of irrelevant variables, suppose that this has been eliminated and that the correct set of variables has been included in the model. A procedure such as the lasso or SLOPE will still make an error due to the shrinkage of regression coefficients: even in a perfect situation, where all the k relevant variables, and those alone, have been selected, and when all columns of the design matrix are realizations of independent random variables, v_i will not be zero. Rather its squared magnitude will be on the order of $\lambda^2 \cdot k/n$. In other words, the variance that would determine the correct Bonferroni threshold should be on the order $1 + \lambda^2 \cdot k/n$, where k is the correct number of variables in the model. In reality, the true k is not known a priori, and the selected k depends on the value of the smoothing parameter λ, so that it is not trivial to implement this correction in the lasso. SLOPE, however, uses a decreasing sequence λ, analogous to a step-down procedure, and this extra noise due to the shrinkage of relevant variables can be incorporated by progressively modifying the λ sequence. In evocative, if not exact terms, λ_1 is used to select the first variable to enter the model: at this stage we are not aware of any variable whose shrunk coefficient is 'effectively increasing' the noise level, and we can keep $\lambda_1 = \lambda_{\mathrm{BH}}(1)$. The value of λ_2 determines the second variable to enter the model and, hence, we know that there is already one important variable whose coefficient has been shrunk by roughly $\lambda_{\mathrm{BH}}(1)$: we can use this information to re-define λ_2. Similarly, when using λ_3 to identify the third variable, we know of two relevant regressors whose coefficients have been shrunk by amounts determined by λ_1 and λ_2, and so on. What follows is an attempt to make this intuition more precise, accounting for the fact that the sequence λ needs to be determined a priori, and we need to make a prediction on the values of the cross products $X_i'X_j$ appearing in the definition of v_i. Before we turn to this, we want to underscore how the explanation offered in this section for the loss of FDR control is consistent with patterns evident from Figure 5: the problem is more serious as k increases (and hence the effect of shrinkage is felt on a larger number of variables), and as the ratio p/n increases (which for Gaussian designs results in larger empirical correlation $|X_i'X_j|$). Our loose analysis suggests that when k is really small, SLOPE with λ_{BH} yields an FDR that is close to the nominal level, as empirically observed.

3.2.2 Adjusting the regularizing sequence for SLOPE

In light of (3.3) we would like an expression for $X_i'X_{\mathcal{S}}(\beta_{\mathcal{S}} - \hat{\beta}_{\mathcal{S}})$, where with $\mathcal{S}$, $X_{\mathcal{S}}$ and $\beta_{\mathcal{S}}$ we indicate the support of β, the subset of variables associated to $\beta_i \neq 0$, and the value of their coefficients, respectively.

Again, to obtain a very rough evaluation of the SLOPE solution, we can start from the lasso. Let us assume that the size of $\beta_{\mathcal{S}}$ and the value of λ are such that the support and the signs of the regression coefficients are correctly recovered in the solution. That is, we assume that $\operatorname{sign}(\beta_j) = \operatorname{sign}(\hat{\beta}_j)$ for all j, with the convention that $\operatorname{sign}(0) = 0$. Without loss of generality, we further assume that $\beta_j \geq 0$. Now, the Karush–Kuhn-Tucker (KKT) optimality conditions for lasso yield

$$X_S'(y - X\hat{\beta}_S) = \lambda \cdot 1_S, \tag{3.4}$$

implying

$$\hat{\beta}_S = (X_S'X_S)^{-1}(X_S'y - \lambda \cdot 1_S).$$

In the case of SLOPE, rather than one λ, we have a sequence $\lambda_1, \ldots, \lambda_p$. Assuming again that this is chosen so that we recover exactly the support $\mathcal{S}$, the estimates of the nonzero components are very roughly equal to

$$\hat{\beta}_S = (X_{\mathcal{S}}'X_{\mathcal{S}})^{-1}(X_{\mathcal{S}}'y - \lambda_{\mathcal{S}}),$$

where $\lambda_{\mathcal{S}} = (\lambda_1, \ldots, \lambda_k)'$. This leads to

$$\mathbb{E}\, X_{\mathcal{S}}(\beta_{\mathcal{S}} - \hat{\beta}_{\mathcal{S}}) \approx X_{\mathcal{S}}(X_{\mathcal{S}}'X_{\mathcal{S}})^{-1}\lambda_{\mathcal{S}},$$

an expression that, plugged into v_i (3.3) tells us the typical size of $X_i'X_{\mathcal{S}}(X_{\mathcal{S}}'X_{\mathcal{S}})^{-1}\lambda_{\mathcal{S}}$.

For the case of Gaussian designs as in Figure 5, where the entries of X are i.i.d. $\mathcal{N}(0, 1/n)$, for $i \notin \mathcal{S}$,

$$\mathbb{E}(X_i'X_{\mathcal{S}}(X_{\mathcal{S}}'X_{\mathcal{S}})^{-1}\lambda_{\mathcal{S}})^2 = \frac{1}{n}\lambda_{\mathcal{S}}'\,\mathbb{E}(X_{\mathcal{S}}'X_{\mathcal{S}})^{-1}\lambda_{\mathcal{S}} = w(|\mathcal{S}|) \cdot \|\lambda_{\mathcal{S}}\|_{\ell_2}^2, \quad w(k) = \frac{1}{n-k-1}. \tag{3.5}$$

This uses the fact that the expected value of an inverse $k \times k$ Wishart with n degrees of freedom is equal to $I_k/(n-k-1)$.

This suggests the sequence of λ's described below denoted by λ_{G} since it is motivated by Gaussian designs. We start with $\lambda_{\mathrm{G}}(1) = \lambda_{\mathrm{BH}}(1)$. At the next stage, however, we need to account for the slight increase in variance so that we do not want to use $\lambda_{\mathrm{BH}}(2)$ but rather

$$\lambda_{\mathrm{G}}(2) = \lambda_{\mathrm{BH}}(2)\sqrt{1 + w(1)\lambda_{\mathrm{G}}(1)^2}.$$

Continuing, this gives

$$\lambda_{\mathrm{G}}(i) = \lambda_{\mathrm{BH}}(i)\sqrt{1 + w(i-1)\sum_{j<i}\lambda_{\mathrm{G}}(j)^2}. \tag{3.6}$$

Figure 6 plots the adjusted values given by (3.6). As is clear, these new values yield a procedure that is more conservative than that based on λ_{BH}. It can be observed that the corrected sequence $\lambda_{\mathrm{G}}(i)$ may no longer be decreasing (as in the case where $n = p/2$ in the figure). It would not make sense to use such a sequence—note that SLOPE would no longer be convex—and letting $k^\star = k(n, p, q)$ be the location of the global minimum, we shall work with

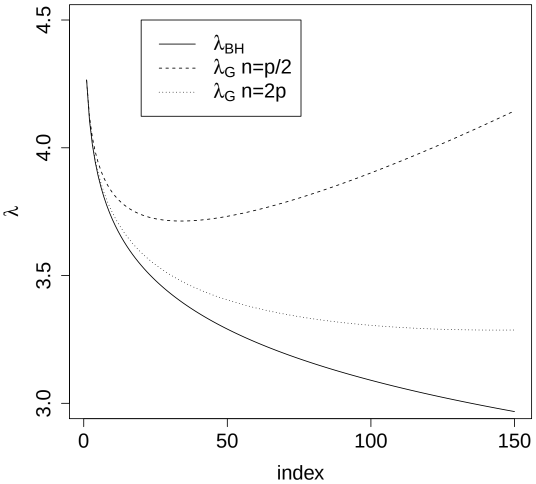
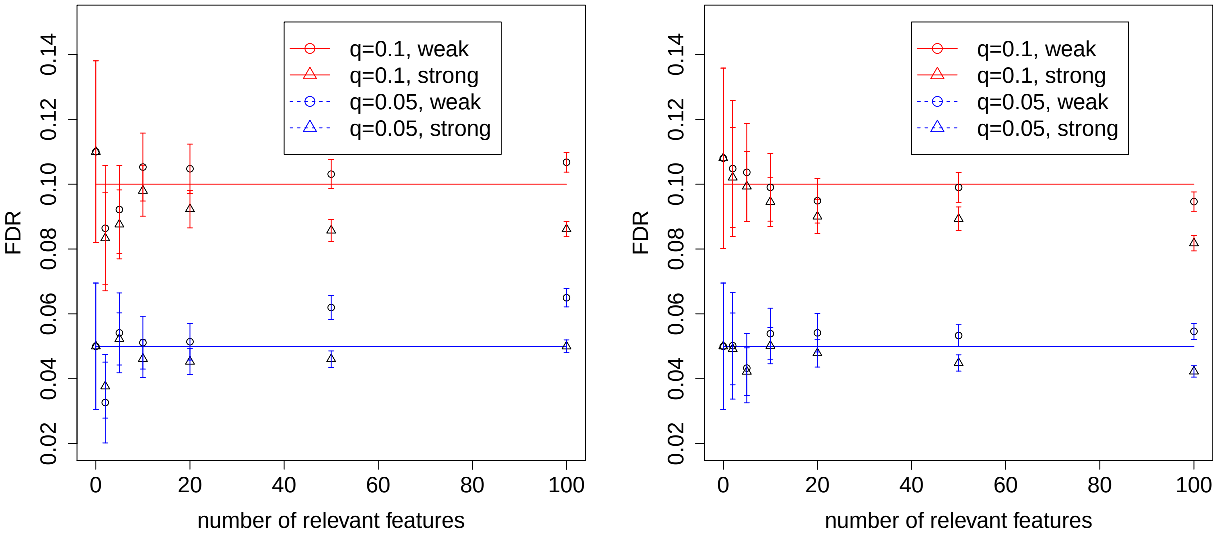


Figure 6: Graphical representation of sequences $\{\lambda_i\}$ for $n = 5000$ and $q = 0.1$. The solid line is λ_{BH}, the dashed (resp. dotted) line is λ_{G} given by (3.6) for $n = p/2$ (resp. $n = 2p$).

$$\lambda_{\mathrm{G}^\star}(i) = \begin{cases} \lambda_{\mathrm{G}}(i), & i \le k^\star, \\ \lambda_{k^\star}, & i > k^\star, \end{cases} \quad \text{with } \lambda_{\mathrm{G}}(i) \text{ as in (3.6).} \tag{3.7}$$

The value $k^\star$, starting from which we need to modify λ_{G} might play a role on the overall performance of the method.

An immediate validation—if the intuition that we have stretched this far has any bearing in reality—is the performance of $\lambda_{\mathrm{G}^\star}$ in the setup of Figure 5. In Figure 7 we illustrate the performance of SLOPE for large signals $\beta_i = 5\sqrt{2\log p}$ as in Figure 5, as well as for rather weak signals with $\beta_i = \sqrt{2\log p}$. The correction works very well, rectifying the loss of FDR control documented in Figure 5. For $p = 2n = 10,000$, the values of the critical point $k^\star$ are 51 for $q = 0.05$ and 68 for $q = 0.1$. For $p = n/2 = 2,500$, they become 95 and 147 respectively. It can be observed that for large signals, SLOPE keeps FDR below the nominal level even after passing the critical point. Interestingly, the control of FDR is more difficult when the coefficients have small amplitudes. We believe that some increase of FDR for weak signals is related to the loss of power, which our correction does not account for. However, even for weak signals the observed FDR of SLOPE with $\lambda_{\mathrm{G}^\star}$ is very close to the nominal level when $k \le k^\star$.



(a) $p = 2n = 10,000$.

(b) $p = n/2 = 2,500$.

Figure 7: Mean FDP ± 2SE for SLOPE with $\lambda_{G^\star}$. Strong signals have nonzero regression coefficients set to $5\sqrt{2\log p}$ while this value is set to $\sqrt{2\log p}$ for weak signals.

In situations where one cannot assume that the design is Gaussian or that columns are independent, we suggest replacing $w(i-1)\sum_{j<i}\lambda_j^2$ in the formula (3.6) with a Monte Carlo estimate of the correction. Let X denote the standardized version of the design matrix, so that each column has a mean equal to zero and unit variance. Suppose we have computed $\lambda_1, \ldots, \lambda_{i-1}$ and wish to compute λ_i. Let $X_{\mathcal{S}}$ indicate a matrix formed by selecting those columns with indices in some set $\mathcal{S}$ of cardinality $i-1$ and let $j \notin \mathcal{S}$. After randomly selecting $\mathcal{S}$ and j, the correction can be approximated by the average of $(X_j' X_{\mathcal{S}}(X_{\mathcal{S}}' X_{\mathcal{S}})^{-1}\lambda_{1:i-1})^2$ across realizations, where $\lambda_{1:i-1} = (\lambda_1, \ldots, \lambda_{i-1})'$. We will apply this strategy in the real-data example from genetics in the following section.

3.2.3 Unknown σ

According to formulas (1.5) and (1.10) the penalty in SLOPE depends on the standard deviation σ of the error term. In many applications σ is not known and needs to be estimated. When n is larger than p, this can easily be done by means of classical unbiased estimators. When $p \geq n$, some solutions for simultaneous estimation of σ and regression coefficients using ℓ_1 optimization schemes were proposed, see e.g. [40] and [41]. Specifically, [41] introduced a simple iterative version of the lasso called the *scaled lasso*. The idea of this algorithm can be applied to SLOPE, with some modifications. For one, our simulation results show that, under very sparse scenarios, it is better to de-bias the estimates of regression parameters by using classical least squares estimates within the selected model to obtain an estimate of σ^2.

We present our algorithm below. There, λ^S is the sequence of SLOPE parameters designed to work with $\sigma = 1$, obtained using the methods from Section 3.2.2.

Algorithm 5 Iterative SLOPE fitting when σ is unknown

```
input: $y$, $X$ and initial sequence $\lambda^S$ (computed for $\sigma = 1$)
initialize: $S_+ = \emptyset$
repeat
    $S = S_+$
    compute the RSS obtained by regressing $y$ onto variables in $S$
    set $\hat{\sigma}^2 = \text{RSS}/(n - |S| - 1)$
    compute the solution $\hat{\beta}$ to SLOPE with parameter sequence $\hat{\sigma} \cdot \lambda^S$
    set $S_+ = \text{supp}(\hat{\beta})$
until $S_+ = S$
```

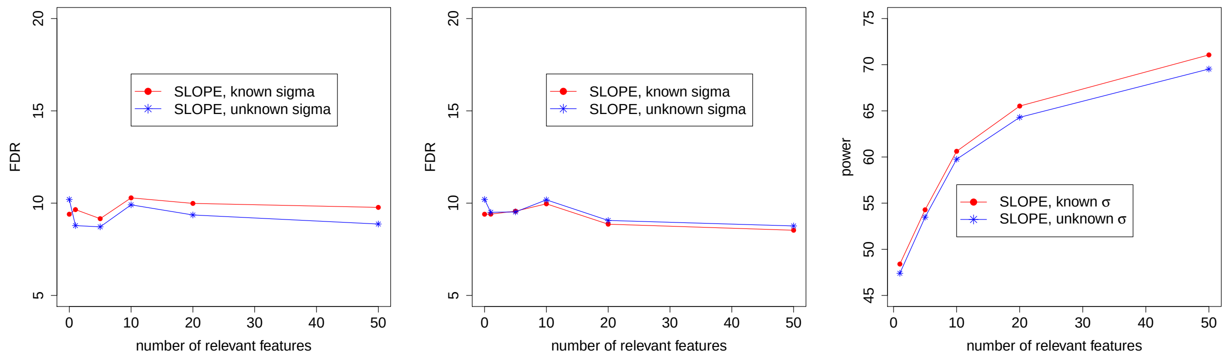
The procedure starts by using a conservative estimate of the standard deviation of the error term $\hat{\sigma}^{(0)} = \text{Std}(y)$ and a related conservative version of SLOPE with $\lambda^{(0)} = \hat{\sigma}^{(0)} \cdot \lambda^S$. Then, in consecutive runs $\hat{\sigma}^{(k)}$ is computed using residuals from the regression model, which includes variables identified by SLOPE with sequence $\sigma^{(k-1)} \cdot \lambda^S$. The procedure is repeated until convergence, i.e. until the next iteration results in exactly the same model as the current one.

To verify the performance of this algorithm we conducted a series of experiments under sparse regression models. Specifically, Figure 8 compares the performance of the 'scaled SLOPE' with the case of known σ. The $5,000 \times 5,000$ design matrix has i.i.d. $\mathcal{N}(0, 1/n)$ entries, the noise level is set to $\sigma = 1$ and the size of the nonzero regression coefficients is equal to $\sqrt{2 \log p} \approx 4.13$ ('weak' signal) and $5\sqrt{2 \log p} = 20.64$. ('strong' signal). We work with $\lambda^S = \lambda_{\text{G}^\star}$ given by the formula (3.7) for $q = 0.1$.

In our simulations, the proposed algorithm converges very quickly. The conservative initial estimate of σ leads to a relatively small model with few false discoveries since $\sigma^{(0)} \cdot \lambda^S$ controls the FDR in sparse scenarios. Typically, iterations to convergence see the estimated value of σ decrease and the number of selected variable increase. When the signals are weak, σ remains slightly overestimated as the residual error is inflated by some undetected signals. This translates into controlling the FDR at a level slightly below the nominal one (Figure 8a) with, however, only a minor decrease in power (Figure 8c). For strong signals, which are always detected in our experiments, σ is slightly underestimated due to the selection of a small number of false regressors. Figure 8b shows that for larger k the FDR of the scaled version very slightly exceeds that of the version operating with a known σ, remaining below the nominal level for all k in the considered range.

3.3 An example from genetics

In this section we illustrate the application of SLOPE to a current problem in genetics. In [39], the authors investigate the role of genetic variants in 17 regions in the genome, selected on the basis of previously reported association with traits related to cardiovascular health. Polymorphisms are identified via exome re-sequencing in approximately 6,000 individuals of Finnish descent: this provides a comprehensive survey of the genetic diversity in the coding portions of these regions and affords the opportunity to investigate which of these variants have an effect on the traits of interest. While the original study has a broader scope, we here tackle the problem of identifying which genetic variants in these regions impact the fasting blood HDL levels. Previous literature reported associations between 9 of the 17 regions and HDL, but the resolution of these earlier



(a) FDR, $\beta_i \approx 4.13$

(b) FDR, $\beta_i \approx 20.64$

(c) Power, $\beta_i \approx 4.13$

Figure 8: FDR and power of two versions of SLOPE with $q = 0.1$ under a $5,000 \times 5,000$ Gaussian design matrix. The noise level σ is equal to 1. In each replicate, the signals are randomly placed over the columns of the design matrix, and the plotted data points are averages over 500 replicates. The power is reported only for weak signals ($\beta_i \approx 4.13$) since for strong signals ($\beta_i \approx 20.64$) it is always equal to 1.

studies was unable to pinpoint to specific variants in these regions or to distinguish if only one or multiple variants within the regions impact HDL. The resequencing study was designed to address this problem.

The analysis in [39] relies substantially on "marginal" tests: the effect of each variant on HDL is examined via a linear regression that has cholesterol level as outcome and the genotype of the variant as explanatory variable, together with covariates that capture possible population stratification. Such marginal tests are common in genetics and represent the standard approach in genome-wide association studies (GWAS). Among their advantages it is worth mentioning that they allow to use all available observations for each variant without requiring imputation of missing data; their computational cost is minimal; and they result in a p-value for each variant that can be used to clearly communicate to the scientific community the strength of the evidence in favor of its impact on a particular trait. Marginal tests, however, cannot distinguish if the association between a variant and a phenotype is "direct" or due to correlation between the variant in question and another, truly linked to the phenotype. Since most of the correlation between genetic variants is due to their location along the genome (with near-by variants often correlated), this confounding is often considered not too serious a limitation in GWAS: multiple polymorphisms associated to a phenotype in one locus simply indicate that there is at least one genetic variant (most likely not measured in the study) with impact on the phenotype in the locus. The situation is quite different in the re-sequencing study we want to analyze, where establishing if one or more variants in the same region influence HDL is one of the goals. To address this, the authors of [39] resort to regressions that include two variables at the time: one of these being the variant with previously documented strongest marginal signal in the region, the other variants that passed an FDR controlling threshold in the single variant analysis. Model selection strategies were only cursory explored with a step-wise search routine that targets BIC. Such limited foray into model selection is motivated by the fact that one major concern in genetics is to control some global measure of type one error and currently available model selection strategies do not offer finite sample guarantees with this regard.

This goal is in line with that of SLOPE and so it is interesting for us to apply this new procedure to this problem.

The dataset in [39] comprises 1,878 variants, on 6,121 subjects. Before analyzing it with SLOPE, or other model selection tools, we performed the following filtering. We eliminated from considerations variants observed only once (a total of 486), since it would not be possible to make inference on their effect without strong assumptions. We examined correlation between variants and selected for analysis a set of variants with pair-wise correlation smaller than 0.3: larger values would make it quite challenging to interpret the outcomes; they render difficult the comparison of results across procedures since these might select different variables from a group of correlated ones; and large correlations are likely to adversely impact the efficacy of any model selection procedure. This reduction was carried out in an iterative fashion, selecting representative from groups of correlated variables, starting from stronger levels of correlation and moving onto lower ones. Among correlated variables, we selected those that had stronger univariate association with HDL, larger minor allele frequency (diversity), and, among very rare variants we privileged those whose annotation was more indicative of possible functional effects. Once variables were identified, we eliminated subjects that were missing values for more than 10 variants, and for HDL. The remaining missing values were imputed using the average allele count per variant. This resulted in a design with 5,375 subjects and 777 variants. The minor allele frequency of the variants included ranges from 2×10^{-4} to 0.5, with a median of 0.001 and a mean of 0.028: the data set still includes a number of rare variants.

In [39], association between HDL and polymorphisms was analyzed only for variants in regions previously identified as having an influence on HDL: *ABCA1, APOA1, CEPT, FADS1, GALNT2, LIPC, LPL, MADD*, and *MVK* (regions are identified with the name of one of the genes they contain). Moreover, only variants with minor allele frequencies larger than 0.01 were individually investigated, while non synonimous rare variants were analyzed with "burden tests." These restrictions were motivated, at least in part, by the desire to reduce tests to the most well powered ones, so that controlling for multiple comparisons would not translate in an excessive decrease of power. Our analysis is based on all variants that survive the described filtering in all regions, including those not directly sequenced in the experiment in [39], but included in the study as landmarks of previously documented associations (*array SNPs* in the terminology of the paper). We compare the following approaches: the (1) marginal tests described above in conjunction with BH and $q = 0.05$; (2) BH and $q = 0.05$ applied to the p-values from the full model regression; (3) lasso with λ_{Bonf} and $\alpha = 0.05$; (4) lasso with λ_{CV} (in these last two cases we use the routines implemented in `glmnet` in R); (5) the R routine Step.AIC in forward direction and BIC as optimality criteria; (6) the R routine Step.AIC in backwards direction and BIC as optimality criteria; (7) SLOPE with $\lambda_{\text{G}^\star}$ and $q = 0.05$; (8) SLOPE with λ obtained via Monte Carlo starting from our design matrix. Defining the λ for lasso–λ_{Bonf} and SLOPE requires a knowledge of the noise level σ^2: we estimated this from the residuals of the full model. When estimating λ via the Monte Carlo approach, for each i we used 5,000 independent random draws of $X_{\mathcal{S}}$ and X_j. Figure 9a illustrates how the Monte Carlo sequence λ_{MC} is larger than $\lambda_{\text{G}^\star}$: this difference increases with the index i, and become substantial for ranges of i that are unlikely to be relevant in the scientific problem at hand.

Tables 1 and 2 in [39] describe a total of 14 variants as having an effect on HDL: two of these are for regions *FADS1* and *MVK* and the strength of the evidence in this specific dataset is quite weak (a marginal p-value of the order of 10^{-3}). Multiple effects are identified in regions *ABCA1, CEPT, LPL*, and *LIPL.* The results of the various "model selection" strategies we explored are in Figure 10, which reports the estimated values of the coefficients. The effect of the shrinkage induced by lasso and SLOPE are evident: to properly compare effect sizes across methods it would be useful to resort to the two-step procedure that we used for the simulation described in Figure 2. Since our interest here is purely model selection, we report the coefficients directly as estimated by the ℓ_1 penalized procedures: this has the welcome side effect of increasing the spread of points in Figure 10, improving visibility.

Of the 14 variants described in [39], 8 are selected by all methods. The remaining 6 are all selected by at least some of the 8 methods we compared. There are additional 5 variants that are selected by all methods but are not in the main list of findings in the original paper: four of these are rare variants, and one is an *array SNP* for a trait other than HDL. While none of these, therefore, was singularly analyzed for association in [39], they are in highlighted regions: one is in *MADD*, and the others in *ABCA1* and *CETP*, where the paper documents a plurality of signals.

Besides this core of common selections that correspond well to the original findings, there are notable differences among the 8 approaches we considered. The total number of selected variables ranges from 15—with BH on the p-values of the full model—to 119 with the cross-validated lasso. It is not surprising that these methods would result in the extreme solutions: on the one hand, the p-values from the full model reflect the contribution of one variable given all the others, which are, however, not necessarily included in the models selected by other approaches; on the other hand, we have seen how the cross-validated lasso tends to select a much larger number of variables and offers no control of FDR. In our case, the cross-validated lasso estimates nonzero coefficients for 90 variables that are not selected by any other methods. Note that the number of variables selected by the cross-validated lasso changes in different runs of the procedure, as implemented in `glmnet` with default parameters. It is quite reasonable to assume that a large number of these are false positives: regions *G6PC2, PANK1, CRY2, MTNR1B*, where the lasso-λ_{CV} selects some variants, have no documented association with lipid levels, and regions *CELSR2, GCKR, ABCG8*, and *NCAN* have been associated previously to total cholesterol and LDL, but not HDL. The other procedures that select some variants in any of these regions are the forward and backward greedy searches trying to optimize BIC, which have hits in *CELSR2* and *ABCG8*, and the BH on univariate p-value, which has one hit in *ABCG8.* SLOPE does not select any variant in regions not known to be associated with HDL. This is true also of the lasso-λ_{Bonf} and BH on the p-values from the full model, but these miss respectively 2 and 6 of the variants described in the original paper, while SLOPE $\lambda_{\text{G}^\star}$ misses only one of them.

Figure 11 focuses on the set of variants where there is some disagreement between the 8 procedures we considered, after eliminating the 90 variants selected only by the lasso-λ_{CV}. In addition to recovering all except one of the variants identified in [39], and to the core of variants selected by all methods, SLOPE-$\lambda_{\text{G}^\star}$ selects 3 rare variants and 3 common variants. While the rare variants were not singularly analyzed in the original study, they are in the two regions where aggregate tests highlighted the role of this type of variation. One is in *ABCA1* and the other two are in *CETP*, and they are both non-synonimous. Two of the three additional common variants are in *CETP* and one is in *MADD*: in addition to SLOPE, these are selected by lasso–λ_{CV} and the marginal tests. One of the common variants and one rare variant in *CETP* are mentioned as a result of the limited foray in model selection in [39]. SLOPE-λ_{MC} selects two less of these variants.

In order to get a handle on the effective FDR control of SLOPE in this setting, we resorted to simulations. We consider a number k of relevant variants ranging from 0 to 100, while concentrating on lower values. At each level, k columns of the design matrix were selected at random and assigned an effect of $\sqrt{2 \log p}$ against a noise level σ set to 1. While analyzing the data with λ_{MC} and $\lambda_{\mathrm{G}^\star}$, we estimated σ from the full model in each run. Figure 9(b-c) reports the average FDP across 500 replicates and their standard error: the FDR of both λ_{MC} and $\lambda_{\mathrm{G}^\star}$ are close to the nominal levels for all $k \leq 100$.

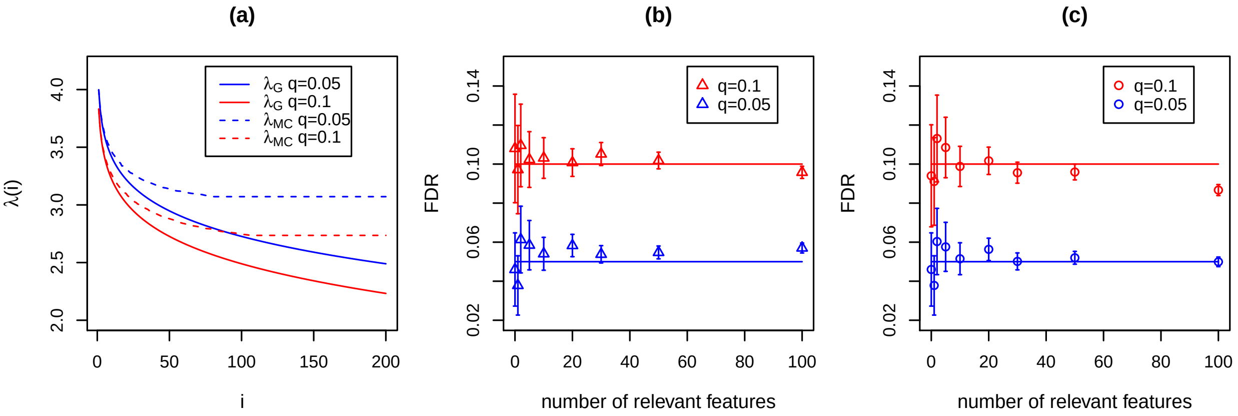


Figure 9: (a) Graphical representation of sequences λ_{MC} and λ_{G} for the variants design matrix. Mean FDP ± 2SE for SLOPE with (b) $\lambda_{\mathrm{G}^\star}$ and (c) λ_{MC} for the variants design matrix and $\beta_1 = \ldots = \beta_k = \sqrt{2 \log p} \approx 3.65$, $\sigma = 1$.

In conclusion, the analysis with SLOPE confirms the results in [39], does not appear to introduce a large number of false positives, and hence it makes it easier to include in the final list of relevant variants a number of polymorphisms that are either directly highlighted in the original paper or in regions that were described as including a plurality of signals, but for which the original multi-step analysis did not allow to make a precise statement.

4 Discussion

The ease with which data are presently acquired has effectively created a new scientific paradigm: in addition to carefully designing experiments to test specific hypotheses, researchers often collect data first, leaving question formulation to a later stage. In this context, linear regression has increasingly been used to identify connections between one response and a large number p of possible explanatory variables. When $p \gg n$, approaches based on convex optimization have been particularly effective: an easily computable solution has the advantage of definitiveness and of reproducibility—another researcher, working on the same dataset, would obtain the same answer. Reproducibility of a scientific finding, or of the association between the outcome and the set of explanatory variables selected among many, however, is harder to achieve. Traditional tools as p-values are often unhelpful in this context because of the difficulties of accounting for the effect of selection. Indeed, the last few years have witnessed a substantive push towards the developing of an inferential framework after selection [11, 12, 34, 21, 29, 18, 31], with the exploration of quite different view-points. We here chose as a useful paradigm that of controlling the expected proportion of irrelevant variables among the selected ones. A similar goal of FDR control is pursued in [24, 28]. While [24] achieves exact FDR control in finite sample irrespective of the structure of the design matrix, this method, at least in the current implementation, is really best tailored for cases where $n > p$. The work in [28] relies on p-values evaluated as in [31], and its limited to the contexts where the assumptions in [31] are met. SLOPE controls FDR under orthogonal designs, and simulation studies also show that SLOPE can keep the FDR close to the nominal level when $p > n$ and the true model is sparse, while offering large power and accurate prediction. This is, of course, only a starting point and many open problems remain.

Firstly, while our heuristics for the choice of the λ sequence allows to keep FDR under control for Gaussian designs and other random design matrices (more examples are provided in [17]), it is by no means a definite solution. Further theoretical research is needed to identify the sequences λ, which would provably control FDR for these designs and other typical design matrices.

Second, just as in the BHq procedure where the test statistics are compared with fixed critical values, we have only considered in this paper fixed values of the regularizing sequence $\{\lambda_i\}$. It would be interesting to know whether it is possible to select such parameters in a data-driven fashion as to achieve desirable statistical properties. For the simpler lasso problem for instance, an important question is whether it is possible to select λ on the lasso path as to control the FDR. In the case where $n \geq p$ a method to obtain this goal was recently proposed in [24]. It would be of great interest to know if similar positive theoretical results can be obtained for SLOPE, in perhaps restricted sparse settings.

In conclusion, we hope that the work presented so far would convince the reader that SLOPE is an interesting convex program with promising applications in statistics and motivate further research.

Acknowledgements

E. C. is partially supported by AFOSR under grant FA9550-09-1-0643, by ONR under grant N00014-09-1-0258 and by a gift from the Broadcom Foundation. M. B. was supported by the Fulbright Scholarship, NSF grant NSF 1043204 and the European Unions 7th Framework Programme for research, technological development and demonstration under Grant Agreement no 602552. E. v.d.B. was supported by National Science Foundation Grant DMS 0906812 (American Reinvestment and Recovery Act). C. S. is supported by NIH grants HG006695 and MH101782. W. S. is supported by a General Wang Yaowu Stanford Graduate Fellowship. We thank the authors of [39] for letting us use their data during the completion of dbGaP release. E. C. would like to thank Stephen Becker for all his help in integrating the sorted ℓ_1 norm software into TFOCS. M. B. would like to thank David Donoho and David Siegmund for encouragement and Hatef Monajemi for helpful discussions. We are very grateful to Lucas Janson for suggesting the acronym SLOPE, and to Rina Foygel Barber and Julie Josse for useful comments about an early version of the manuscript.

References

[1] F. Abramovich and Y. Benjamini. Thresholding of wavelet coefficients as multiple hypotheses testing procedure. In *In Wavelets and Statistics, Lecture Notes in Statistics 103, Antoniadis*, pages 5–14. Springer-Verlag, 1995.

[2] F. Abramovich, Y. Benjamini, D. L. Donoho, and I. M. Johnstone. Adapting to unknown sparsity by controlling the false discovery rate. *Ann. Statist.*, 34(2):584–653, 2006.

[3] H. Akaike. A new look at the statistical model identification. *IEEE Trans. Automatic Control*, AC-19:716–723, 1974. System identification and time-series analysis.

[4] R. E. Barlow, D. J. Bartholomew, J-M. Bremner, and H.D. Brunk. *Statistical inference under order restrictions: The theory and application of isotonic regression.* Wiley New York, 1972.

[5] P. Bauer, B. M. Pötscher, and P. Hackl. Model selection by multiple test procedures. *Statistics*, 19:39–44, 1988.

[6] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2:183–202, March 2009.

[7] S. Becker, E. J. Candès, and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3):165–218, August 2011.

[8] Y. Benjamini and Y. Gavrilov. A simple forward selection procedure based on false discovery rate control. *Ann. Appl. Stat.*, 3(1):179–198, 2009.

[9] Y. Benjamini and Y. Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[10] Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Ann. Statist.*, 29(4):1165–1188, 2001.

[11] Y. Benjamini and D. Yekutieli. False discovery rate-adjusted multiple confidence intervals for selected parameters. *J. Amer. Statist. Assoc.*, 100(469):71–93, 2005. With comments and a rejoinder by the authors.

[12] R. Berk, L. Brown, A. Buja, K. Zhang, and L. Zhao. Valid post-selection inference. *Ann. Statist.*, 41(2):802–837, 2013.

[13] M. J. Best and N. Chakravarti. Active set algorithms for isotonic regression; a unifying framework. *Mathematical Programming*, 47(1-3):425–439, 1990.

[14] L. Birgé and P. Massart. Gaussian model selection. *J. Eur. Math. Soc. (JEMS)*, 3(3):203–268, 2001.

[15] M. Bogdan, A. Chakrabarti, F. Frommlet, and J. K. Ghosh. Asymptotic Bayes optimality under sparsity of some multiple testing procedures. *Annals of Statistics*, 39:1551–1579, 2011.

[16] M. Bogdan, J. K. Ghosh, and M. Żak-Szatkowska. Selecting explanatory variables with the modified version of bayesian information criterion. *Quality and Reliability Engineering International*, 24:627–641, 2008.

[17] M. Bogdan, E. van den Berg, W. Su, and E. J. Candès. Statistical estimation and testing via the ordered ℓ_1 norm. arXiv:1310.1969v2, 2013.

[18] P. Bühlmann. Statistical significance in high-dimensional linear models. *Bernoulli*, 19(4):1212–1242, 2013.

[19] E. J. Candès and T. Tao. The Dantzig Selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

[20] E. J. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted l1 minimization. *J. Fourier Anal. Appl.*, 14:877–905, 2008.

[21] B. Efron. Tweedie's formula and selection bias. *J. Amer. Statist. Assoc.*, 106(496):1602–1614, 2011.

[22] D. P. Foster and E. I. George. The risk inflation criterion for multiple regression. *Ann. Statist.*, 22(4):1947–1975, 1994.

[23] D. P. Foster and R. A. Stine. Local asymptotic coding and the minimum description length. *IEEE Transactions on Information Theory*, 45(4):1289–1293, 1999.

[24] R. Foygel-Barber and E. J. Candès. Controlling the false discovery rate via knockoffs. arXiv:1404.5609, 2014.

[25] F. Frommlet and M. Bogdan. Some optimality properties of FDR controlling rules under sparsity. *Electronic Journal of Statistics*, 7:1328–1368, 2013.

[26] F. Frommlet, F. Ruhaltinger, P. Twaróg, and M. Bogdan. A model selection approach to genome wide association studies. *Computational Statistics & Data Analysis*, 56:1038–1051, 2012.

[27] S.J. Grotzinger and C. Witzgall. Projections onto order simplexes. *Applied Mathematics and Optimization*, 12(1):247–270, 1984.

[28] M. Grazier G'Sell, T. Hastie, and R. Tibshirani. False variable selection rates in regression. *arXiv:1302.2303*, 2013.

[29] A. Javanmard and A. Montanari. Confidence Intervals and Hypothesis Testing for High-Dimensional Regression. *ArXiv e-prints*, June 2013.

[30] J. B. Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2):115–129, 1964.

[31] R. Lockhart, J. Taylor, R. Tibshirani, and R. Tibshirani. A significance test for the lasso. To appear in the *Annals of Statistics*, 2013.

[32] P. Mair, K. Hornik, and J. de Leeuw. Isotone optimization in R: pool-adjacent-violators algorithm (PAVA) and active set methods. *Journal of statistical software*, 32(5):1–24, 2009.

[33] C. L. Mallows. Some comments on c_p. *Technometrics*, 15(2):661–676, 1973.

[34] N. Meinshausen and P. Bühlmann. Stability selection. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 72(4):417–473, 2010.

[35] Y. Nesterov. *Introductory lectures on convex optimization: A basic course.* Kluwer Academic Publishers, 2004.

[36] Y. Nesterov. Gradient methods for minimizing composite objective function. `http://www.ecore.be/DPs/dp_1191313936.pdf`, 2007. CORE discussion paper.

[37] N. Parikh and S. Boyd. Proximal algorithms. In *Foundations and Trends in Optimization*, volume 1, pages 123–231. 2013.

[38] S. K. Sarkar. Some results on false discovery rate in stepwise multiple testing procedures. *Annals of Statistics*, pages 239–257, 2002.

[39] S. K. Service, T. M. Teslovich, C. Fuchsberger, V. Ramensky, P. Yajnik, D. C. Koboldt, D. E. Larson, Q. Zhang, L. Lin, R. Welch, L. Ding, M. D. McLellan, M. O'Laughlin, C. Fronick, L. L. Fulton, V. Magrini, A. Swift, P. Elliott, M. R. Jarvelin, M. Kaakinen, M. I. McCarthy, L. Peltonen, A. Pouta, L. L. Bonnycastle, F. S. Collins, N. Narisu, H. M. Stringham, J. Tuomilehto, S. Ripatti, R. S. Fulton, C. Sabatti, R. K. Wilson, M. Boehnke, and N. B. Freimer. Re-sequencing expands our understanding of the phenotypic impact of variants at GWAS loci. *PLoS Genet.*, 10(1):e1004147, Jan 2014.

[40] N. Städler, P. Bühlmann, and S. van de Geer. ℓ_1-penalization for mixture regression models (with discussion). *Test*, 19:209–285, 2010.

[41] T. Sun and C.-H. Zhang. Scaled sparse linear regression. *Biometrika*, 99(4):879–898, 2012.

[42] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, February 1996.

[43] R. Tibshirani and K. Knight. The covariance inflation criterion for adaptive model selection. *J. Roy. Statist. Soc. B*, 55:757–796, 1999.

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

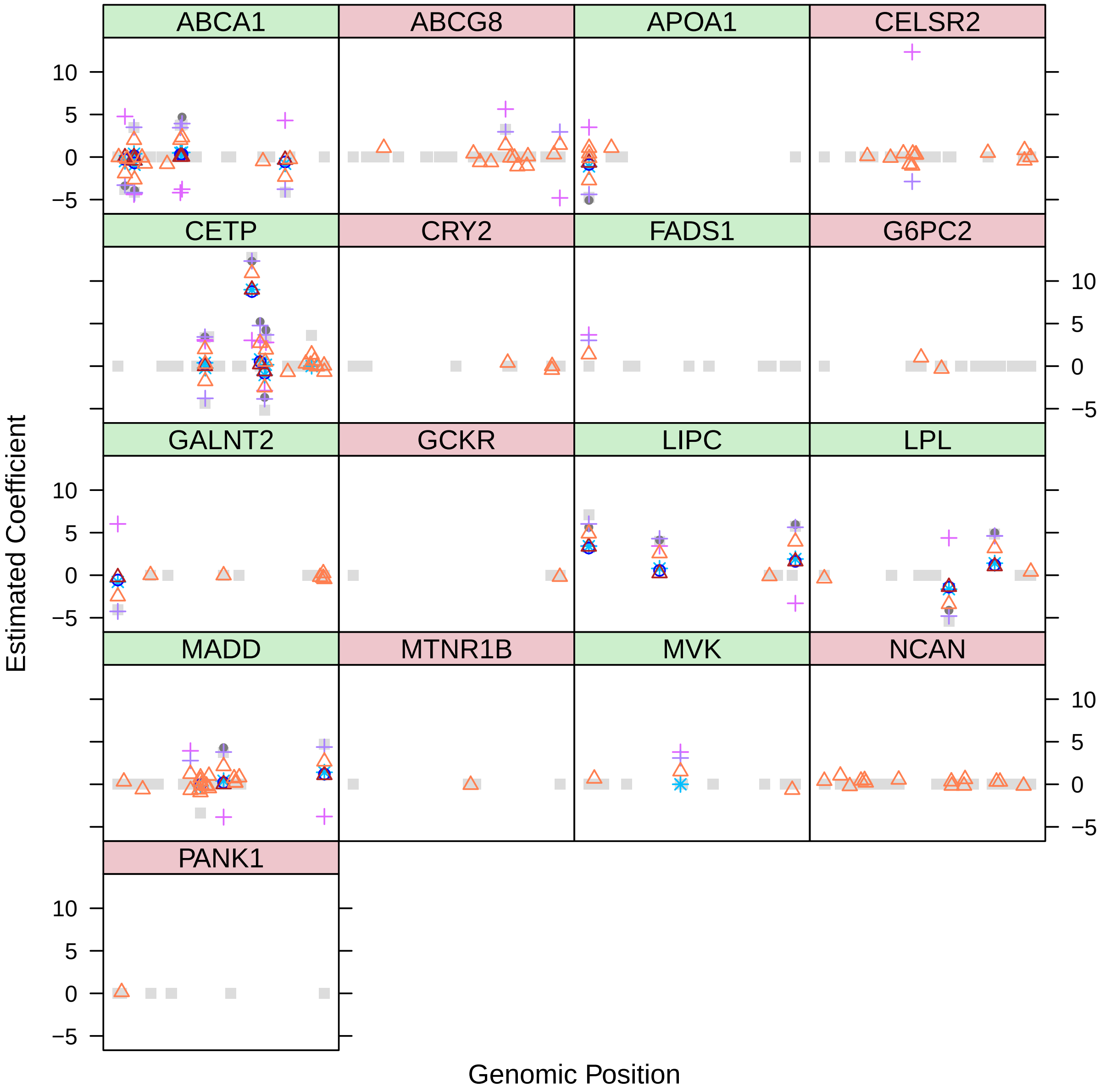


Figure 10: Estimated effects on HDL for variants in 17 regions. Each panel corresponds to a region and is identified by the name of a gene in the region, following the convention in [39]. Regions with (without) previously reported association to HDL are on green (red) background. On the x-axis variants position in base-pairs along their respective chromosomes. On the y-axis estimated effect according to different methodologies. With the exception of marginal tests—which we use to convey information on the number of variables and indicated with light gray squares—we report only the value of non zero coefficients. The rest of the plotting symbols and color convention is as follows: dark gray bullet—BH on p-values from full model; magenta cross—forward BIC; purple cross—backwards BIC; red triangle—lasso–λ_{Bonf}; orange triangle—lasso–λ_{CV}; cyan star—SLOPE–$\lambda_{\text{G}^\star}$; black circle—SLOPE with λ defined with Monte Carlo strategy.

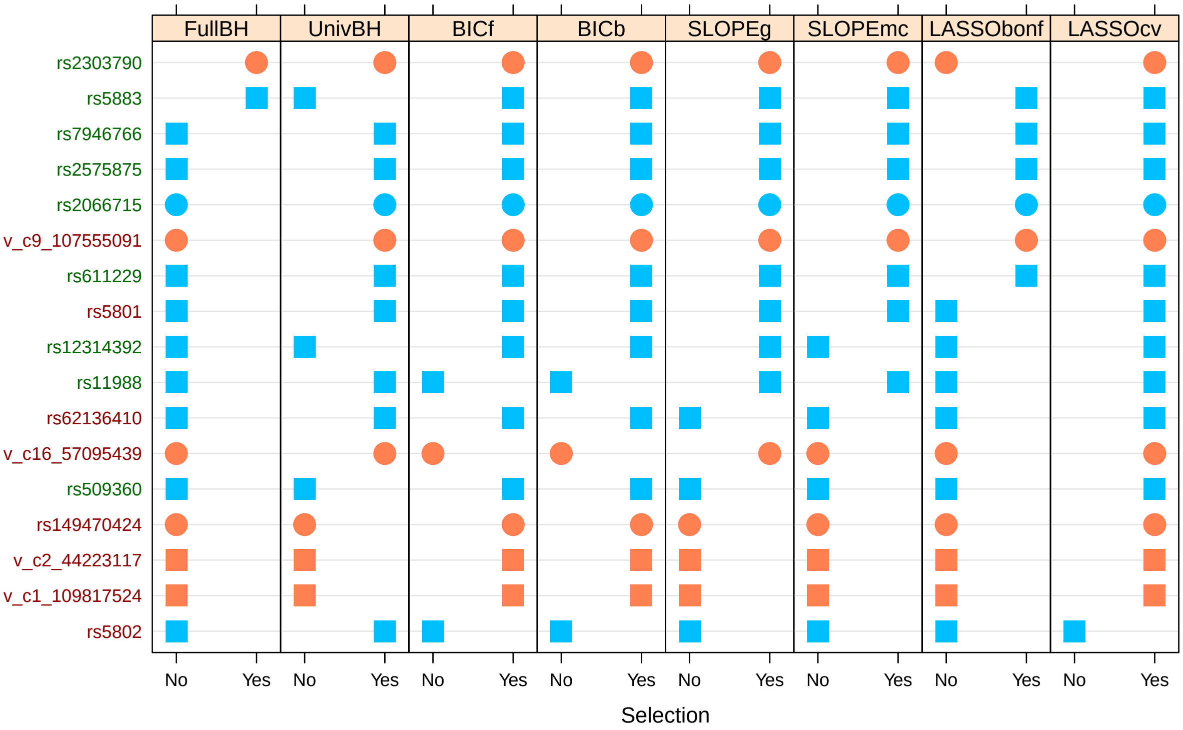


Figure 11: Each row corresponds to a variant in the set differently selected by the the compared procedures, indicated by columns. Orange is used to represent rare variants and blue common ones. Squares indicate synonymous (or non coding variants) and circle non-synonimous ones. Variants are ordered according to the frequency with which they are selected. Variants with names in green are mentioned in [39] as to have an effect on LDL, while variants with names in red are not (if a variant was not in dbSNP build 137, we named it by indicating chromosome and position, following the convention in [39]).

A FDR Control Under Orthogonal Designs

In this section, we prove FDR control in the orthogonal design, namely, Theorem 1.1. As we have seen in Section 1, the SLOPE solution reduces to

$$\min_{b\in\mathbb{R}^p} \tfrac{1}{2}\|\tilde{y}-b\|_{\ell_2}^2 + \sum_{i=1}^p \lambda_i |b|_{(i)},$$

where $\tilde{y} = X'y \sim \mathcal{N}(\beta, I_p)$. From this, it is clear that it suffices to consider the setting in which $y \sim \mathcal{N}(\beta, I_n)$, which we assume from now on.

We are thus testing the n hypotheses $H_i : \beta_i = 0$, $i = 1,\dots,n$ and set things up so that the first n_0 hypotheses are null, i.e. $\beta_i = 0$ for $i \le n_0$. The SLOPE solution is

$$\hat{\beta} = \arg\min \tfrac{1}{2}\|y-b\|_{\ell_2}^2 + \sum_{i=1}^n \lambda_i |b|_{(i)} \tag{A.1}$$

with $\lambda_i = \Phi^{-1}(1-iq/2n)$. We reject H_i if and only if $\hat{\beta}_i \neq 0$. Letting V (resp. R) be the number of false rejections (resp. the number of rejections) or, equivalently, the number of indices in $\{1,\dots,n_0\}$ (resp. in $\{1,\dots,n\}$) for which $\hat{\beta}_i \neq 0$, we have

$$\mathrm{FDR} = \mathbb{E}\left[\frac{V}{R\vee 1}\right] = \sum_{r=1}^n \mathbb{E}\left[\frac{V}{r}\mathbb{1}_{\{R=r\}}\right] = \sum_{r=1}^n \frac{1}{r}\,\mathbb{E}\left[\sum_{i=1}^{n_0} \mathbb{1}_{\{H_i \text{ is rejected}\}}\mathbb{1}_{\{R=r\}}\right]. \tag{A.2}$$

The proof of Theorem 1.1 now follows from the two key lemmas below.

Lemma A.1. *Let H_i be a null hypothesis and let $r \ge 1$. Then*

$$\{y\colon\ H_i \text{ is rejected and } R = r\} = \{y\colon\ |y_i| > \lambda_r \text{ and } R = r\}.$$

Lemma A.2. *Consider applying the* SLOPE *procedure to $\tilde{y} = (y_1,\dots,y_{i-1},y_{i+1},\dots,y_n)$ with weights $\tilde{\lambda} = (\lambda_2,\dots,\lambda_n)$ and let $\tilde{R}$ be the number of rejections this procedure makes. Then with $r \ge 1$,*

$$\{y\colon\ |y_i| > \lambda_r \text{ and } R = r\} \subset \{y : |y_i| > \lambda_r \text{ and } \tilde{R} = r-1\}.$$

To see why these intermediate results give Theorem 1.1, observe that

$$\begin{aligned}\mathbb{P}(H_i \text{ rejected and } R = r) &\le \mathbb{P}(|y_i| \ge \lambda_r \text{ and } \tilde{R} = r-1)\\ &= \mathbb{P}(|y_i| \ge \lambda_r)\,\mathbb{P}(\tilde{R} = r-1)\\ &= \frac{qr}{n}\,\mathbb{P}(\tilde{R} = r-1),\end{aligned}$$

where the inequality is a consequence of the lemmas above and the first equality follows from the independence between y_i and $\tilde{y}$. Plugging this inequality into (A.2) gives

$$\mathrm{FDR} = \sum_{r=1}^n \frac{1}{r}\sum_{i=1}^{n_0} \mathbb{P}(H_i \text{ rejected and } R = r) \le \sum_{r\ge 1} \frac{qn_0}{n}\,\mathbb{P}(\tilde{R} = r-1) = \frac{qn_0}{n},$$

which finishes the proof.

A.1 Proof of Lemma A.1

We begin with a lemma we shall use more than once.

Lemma A.3. *Consider a pair of nonincreasing and nonnegative sequences $y_1 \geq y_2 \geq \ldots \geq y_n \geq 0$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$, and let $\hat{b}$ be the solution to*

$$\begin{array}{ll} \text{minimize} & f(b) = \frac{1}{2}\|y-b\|_{\ell_2}^2 + \sum_{i=1}^n \lambda_i b_i \\ \text{subject to} & b_1 \geq b_2 \geq \ldots \geq b_n \geq 0. \end{array}$$

If $\hat{b}_r > 0$ and $\hat{b}_{r+1} = 0$, then for every $j \leq r$, it holds that

$$\sum_{i=j}^{r} (y_i - \lambda_i) > 0 \tag{A.3}$$

and for every $j \geq r+1$,

$$\sum_{i=r+1}^{j} (y_i - \lambda_i) \leq 0. \tag{A.4}$$

Proof To prove (A.3), consider a new feasible sequence b, which differs from $\hat{b}$ only by subtracting a small positive scalar $h < \hat{b}_r$ from $\hat{b}_j, \ldots, \hat{b}_r$. Now

$$f(b) - f(\hat{b}) = h \sum_{i=j}^{r} (y_i - \lambda_i - \hat{b}_i) + h^2 \sum_{i=j}^{r} \tfrac{1}{2}.$$

Taking the limit as h goes to zero, the optimality of $\hat{b}$ implies that $\sum_{i=j}^r (y_i - \lambda_i - \hat{b}_i) \geq 0$, which gives

$$\sum_{i=j}^{r} (y_i - \lambda_i) \geq \sum_{i=j}^{r} \hat{b}_i > 0.$$

For the second claim (A.4), consider a new sequence b, which differs from $\hat{b}$ by replacing $\hat{b}_{r+1}, \ldots, \hat{b}_j$ with a positive scalar $0 < h < \hat{b}_r$. Now observe that

$$f(b) - f(\hat{b}) = -h \sum_{i=r+1}^{j} (y_i - \lambda_i) + h^2 \sum_{i=r+1}^{j} \tfrac{1}{2}.$$

The claim follows from the optimality of $\hat{b}$. ■

It is now straightforward so see how these simple relationships give Lemma A.1. Observe that when $R = r$, we must have $|y|_{(r)} > \lambda_r$ and $|y|_{(r+1)} \leq \lambda_{r+1}$. Hence, if H_1 is rejected, it must hold that $|y_1| \geq |y|_{(r)} > \lambda_r$. This shows that $\{H_1 \text{ is rejected and } R = r\} \subset \{|y_1| > \lambda_r \text{ and } R = r\}$. Conversely, assume that $|y_1| > \lambda_r$ and $R = r$. Then H_1 must be rejected since $|y_1| > |y|_{(r+1)}$. This shows that $\{H_1 \text{ is rejected and } R = r\} \supset \{|y_1| > \lambda_r \text{ and } R = r\}$.

A.2 Proof of Lemma A.2

We assume without loss of generality that $y \geq 0$ (the extension to arbitrary signs is trivial). By assumption the solution to (A.1) with $\lambda_i = \Phi^{-1}(1 - iq/2n)$ has exactly r strictly positive entries, and we need to show that when y_1 is rejected, the solution to

$$\min J(\tilde{b}) := \sum_{i=1}^{n-1} \tfrac{1}{2}(\tilde{y}_i - \tilde{b}_i)^2 + \sum_{i=1}^{n-1} \tilde{\lambda}_i |\tilde{b}|_{(i)} \tag{A.5}$$

in which $\tilde{\lambda}_i = \lambda_{i+1}$ has exactly $r - 1$ nonzero entries. We prove this in two steps:

(i) The optimal solution $\hat{b}$ to (A.5) has at least $r - 1$ nonzero entries.

(ii) The optimal solution $\hat{b}$ to (A.5) has at most $r - 1$ nonzero entries.

A.2.1 Proof of (i)

Suppose by contradiction that $\hat{b}$ has fewer than $r - 1$ entries; i.e., $\hat{b}$ has $j - 1$ nonzero entries with $j < r$. Letting I be those indices for which the rank of $\tilde{y}_i$ is between j and $r - 1$, consider a feasible point b as in the proof of Lemma A.3 defined as

$$b_i = \begin{cases} h & i \in I, \\ \hat{b}_i & \text{otherwise}; \end{cases}$$

here, the positive scalar h obeys $0 < h < b_{(j-1)}$. By definition,

$$J(b) - J(\hat{b}) = -h \sum_{i=j}^{r-1} (\tilde{y}_{(i)} - \tilde{\lambda}_i) + h^2 \sum_{i=j}^{r-1} \tfrac{1}{2}.$$

Now

$$\sum_{j \leq i \leq r-1} \tilde{y}_{(i)} - \tilde{\lambda}_i = \sum_{j+1 \leq i \leq r} \tilde{y}_{(i-1)} - \lambda_i \geq \sum_{j+1 \leq i \leq r} y_{(i)} - \lambda_i > 0.$$

The first equality follows from $\tilde{\lambda}_i = \lambda_{i+1}$, the first inequality from $y_{(i)} \leq \tilde{y}_{(i-1)}$ and the last from (A.3). By selecting h small enough, this gives $J(b) < J(\hat{b})$, which contradicts the optimality of $\hat{b}$.

A.2.2 Proof of (ii)

The proof is similar to that of (i). Suppose by contradiction that $\hat{b}$ has more than $r - 1$ entries; i.e. $\hat{b}$ has j nonzero entries with $j \geq r$. Letting I be those indices for which the rank of $\tilde{y}_i$ is between r and j, consider a feasible point b as in the proof of Lemma A.3 defined as

$$b_i = \begin{cases} b_i - h & i \in I \\ \hat{b}_i & \text{otherwise}; \end{cases}$$

here, the positive scalar h obeys $0 < h < b_{(j)}$. By definition,

$$J(b) - J(\hat{b}) = h \sum_{i=r}^{j} (\tilde{y}_{(i)} - \tilde{\lambda}_i - \hat{b}_{(i)}) + h^2 \sum_{i=r}^{j} \tfrac{1}{2}.$$

Now

$$\sum_{r\leq i\leq j}(\tilde{y}_{(i)}-\tilde{\lambda}_i)=\sum_{r+1\leq i\leq j+1}(y_{(i)}-\lambda_i)\leq 0.$$

The equality follows from the definition and the inequality from (A.4). By selecting h small enough, this gives $J(b)<J(\hat{b})$, which contradicts the optimality of $\hat{b}$.

B Algorithmic Issues

B.1 Duality-based stopping criteria

To derive the dual of (1.10) we first rewrite it as

$$\underset{b,r}{\text{minimize}}\quad \tfrac{1}{2}r'r+J_\lambda(b)\quad \text{subject to}\quad Xb+r=y.$$

The dual is then given by

$$\underset{w}{\text{maximize}}\quad \mathcal{L}(b,r,w),$$

where

$$\begin{aligned}\mathcal{L}(b,r,w)\ &:=\ \inf_{b,r}\ \{\tfrac{1}{2}r'r+J_\lambda(b)-w'(Xb+r-y)\}\\ &=\ w'y-\sup_r\ \{w'r-\tfrac{1}{2}r'r\}-\sup_b\ \{(X'w)'b-J_\lambda(b)\}.\end{aligned}$$

The first supremum term evaluates to $\frac{1}{2}w'w$ by choosing $r=w$. The second term is the conjugate function J^* of J evaluated at $v=X'w$, which can be shown to reduce to

$$J_\lambda^*(v):=\sup_b\ \{v'b-J_\lambda(b)\}=\begin{cases}0 & v\in C_\lambda,\\ +\infty & \text{otherwise,}\end{cases}$$

where the set C_λ is the unit ball of the dual norm to $J_\lambda(\cdot)$. In details,

$$w\in C_\lambda\quad\Longleftrightarrow\quad \sum_{j\leq i}|w|_{(j)}\leq\sum_{j\leq i}\lambda_j\ \text{for all } i=1,\ldots,p.$$

The dual problem is thus given by

$$\underset{w}{\text{maximize}}\quad w'y-\tfrac{1}{2}w'w\quad\text{subject to}\quad w\in C_\lambda.$$

The dual formulation can be used to derive appropriate stopping criteria. At the solution we have $w=r$, which motivates estimating a dual point by setting $\hat{w}=r=:y-Xb$. At this point the primal-dual gap at b is the difference between the primal and dual objective:

$$\delta(b)=(Xb)'(Xb-y)+J_\lambda(b).$$

However, $\hat{w}$ is not guaranteed to be feasible, i.e., we may not have $\hat{w}\in C_\lambda$. Therefore we also need to compute a level of infeasibility of $\hat{w}$, for example

$$\text{infeasi}(\hat{w})=\max\Big\{0,\ \max_i\sum_{j\leq i}(|\hat{w}|_{(j)}-\lambda_j)\Big\}.$$

The algorithm used in the numerical experiments terminates whenever both the infeasibility and primal-dual gap are sufficiently small. In addition, it imposes a limit on the total number of iterations to ensure termination.

B.2 Proof of Lemma 2.3

It is useful to think of the prox as the solution to the quadratic program (2.3) and we begin by recording the Karush-Kuhn-Tucker (KKT) optimality conditions for this QP.

Primal feasibility: $x_1 \geq x_2 \geq \cdots \geq x_n \geq 0$.

Dual feasibility: $\mu = (\mu_1, \ldots, \mu_n)$ obeys $\mu \geq 0$.

Complementary slackness: $\mu_i(x_i - x_{i+1}) = 0$ for all $i = 1, \ldots, n$ (with the convention $x_{n+1} = 0$).

Stationarity of the Lagrangian: with the convention that $\mu_0 = 0$,

$$x_i - y_i + \lambda_i - (\mu_i - \mu_{i-1}) = 0.$$

We now turn to the proof of the second claim of the lemma. Set $x = (y - \lambda)_+$, which by assumption is primal feasible, and let i_0 be the last index such that $y_i - \lambda_i > 0$. Set $\mu_1 = \mu_2 = \ldots = \mu_{i_0} = 0$ and for $j > i_0$, recursively define

$$\mu_j = \mu_{j-1} - (y_j - \lambda_j) \geq 0.$$

Then it is straightforward to check that the pair (x, μ) obeys the KKT optimality conditions. Hence x is solution.

Consider now the first claim. We first argue that the prox has to be constant over any monotone segment of the form

$$y_i - \lambda_i \leq y_{i+1} - \lambda_{i+1} \leq \ldots \leq y_j - \lambda_j.$$

To see why this is true, set $x = \text{prox}(y; \lambda)$ and suppose the contrary: then over a segment as above, there is $k \in \{i, i+1, \ldots, j-1\}$ such that $x_k > x_{k+1}$ (we cannot have a strict inequality in the other direction since x has to be primal feasible). By complementary slackness, $\mu_k = 0$. This gives

$$\begin{aligned} x_k &= y_k - \lambda_k - \mu_{k-1} \\ x_{k+1} &= y_{k+1} - \lambda_{k+1} + \mu_{k+1}. \end{aligned}$$

Since $y_{k+1} - \lambda_{k+1} \geq y_k - \lambda_k$ and $\mu \geq 0$, we have $x_k \leq x_{k+1}$, which is a contradiction.

Now an update replaces an increasing segment as in (2.4) with a constant segment and we have just seen that both proxes must be constant over such segments. Now consider the cost function associated with the prox with parameter λ and input y over an increasing segment as in (2.4),

$$\sum_{i \leq k \leq j} \Big\{ \tfrac{1}{2}(y_k - x_k)^2 + \lambda_k x_k \Big\}. \tag{B.1}$$

Since all the variables x_k must be equal to some value z over this block, this cost is equal to

$$\begin{aligned} \sum_{i \leq k \leq j} \Big\{ \tfrac{1}{2}(y_k - z)^2 + \lambda_k z \Big\} &= \sum_k \tfrac{1}{2}(y_k - \bar{y})^2 + \sum_{i \leq k \leq j} \Big\{ \tfrac{1}{2}(\bar{y} - z)^2 + \bar{\lambda} z \Big\} \\ &= \sum_k \tfrac{1}{2}(y_k - \bar{y})^2 + \sum_{i \leq k \leq j} \Big\{ \tfrac{1}{2}(y_k^+ - z)^2 + \bar{\lambda}_k^+ z \Big\}, \end{aligned}$$

where $\bar{y}$ and $\bar{\lambda}$ are block averages. The second term in the right-hand side is the cost function associated with the prox with parameter λ^+ and input y^+ over the same segment since all the variables over this segment must also take on the same value. Therefore, it follows that replacing each appearance of block sums as in (B.1) in the cost function yields the same minimizer. This proves the claim.