bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

A fast and efficient smoothing approach to Lasso regression and an application in statistical genetics: polygenic risk scores for chronic obstructive pulmonary disease (COPD)

Georg Hahn, Sharon M. Lutz, Nilanjana Laha, Michael H. Cho,
Edwin K. Silverman, and Christoph Lange

Abstract

High dimensional linear regression problems are often fitted using Lasso approaches. Although the Lasso objective function is convex, it is not differentiable everywhere, making the use of gradient descent methods for minimization not straightforward. To avoid this technical issue, we apply Nesterov smoothing to the original (unsmoothed) Lasso objective function. We introduce a closed-form smoothed Lasso which preserves the convexity of the Lasso function, is uniformly close to the unsmoothed Lasso, and allows us to obtain closed-form derivatives everywhere for efficient and fast minimization via gradient descent. Our simulation studies are focused on polygenic risk scores using genetic data from a genome-wide association study (GWAS) for chronic obstructive pulmonary disease (COPD). We compare accuracy and runtime of our approach to the current gold standard in the literature, the FISTA algorithm. Our results suggest that the proposed methodology provides estimates with equal or higher accuracy than the FISTA algorithm while having the same asymptotic runtime scaling. The proposed methodology is implemented in the R-package *smoothedLasso*, available on the Comprehensive R Archive Network (CRAN).

Keywords: COPD; FISTA; LASSO; Nesterov; Penalized linear regression; Polygenic risk scores; Smoothing.

1 Introduction

Many substantive research questions in health, economic and social science require solving a classical linear regression problem $y = X\beta + \epsilon$. Here, the data matrix $X \in \mathbb{R}^{n\times p}$, the parameter vector $\beta \in \mathbb{R}^p$, and the response vector $y \in \mathbb{R}^n$ encode $n \in \mathbb{N}$ linear equations in $p \in \mathbb{N}$ variables, and

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

$\epsilon \sim N_n(0, \sigma^2 I_n)$ is an n-dimensional, independently and normally distributed error term with mean zero and some variance $\sigma^2 > 0$ (where I_n denotes the n-dimensional identity matrix). The approach remains one of the most widely used statistical analysis tools.

Traditionally, linear regression problems have been solved by finding parameter estimates β that minimize the squared error, leading to the least squares estimate $\arg\min_\beta \|X\beta - y\|_2^2$. However, their lack of robustness, as well as sparsity requirements in high dimensional settings with $p \gg n$, are two problems that have led to the development of alternative estimation approaches, e.g. the *least absolute shrinkage and selection operator* (Lasso) of Tibshirani (1996) or *least-angle regression* (LARS) of Efron et al. (2004).

This article focuses on Lasso regression, which obtains regression estimates $\hat{\beta}$ by solving

$$\hat{\beta} = \arg\min_\beta \frac{1}{n} \|X\beta - y\|_2^2 + \lambda \|\beta\|_1 , \tag{1}$$

where $\|\cdot\|_2$ is the Euclidean norm, $\|\cdot\|_1$ is the L_1 norm, and $\lambda > 0$ is a tuning parameter (called the Lasso regularization parameter) controlling the sparseness of the solution $\hat{\beta}$.

As the objective function in eq. (1) is convex, minimization via steepest descent (quasi-Newton) methods is sensible. However, many applications in biostatistics, especially those that are focused on *big data*, such as the simultaneous analysis of genome-wide association studies (Wu et al., 2009) or the calculation of polygenic risk scores (Mak et al., 2016), involve data sets with several thousand parameters and are often sparse. In such applications, conventional gradient-free solvers can lose accuracy. This is due to the non-differentiability of the L_1 penalty term in eq. (1), i.e. the term $\|\beta\|_1 = \sum_{i=1}^p |\beta_i|$.

We address this issue by smoothing the Lasso objective function. We apply Nesterov smoothing (Nesterov, 2005) to the non-differentiable $\|\beta\|_1$ penalty term in eq. (1). This will result in an approximation of the Lasso objective function that is differentiable everywhere. The Nesterov formalism depends on a smoothing parameter μ that controls the smoothness level.

Our approach has three major advantages: (1) The smoothing preserves the convexity of the Lasso objective function; (2) it allows us to obtain closed-form derivatives of the smoothed function everywhere which we use in a gradient descent algorithm; and (3) it provides uniform error bounds on the difference between the smoothed and the original objective functions. The error bounds

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

depend only on the smoothing parameter μ, and by letting $\mu \to 0$, the smoothed Lasso objective function can be made arbitrarily close to the original Lasso objective function, thus justifying its use as a surrogate of the original Lasso objective. However, when smoothing the L_1 penalty of the Lasso, the resulting regression estimates are typically dense. That is, the lack of the L_1 penalty somewhat causes a loss of the variable selection property, though regression estimates for those variables which are non-selected by the classical Lasso will typically be very small. Therefore, an easy fix to obtain sparse estimates again is to apply some form of thresholding.

We benchmark our proposed smoothing approach against the current gold standard for minimizing the Lasso objective function, the FISTA algorithm of Beck and Teboulle (2009). FISTA is a proximal gradient version of the algorithm of Nesterov (1983) which combines the basic iterations of the Iterative Shrinkage-Thresholding Algorithm (Daubechies et al., 2004) with a Nesterov acceleration step. Among others, the algorithm is implemented in the R-package *fasta* on CRAN (Chi et al., 2018). We use this package as a benchmark. The FISTA algorithm requires the separate specification of the smooth and non-smooth parts of the objective function including their explicit gradients. In contrast to our approach, a variety of tuning parameters need to be selected by the user, e.g. an initial starting value, an initial stepsize, parameters determining the *lookback* window for non-monotone line search, and a shrinkage parameter for the stepsize.

The benchmark of our approach against FISTA is carried out on simulated data, with the aim to assess accuracy and runtime scaling of both algorithms in n and p, and on a real-world dataset of polygenic risk scores of COPDGene (genetic epidemiology of COPD). Polygenic risk scores are a statistical aggregate of risks typically associated with a set of established DNA variants. The goal of polygenic risk scores is to predict the disease risk of an individual, that is the susceptibility to a certain disease given the individual's genotypes at the established risk loci and other important epidemiological covariates such as age, sex etc. Such scores are usually calibrated on large genome-wide association studies (GWAS) via high-dimensional regression of the genetic variants to the outcome. Since the potential for broad-scale clinical use to identify people at high risk for certain diseases has been demonstrated (Khera et al., 2018), polygenic risk scores have become a widespread tool for the early identification of patients who are at high risk for a certain disease and who could benefit from intervention measures. For this reason, increasing the accuracy of scores is desirable, which is what we address with the proposed smoothing approach.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

This article is structured as follows. Section 2 applies Nesterov smoothing to the Lasso objective function. Using polygenic risk scores as an example, we evaluate the proposed methodology in a simulation study in Section 3. The article concludes with a discussion in Section 4. The proof of our theoretical result is given in Appendix C. The methodology of this article is implemented in the R-package *smoothedLasso*, available on CRAN (Hahn et al., 2020).

Throughout the article, $X_{\cdot,i}$ denotes the ith column of a matrix X. Similarly, $X_{I,\cdot}$ (and y_I) denote the submatrix (subvector) consisting of all rows of X (entries of y) indexed in the set I. Moreover, $X_{-I,\cdot}$ (and y_{-I}) denote the submatrix (subvector) consisting of all rows of X (entries of y) not indexed in the set I. Finally, $|\cdot|$ denotes the absolute value, and $\|\cdot\|_\infty$ denotes the supremum norm.

2 Smoothing the Lasso objective function

This section lays the theoretical foundation for the modified Lasso approach we propose to address the non-differentiability of the L_1 penality term in eq. (1), while the smooth L_2 term remains unchanged. The substitute of the term $\|\beta\|_1 = \sum_{i=1}^p |\beta_i|$ in eq. (1) will be computed with the help of a technique called Nesterov smoothing, whose details are given in Section 2.1. In Section 2.2, we simplify the general Nesterov smoothing approach for the particular case of the Lasso penalty and show how our approach results in explicit closed-form expressions for both the smoothed Lasso and its gradient. The uniform closeness of the smoothed Lasso we propose and the original Lasso objective function is proven in Section 2.3.

2.1 Summary of Nesterov smoothing

We briefly summarize the idea of Nesterov smoothing. We are given a piecewise affine and convex function $f : \mathbb{R}^q \to \mathbb{R}$ which we aim to smooth, where $q \in \mathbb{N}$. We assume that f is composed of $k \in \mathbb{N}$ linear pieces (components) and can thus be expressed as

$$f(z) = \max_{i=1,\dots,k} \left(A[z,1]^\top \right)_i ,$$

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

where $A \in \mathbb{R}^{k\times(q+1)}$ is a matrix whose rows contain the linear coefficients for each of the k linear pieces (with the constant coefficients being in column $q+1$), $z \in \mathbb{R}^q$, and $[z,1] \in \mathbb{R}^{q+1}$ denotes the vector obtained by concatenating z and the scalar 1.

Let $\mu \geq 0$ be the Nesterov smoothing parameter. Using a so-called proximity (or prox) function ρ, which is assumed to be nonnegative, continuously differentiable, and strongly convex, f is replaced by the approximation

$$f^{\mu}(z) = \max_{w\in Q_k} \left\{ \langle A[z,1]^{\top}, w\rangle - \mu\rho(w) \right\},$$

where $Q_k \subseteq \mathbb{R}^k$ is the unit simplex in k dimensions defined as

$$Q_k = \left\{ w \in \mathbb{R}^k : \sum_{i=1}^{k} w_i = 1, w_i \geq 0 \ \forall i = 1,\ldots,k \right\}.$$

As shown in (Nesterov, 2005, Theorem 1), f^{μ} is both smooth and uniformly close to f, meaning that for all $z \in \mathbb{R}^q$, the approximation error is uniformly upper bounded by

$$\sup_{z\in\mathbb{R}^q} |f(z) - f^{\mu}(z)| \leq \mu \sup_{w\in Q_k} \rho(w) = O(\mu). \tag{2}$$

The larger the value of μ the more f is smoothed, while $\mu = 0$ recovers the original function $f = f^0$. Two choices of the prox function are given a closer look in Section 2.2. Smoothing with the so-called entropy prox function results in the smooth approximation f_e^{μ} having a closed-form expression given by

$$f_e^{\mu}(z) = \mu \log\left(\frac{1}{k} \sum_{i=1}^{k} e^{\frac{\left(A[z,1]^{\top}\right)_i}{\mu}} \right), \tag{3}$$

which satisfies the uniform bound

$$\sup_{z\in\mathbb{R}^q} |f(z) - f_e^{\mu}(z)| \leq \mu \log(k). \tag{4}$$

Similarly, the smooth approximation of f with the help of the so-called squared error prox function

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

given by $\rho_s(w) = \frac{1}{2}\sum_{i=1}^{k}\left(w_i - \frac{1}{k}\right)^2$ for $w \in \mathbb{R}^k$ can be written as

$$f_s^\mu(z) = \langle \hat{c}(z), A[z,1]^\top \rangle - \mu\rho_s(\hat{c}(z)). \tag{5}$$

In eq. (5), $\hat{c}(z) \in \mathbb{R}^k$ is the Michelot projection (Michelot, 1986) of the vector $c(z) = (c_1(z), \ldots, c_k(z))$, given componentwise by $c_i(z) = 1/\mu \cdot \left(A[z,1]^\top\right)_i - 1/k$ for $i \in \{1, \ldots, k\}$, onto the k-dimensional unit simplex Q_k. The approximation f_s^μ via squared error prox function satisfies the uniform bound

$$\sup_{z\in\mathbb{R}^q} |f(z) - f_s^\mu(z)| \leq \mu\left(1 - \frac{1}{k}\right). \tag{6}$$

2.2 Application to the Lasso objective function

For given $X \in \mathbb{R}^{n\times p}$, $y \in \mathbb{R}^n$, and $\lambda > 0$, according to eq. (1), the Lasso objective function $L : \mathbb{R}^p \to \mathbb{R}$ given by

$$L(\beta) = \frac{1}{n}\|X\beta - y\|_2^2 + \lambda\|\beta\|_1 \tag{7}$$

is smooth in its first term but non-differentiable in the penalty term $\|\beta\|_1 = \sum_{i=1}^{p}|\beta_i|$. We thus smooth the latter, where it suffices to apply Nesterov smoothing to each absolute value independently.

Let $k = 2$. Using one specific choice of the matrix $A \in \mathbb{R}^{2\times 2}$ given by

$$A = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix},$$

we rewrite the (one dimensional) absolute value as $f(z) = \max\{-z, z\} = \max_{i=1,2}\left(A[z,1]^\top\right)_i$, where here and in the following subsections $z \in \mathbb{R}$ is a scalar.

2.2.1 Entropy prox function

For the entropy prox function, eq. (3) with A as in Section 2.2 simplifies to

$$f_e^\mu(z) = \mu\log\left(\frac{1}{2}e^{-z/\mu} + \frac{1}{2}e^{z/\mu}\right),$$

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

which according to eq. (4) satisfies the approximation bound

$$\sup_{z\in\mathbb{R}} |f(z) - f_e^\mu(z)| \leq \mu \log(2). \tag{8}$$

The first and second derivatives of f_e^μ are given by

$$\begin{aligned}
\frac{\partial}{\partial z} f_e^\mu(z) &= \frac{-e^{-z/\mu} + e^{z/\mu}}{e^{-z/\mu} + e^{z/\mu}} =: g_e^\mu(z), \\
\frac{\partial^2}{\partial z^2} f_e^\mu(z) &= \frac{4}{\mu(e^{-z/\mu} + e^{z/\mu})^2} =: h_e^\mu(z).
\end{aligned}$$

Together, smoothing eq. (7) with the entropy prox function results in

$$\begin{aligned}
L_e^\mu(\beta) &= \frac{1}{n}\|X\beta - y\|_2^2 + \lambda \sum_{i=1}^p f_e^\mu(\beta_i), \\
\frac{\partial L_e^\mu}{\partial \beta_i}(\beta) &= -\frac{2}{n}\langle y - X\beta, X_{\cdot,i}\rangle + \lambda g_e^\mu(\beta_i), \\
\frac{\partial^2 L_e^\mu}{\partial \beta_i \partial \beta_j}(\beta) &= \frac{2}{n}(X^\top X)_{ij} + \mathbb{I}(i=j)\cdot \lambda h_e^\mu(\beta_i),
\end{aligned} \tag{9}$$

where the entropy prox smoothed Lasso is L_e^μ, its explicit gradient is $\partial L_e^\mu / \partial \beta_i$, and its Hessian matrix is given by $\partial^2 L_e^\mu / \partial \beta_i \partial \beta_j$. The function $\mathbb{I}(\cdot)$ denotes the indicator function.

In principle, the Lasso objective can be minimized using a (second order) Newton-Raphson or a (first order) quasi-Newton approach. However, since $X \in \mathbb{R}^{n\times p}$ with $n < p$, the matrix $X^\top X$ is singular, meaning that for the Hessian to be invertible one needs the added diagonal elements $\lambda h_e^\mu(\beta_i)$ to be large. However, this is usually not true, since if β_i is nonzero, then in a neighborhood of the true Lasso estimate the term $(e^{-\beta_i/\mu} + e^{\beta_i/\mu})^{-2}$ will be exponentially small. Thus to make $\lambda h_e^\mu(\beta_i)$ large for a fixed μ, we need λ to be exponentially large. Likewise, given λ and β_i, too small or too large values of μ will make h_e^μ vanish. However, since typically λ and μ are fixed, the Hessian in eq. (9) will be singular except for a few artificial cases and thus the second order Newton-Raphson method will not be applicable. In the simulations we therefore focus on quasi-Newton methods which require only L_e^μ and its gradient $\partial L_e^\mu / \partial \beta_i$.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

2.2.2 Squared error prox function

Similarly, eq. (5) with A as in Section 2.2 simplifies to

$$f_s^\mu(z) = \langle \hat{c}(z), [-z, z] \rangle - \mu \rho_s(\hat{c}(z)).$$

where $\hat{c}(z) \in \mathbb{R}^2$ is the Michelot projection of the vector $c(z) = 1/\mu \cdot [-z, z] - 1/k$ onto the two-dimensional unit simplex Q_2. According to eq. (6), we obtain the approximation bound

$$\sup_{z \in \mathbb{R}} |f(z) - f_s^\mu(z)| \leq \frac{1}{2}\mu. \tag{10}$$

The derivative of f_s^μ is given by

$$\frac{\partial}{\partial z} f_s^\mu(z) = \langle \hat{c}(z), [-1, 1] \rangle =: g_s^\mu(z),$$

see (Hahn et al., 2017, Lemma 4) for a proof of this result. The second derivative $\partial^2 f_s^\mu / \partial z^2$ does not have a closed form expression, though it can be approximated numerically. Analogously to the results for the entropy prox function, smoothing eq. (7) with the squared error prox function results in

$$\begin{aligned} L_s^\mu(\beta) &= \frac{1}{n}\|X\beta - y\|_2^2 + \lambda \sum_{i=1}^p f_s^\mu(\beta_i), \\ \frac{\partial L_s^\mu}{\partial \beta_i}(\beta) &= -\frac{2}{n}\langle y - X\beta, X_{\cdot,i} \rangle + \lambda g_s^\mu(\beta_i), \\ \frac{\partial^2 L_s^\mu}{\partial \beta_i \partial \beta_j}(\beta) &= \frac{2}{n}(X^\top X)_{ij} + \mathbb{I}(i = j) \cdot \lambda \left. \frac{\partial^2 f_s^\mu}{\partial z^2} \right|_{z = \beta_i} \end{aligned} \tag{11}$$

where as before, the smoothed Lasso via squared error prox function is L_s^μ, its explicit gradient is $\partial L_s^\mu / \partial \beta_i$, and its Hessian matrix is $\partial^2 L_s^\mu / \partial \beta_i \partial \beta_j$.

As in Section 2.2.1 we observe that the Hessian matrix is singular since $X^\top X$ is singular, and since the additional diagonal entries stemming from $\lambda \partial^2 f_s^\mu / \partial z^2$ are usually too small to make the Hessian invertible. We therefore again resort to a quasi-Newton method to minimize L_s^μ with the help of its gradient $\partial L_s^\mu / \partial \beta_i$ only.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

2.3 Theoretical error bounds

The approximation bounds on the absolute value, given in eqs. (8) and (10), carry over to bounds on the distance between the unsmoothed and smoothed Lasso objective functions, as shown in the following proposition.

Proposition 1. *The smoothed Lasso objective functions with entropy and squared error prox, given in eqs.* (9) *and* (11)*, satisfy the following uniform bounds:*

$$\sup_{\beta \in \mathbb{R}^p} |L_e^\mu(\beta) - L(\beta)| \leq \lambda p \mu \log(2),$$
$$\sup_{\beta \in \mathbb{R}^p} |L_s^\mu(\beta) - L(\beta)| \leq \frac{\lambda p \mu}{2}.$$

Moreover, both L_e^μ and L_s^μ are strictly convex.

In Proposition 1 the Lasso parameter $\lambda > 0$ and the dimension p are fixed for a particular estimation problem, thus allowing to make the approximation error arbitrarily small as the smoothing parameter $\mu \to 0$. This justifies the use of our proposed smoothed Lasso as a surrogate of the original (non smooth) Lasso objective function. Moreover, the convexity of the Lasso is preserved.

3 Simulation studies for polygenic risk scores for COPD

In this section, we evaluate three approaches to compute Lasso estimates on simulated data (Section 3.1) as well as real data coming from a genome-wide association study for COPDGENE (Regan et al., 2010), in which polygenic risk scores are computed and evaluated (Section 3.2). Of the three approaches we consider, the first two utilize existing methodology, while the last approach implements the methodology we developed in this paper:

1. We carry out the minimization of eq. (1) using R's function *optim*. The *optim* function implements the quasi-Newton method *BFGS* for which we supply the explicit (though non-smooth) gradient $\partial L / \partial \beta_i = -2/n \cdot \langle y - X\beta, X_{\cdot,i} \rangle + \lambda \text{sign}(\beta_i)$. This approach will be referred to as the *unsmoothed Lasso*;

2. We use the FISTA algorithm as implemented in the *fasta* R-package (Chi et al., 2018), available on *The Comprehensive R Archive Network* (R Core Team, 2014);

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

3. We minimize the smoothed Lasso objective function of eq. (9) using its explicit gradient. Note that due to the smoothed L_1 penalty, the resulting estimates returned by the smoothed Lasso are typically dense, though sparse estimates are easily obtained by applying some form of thresholding.

The main function of the *fasta* package which implements the FISTA algorithm, also called *fasta*, requires the separate specification of the smooth and non-smooth parts of the objective function including their explicit gradients. We follow Example 2 in the *vignette* of the *fasta* R-package in Chi et al. (2018) and supply both as specified in eq. (7). Additionally, we employ a uniform random starting value as done for the unsmoothed and smoothed Lasso. The initial stepsize is set to $\tau = 10$ as in Example 2 of Chi et al. (2018). The lookback window for non-monotone line search and the shrinkage parameter for the stepsize are left at their default values.

Three potential implementations are unconsidered in this simulation section for the following reasons. The *glmnet* algorithm of Friedman et al. (2010), available in the R-package *glmnet*, is a variant of FISTA which performs a cyclic update of all coordinates, whereas FISTA updates all coordinates per iteration. We thus focus on FISTA. Since the R-package *SIS* accompanying Fan and Li (2001) does itself rely on *glmnet* for computing regression estimates, we omit it in this section. The LARS algorithm of Efron et al. (2004) is implemented in the R-package *lars* on CRAN (Hastie and Efron, 2013). As remarked in Friedman et al. (2010), LARS is slower than glmnet/ FISTA. Additionally, since the implementation of Hastie and Efron (2013) always computes a full Lasso path, it is considerably slower than the other methods.

All results are averages over 100 repetitions. The Lasso regularization parameter was fixed at $\lambda = 0.3$ after performing 10-fold cross validation. Details of the cross validation computation are given in Appendix A. The smoothing parameter of the approach of Section 2.2 was fixed at $\mu = 0.1$ in all simulations. As shown in Proposition 1, the smaller μ the better the proposed smoothed Lasso will approximate the original Lasso, though numerical instabilities may occur if μ is chosen too close to the machine precision. All computations were carried out on 100 cluster nodes, each one having a lx24-amd64 architecture with 4 cores per node, 94 GB of memory, and running 64-bit CentOS 6 Linux.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

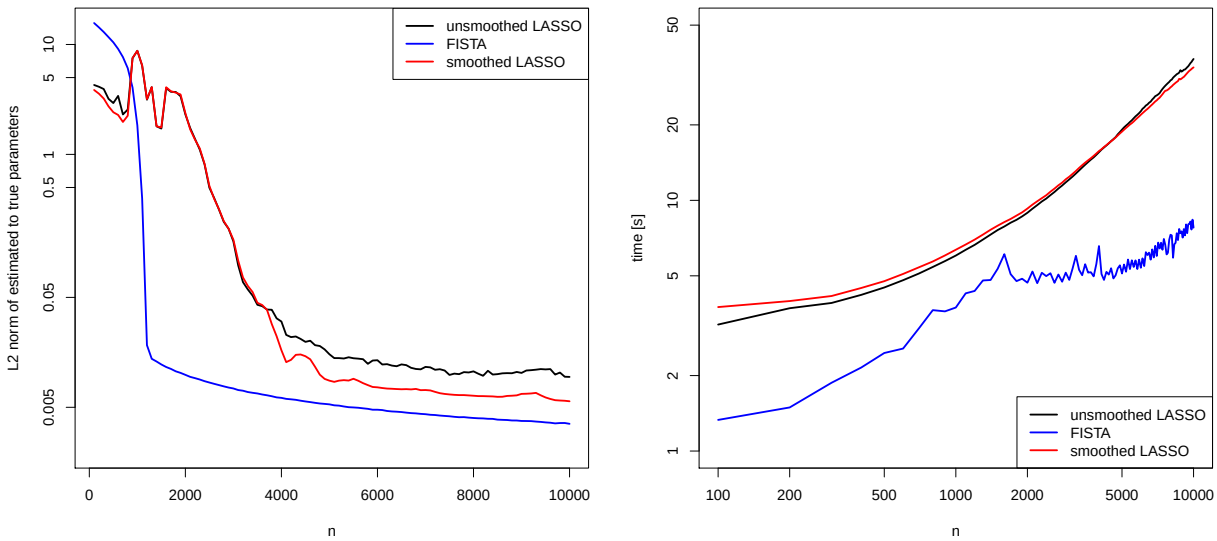

Figure 1: Simulation study: L_2 norm of parameter estimate to truth (left) and runtime in seconds (right) as a function of $n \in [1, 10000]$ while $p = 1000$. Logarithmic scale on the y-axes.

3.1 Application to simulated data

We designed our simulation study so that it mimics the application to a genome-wide association study, i.e. we simulate $X \in \mathbb{R}^{n \times p}$ from a multidimensional normal distribution, where the entries of the mean vector of the multidimensional normal distribution are sampled independently from a uniform distribution in $[0, 0.5]$. To ensure positive definiteness of the covariance matrix Σ of the multidimensional normal distribution, we set $\Sigma = \frac{1}{2}(A + A^\top) + nD_n$, where D_n is a $n \times n$ matrix with ones on its diagonal, and zeros otherwise. The added term nD_n ensures positive definiteness.

After X is obtained, we generate the entries of the true β independently from a standard normal distribution, and set all but $nz \in \{0, \ldots, p\}$ out of the p entries to zero. The number of non-zero entries $nz \in \mathbb{N}$ is a parameter of the simulations. The response $y \in \mathbb{R}^n$ is then easily obtained as $y = X\beta + \epsilon$, where the entries of the noise vector $\epsilon \in \mathbb{R}^n$ are generated independently from a Normal distribution with mean zero and some standard deviation σ. The smaller the standard deviation, the easier the recovery of β will be (see Appendix B for a study on how the choice of σ influences the recovery of β). We will employ $\sigma = 0.5$ in all simulations. In this subsection, we fix the number of true non-zero parameters at 20% (that is, $nz = 0.2p$), resulting in sparse parameter vectors.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

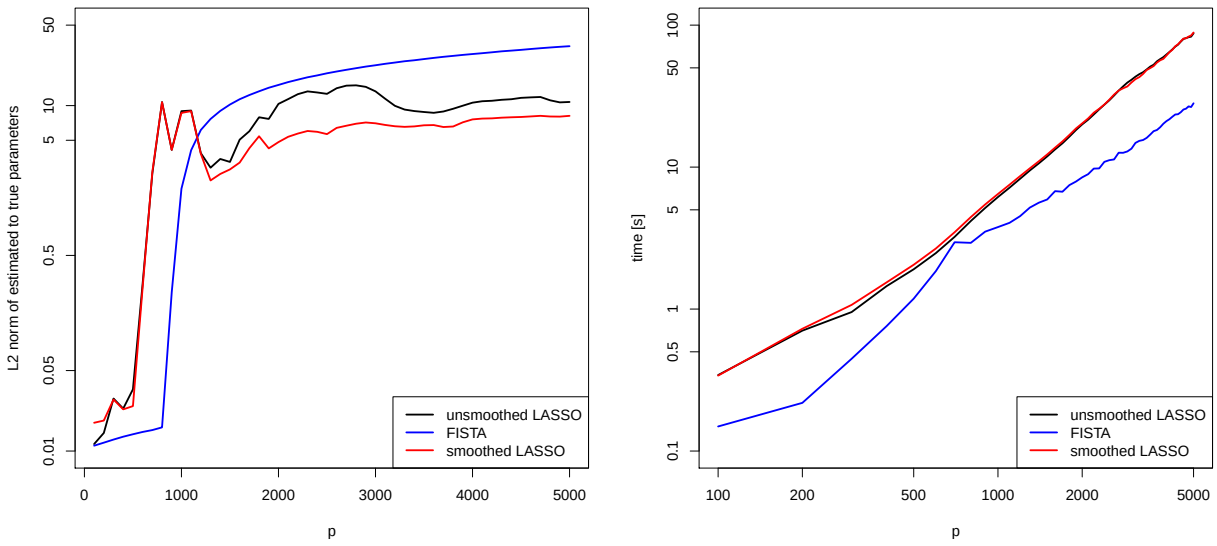

Figure 2: Simulation study: L_2 norm of parameter estimate to truth (left) and runtime in seconds (right) as a function of $p \in [1, 5000]$ while $n = 1000$. Logarithmic scale on the y-axes.

Figure 1 (left) shows results on simulated data of dimension $n \in [1, 10000]$ while keeping $p = 1000$ fixed. We measure the accuracy of the obtained Lasso estimates through their L_2 norm to the generated true parameters. We observe that the unsmoothed and smoothed Lasso approaches seem to suffer from numerical instabilities for small n. As n increases, both the unsmoothed and smoothed Lasso approaches stabilize. The FISTA algorithm yields stable estimates for all n. The accuracy of the smoothed Lasso approaches the one of FISTA for large n. As expected, all methods become more accurate as the number of data points n increases.

Figure 1 (right) shows that the unsmoothed and smoothed Lasso approaches have an almost identical runtime. We find that although highly optimized, FISTA is only a low multiple factor faster than our approaches and, more importantly, its asymptotic runtime scaling is comparable to the one of the smoothed Lasso.

Similarly to the previous experiment, in Figure 2 (left) we keep $n = 1000$ fixed and investigate the dependence of all four approaches on $p \in [1, 5000]$. Again, the unsmoothed and smoothed Lasso approaches seem to suffer from numerical instabilities for smaller p, while FISTA is very stable. Importantly, the smoothed Lasso seems to outperform FISTA for very large p. As expected, while keeping the data size n fixed, estimation becomes more challenging for all methods as p increases.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Figure 2 (right) confirms the timing results seen in the assessment of the dependence on n. The unsmoothed and smoothed approaches have virtually equal speeds, while FISTA is again slightly faster. Notably, the scaling of the runtimes seems to be roughly equal for all three methods.

Since the unsmoothed Lasso does not come with the guarantee we established for our smoothing approach (see Section 2.3) and is outperformed by the smoothed Lasso, we will focus in the remainder of the simulations on FISTA and smoothed Lasso only.

3.2 Application to polygenic risk scores

We evaluate the smoothed Lasso approach on polygenic risk scores of the COPDGene study (genetic epidemiology of COPD), a multi-center case-control study designed to identify genetic determinants of COPD and COPD-related phenotypes (Regan et al., 2010). The study has been sequenced as part of the TOPMed Project. The data is available through NHLBI TOPMed (2018).

For the study, non-Hispanic Whites and African Americans aged 45 to 80 were recruited as COPD cases and controls. The dataset contains $n = 4010$ individuals (rows), all of which had at least 10 pack-years of smoking history. For each individual, we observe $p = 9153$ datapoints, among them the covariates *age*, *sex*, *packyears*, *height* and five PCA vectors. The remaining entries are SNP data per individual. The input data are summarized in a matrix $X \in \mathbb{R}^{n \times p}$. The response $y \in \mathbb{R}^n$ is the *fev1* ratio, also called the Tiffeneau-Pinelli index, per individual. It describes the proportion of lung volume that a person can exhale within the first second of a forced expiration in a spirometry (pulmonary function) test. We aim to fit the four covariates, the five PCA vectors, and the SNP data to the response using our smoothing approach, with the aim to obtain a polygenic risk score of higher accuracy than obtained through conventional methods (see Section 1).

3.2.1 Results from a single run

We solve $y = X\beta$ for the given X and y using our smoothed Lasso approach of eq. (9), and compare it to the FISTA algorithm.

Table 1 shows results for a single application of both algorithms. After computing the estimate $\hat{\beta}$ with each method, we consider $\|y - X\hat{\beta}\|_2$, the L_2 norm between the fitted and generated (true) response. We observe that FISTA with standard choices of its tuning parameters seems to have trouble locating the minimum of the Lasso objective function, and is thus worse than smoothed

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

method	L_2 norm	runtime [s]
FISTA	284.6	66.6
smoothed Lasso	26.3	116.6

Table 1: COPDGene study: L_2 norm of fitted to true response and runtime in seconds for a single application of any method to the dataset of polygenic risk scores.

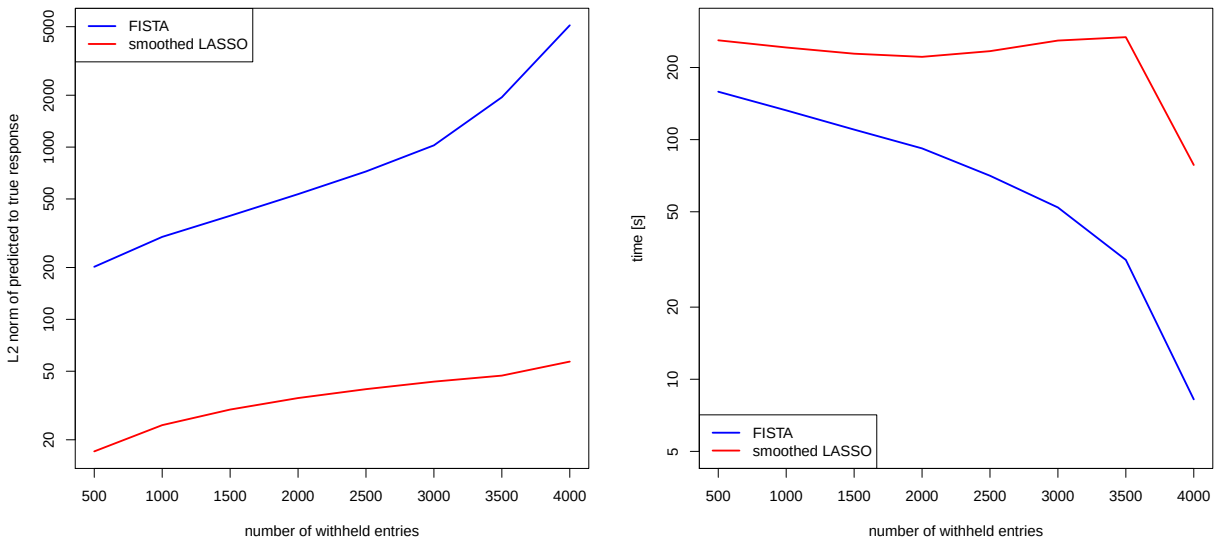

Figure 3: COPDGene study: L_2 norm of predicted to withheld data in cross-validation (left) and runtime in seconds (right) as a function of the number of withheld entries. Dataset of polygenic risk scores. Logarithmic scale on the y-axes.

Lasso. However, the FISTA method beats our approach in terms of runtime.

3.2.2 Cross validation

To quantify the accuracy of the methods further, we perform a simple cross validation experiment in which we withhold a random set of row indices I (of varying size) of the dataset X and the corresponding entries of the response y and fit a linear model to the rest of the data, that is we fit $y_{-I} = X_{-I,\cdot}\beta$. After obtaining an estimate $\hat{\beta}$, we use the withheld rows of X to predict the withheld entries of y, that is we compute $X_{I,\cdot}\hat{\beta}$. We evaluate the quality of the prediction by computing the L_2 norm $\|X_{I,\cdot}\hat{\beta} - y_I\|_2$ between predicted and withheld data.

Figure 3 (left) shows results of this cross-validation experiment. We observe that FISTA with default choices of its tuning parameters seems to have difficulties to converge to the minimum of the Lasso objective function, whereas smoothed Lasso performs better. Not surprisingly, the

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

quality of the prediction becomes worse in general as the number of withheld entries increases, since predictions are based on fewer and fewer datapoints. However, this decrease in accuracy seems to be more pronounced for FISTA than for our smoothed Lasso.

Finally, Figure 3 (right) displays runtime measurements for both approaches. We observe that smoothed Lasso seems to be rather insensitive to the number of withheld entries apart from a very large number of withheld entries. The FISTA algorithm is the faster method, however it exhibits a greater sensitivity to the number of withheld entries, that is the size of the estimation problem.

3.3 Application to synthetic polygenic risk scores

We aim to extend the simulations of Section 3.2 in order to vary the sparsity of the parameter estimate $\hat{\beta}$. To this end, we change the simulation setting as follows. Leaving $X \in \mathbb{R}^{n \times p}$ unchanged ($n = 4010$ individuals and $p = 9153$ datapoints), we simulate a parameter vector $\beta \in \mathbb{R}^p$ in which $nz \in \{0, \ldots, p\}$ entries are drawn from a Beta distribution with shape parameters 1.5 and 10. This will produce nonzero entries in the vector β of magnitude around 0.15, which is realistic in practice. The remaining $p - nz$ entries are set to zero. We then calculate the response as $y = X\beta + \epsilon$, where the entries of the noise vector $\epsilon \in \mathbb{R}^n$ are generated independently from a Normal distribution with mean 0 and standard deviation 0.1.

After generating X and y, we again use FISTA and smoothed Lasso to recover an estimate $\hat{\beta}$. We evaluate the quality of the estimate using $\|\beta - \hat{\beta}\|_2$, that is using the L_2 norm between truth and estimate.

Figure 4 (left) shows results as a function of the number nz of non-zero entries in the generated true β. We observe that in this experiment, and in contrast to Figures 1 and 2, the smoothed Lasso yields both more accurate and more stable estimates than FISTA (as expressed through a lower deviation from the truth in L_2 norm).

Figure 4 (right) shows runtime results for both approaches, showing that in this experiment the runtime scaling of FISTA seems to be rather insensitive to the simulation scenario. Our smoothed Lasso is slower than FISTA, and its runtime is more volatile. We note that in this experiment, the almost constant runtime of FISTA, irrespective of the simulation scenario, is likely a result of its inability to produce stable estimates in the first place.

Overall, we conclude from the simulations that in the scenarios we investigated, our smoothed

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

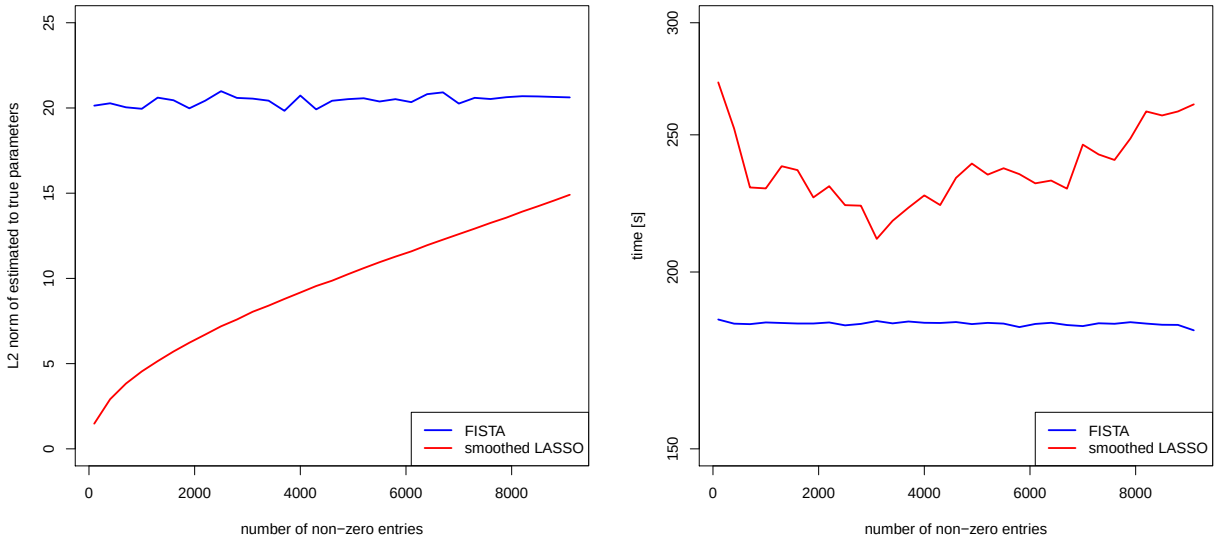

Figure 4: COPDGene study: L_2 norm of parameter estimate to truth (left) and runtime in seconds (right) as a function of the number of non-zero entries in the simulated true parameter vector. Dataset of polygenic risk scores. Logarithmic scale on the y-axis of the time plot.

Lasso approach yields estimates with equal or improved accuracy in comparison to the gold standard FISTA, exhibits roughly the same runtime scaling as FISTA, and comes with a guarantee on its accuracy.

4 Discussion

This article investigated a smoothing approach for penalized regression using the Lasso. The smoothing approach allows us to obtain smooth gradients everywhere, which facilitate minimization of the convex but non-smooth Lasso objective function.

Importantly, our approach comes with a guarantee of correctness. We prove a uniform bound on the distance between the unsmoothed and smoothed Lasso objective function, thus justifying the use of our smoothed Lasso as a surrogate of the original (non-smooth) Lasso objective function.

In the scenarios that we investigated, our simulation results suggest that the proposed smoothing approach yields Lasso estimates with equal or improved accuracy than the gold standard in the literature, the FISTA algorithm of Beck and Teboulle (2009). Our approach can directly be utilized for the computation of more accurate polygenic risk scores via high-dimensional regression of genetic

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

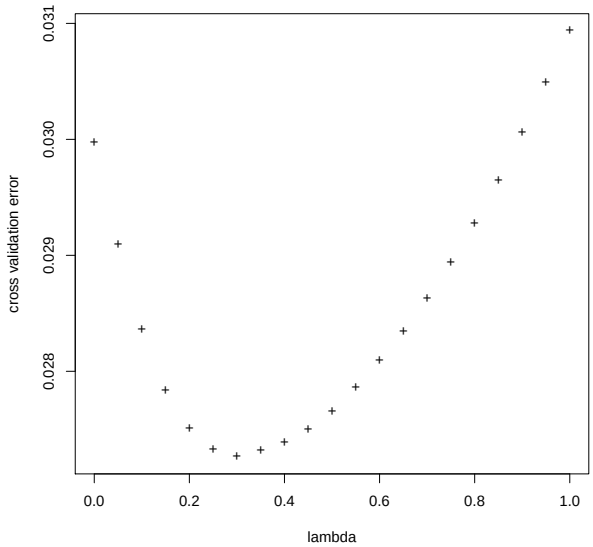

Figure 5: Cross validation error as a function of the Lasso regularization parameter λ for the FISTA algorithm. Simulated data with $n = 1000$ and $p = 100$.

variants from genome-wide association studies (and other covariates) to a certain outcome. This has the potential to increase the precision of risk prediction, thereby potentially identifying patients at high risk early on when they are still able to benefit from preventive steps targeted at reducing their chance of getting the disease.

Moreover, the idea of smoothing is not restricted to Lasso regression. Other important and widespread regression operators, such as the elastic net or graphical Lasso, might benefit from a similar smoothing approach to increase the accuracy of the obtained estimates. Extending the methodology to those operators remains for future work.

A Selection of the Lasso regularization parameter via cross validation

We aim to select the Lasso regularization parameter λ using cross validation. To this end, for the simulation scenario described in Section 3.1 (in particular, for the chosen noise level of $\sigma = 0.5$ and the sparsity level of 20%, as well as $n = 1000$ and $p = 100$), we perform 10-fold cross validation as described in Tibshirani (2013).

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

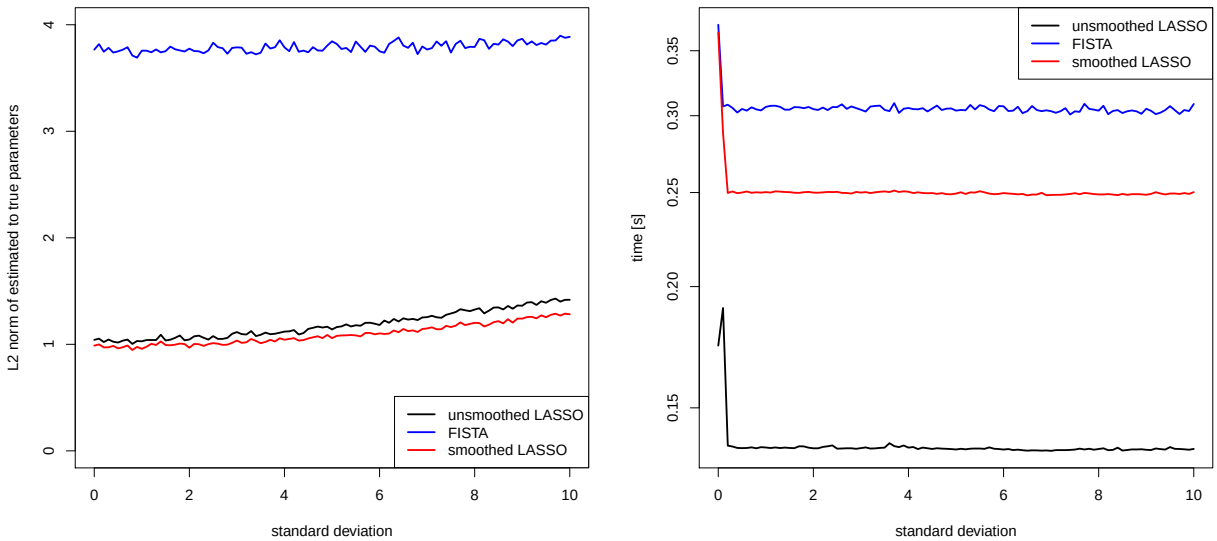

Figure 6: Sensitivity analysis on simulated data: L_2 norm of parameter estimate to truth (left) and runtime in seconds (right) as a function of the standard deviation σ while $n = 1000$ and $p = 2000$. Logarithmic scale on the y-axis of the right plot.

To be precise, we first fix a grid of admissible values of λ from which we would like to choose the regularization parameter (here, $\lambda \in \{0, 0.05, 0.1, 0.15, \ldots, , 1\}$). We then randomly divide the n data points into $K = 10$ disjoint sets (folds) $I_1, \ldots, I_K$ such that $\bigcup_{j=1}^{K} I_j = \{1, \ldots, n\}$. For each $j \in \{1, \ldots, K\}$, we withhold the indices in I_j and fit a linear model $y_{-I_j} = X_{-I_j,\cdot}\beta$ using the FISTA algorithm. After obtaining an estimate $\hat{\beta}$, we use the withheld rows of X indexed in I_j to predict the withheld entries of y, that is we compute $X_{I_j,\cdot}\hat{\beta}$. We evaluate the accuracy of the prediction with the L_2 norm, that is we compute $\|X_{I_j,\cdot}\hat{\beta} - y_{I_j}\|_2$. Repeating this computation for all $j \in \{1, \ldots, K\}$ allows us to compute an average L_2 error over the K folds (called the *cross validation error*). We plot this error as a function of λ.

The result is shown in Figure 5. We observe that for the simulation scenario we consider in Section 3.1, the choice $\lambda = 0.3$ is sensible.

B Sensitivity analysis

In the linear regression model $y = X\beta + \epsilon$ under consideration in this work (see Section 1), it is easy to see that the larger the noise/error ϵ, the harder it will be to obtain accurate estimates of β.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

To quantify this statement, Figure 6 presents a sensitivity analysis for the recovery accuracy of the parameter estimate β (measured as the L_2 norm between the fitted parameter estimate returned by the unsmoothed Lasso, FISTA algorithm, and smoothed Lasso, to the truth) as a function of the standard deviation σ. The setup of the simulation is identical to the one of Section 3.1, though now $n = 100$ and $p = 200$ are fixed. The entries of the noise vector $\epsilon \in \mathbb{R}^n$ in the model $y = X\beta + \epsilon$ are generated independently from a Normal distribution with mean zero and a varying standard deviation $\sigma \in [0, 10]$.

Figure 6 (left) shows that, as expected, the accuracy of the recovered estimate of β decreases for all methods as σ increases. However, this increase seems rather slow. The runtime as a function of σ, depicted in Figure 6 (right), stays roughly constant for all methods, as expected.

C Proof of Proposition 1

Proof. The bounds on L_e^μ and L_s^μ follow from eq. (8) and eq. (10) after a direct calculation. In particular, for the entropy prox function,

$$\begin{aligned}
\sup_{\beta \in \mathbb{R}^p} |L_e^\mu(\beta) - L(\beta)| &= \sup_{\beta \in \mathbb{R}^p} \left| \lambda \sum_{i=1}^p f_s^\mu(\beta_i) - \lambda \sum_{i=1}^p |\beta_i| \right| \\
&\leq \sup_{\beta \in \mathbb{R}^p} \lambda \sum_{i=1}^p \left| f_s^\mu(\beta_i) - f(\beta_i) \right| \\
&\leq \sup_{\beta \in \mathbb{R}^p} \lambda \sum_{i=1}^p \mu \log(2) \\
&\leq \lambda p \mu \log(2),
\end{aligned}$$

where f is as defined in Section 2.2 and where it was used that $\lambda \geq 0$. The result for the squared error prox smoothed L_s^μ is proven analogously.

Since both f_e^μ and f_s^μ are convex according to (Nesterov, 2005, Theorem 1), and since the least squares term $\frac{1}{2}\|X\beta - y\|_2^2$ is convex, it follows that both L_e^μ and L_s^μ remain convex as the sum of two convex functions.

To be precise, strict convexity holds true. Observe that the second derivative of the entropy

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

smoothed absolute value of Section 2.2.1 is given by

$$\frac{\partial^2}{\partial z^2} f_e^\mu(z) = \frac{4e^{2x/\mu}}{\mu\left(e^{2x/\mu}+1\right)^2},$$

which is always positive, thus making f_e^μ strictly convex. Therefore, L_e^μ is strictly convex as the sum of a convex function and a strictly convex function. Similar arguments show that L_s^μ is strictly convex. □

References

Beck, A. and Teboulle, M. (2009). A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM J Imaging Sciences*, 2(1):183–202.

Chi, E., Goldstein, T., Studer, C., and Baraniuk, R. (2018). fasta: Fast Adaptive Shrinkage/Thresholding Algorithm. R-package version 0.1.0.

Daubechies, I., Defrise, M., and Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, 57(11):1413–1457.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Ann Stat*, 32(2):407–499.

Fan, J. and Li, R. (2001). Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *J Am Stat Assoc*, 96(456):1348–1360.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 33(1):1–22.

Hahn, G., Banerjee, M., and Sen, B. (2017). Parameter Estimation and Inference in a Continuous Piecewise Linear Regression Model.

Hahn, G., Lutz, S. M., Laha, N., and Lange, C. (2020). smoothedLasso: Smoothed LASSO Regression via Nesterov Smoothing. R-package version 1.3: `https://cran.r-project.org/package=smoothedLasso`.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.06.980953; this version posted March 22, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Hastie, T. and Efron, B. (2013). lars: Least Angle Regression, Lasso and Forward Stagewise. R-package version 1.2.

Khera, A. V., Chaffin, M., Aragam, K. G., Haas, M. E., Roselli, C., Choi, S. H., Natarajan, P., Lander, E. S., Lubitz, S. A., Ellinor, P. T., and Kathiresan, S. (2018). Genome-wide polygenic scores for common diseases identify individuals with risk equivalent to monogenic mutations. *Nature Genetics*, 50:1219–1224.

Mak, T., Porsch, R., Choi, S., Zhou, X., and Sham, P. (2016). Polygenic scores via penalized regression on summary statistics. *Genet Epidemiol*, 41(6):469–480.

Michelot, C. (1986). A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$. *J Optimiz Theory App*, 50(1):195–200.

Nesterov, Y. (1983). A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Dokl Akad Nauk SSSR*, 269(3):543–547.

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Math. Program. Ser. A*, 103:127–152.

NHLBI TOPMed (2018). Boston Early-Onset COPD Study in the National Heart, Lung, and Blood Institute (NHLBI) Trans-Omics for Precision Medicine (TOPMed) Program.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Stat Comp, Vienna, Austria.

Regan, E., Hokanson, J., Murphy, J., Make, B., Lynch, D., Beaty, T., Curran-Everett, D., Silverman, E., and Crapo, J. (2010). Genetic epidemiology of copd (copdgene) study design 2. *COPD*, 7:32–43.

Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *J Roy Stat Soc B Met*, 58(1):267–288.

Tibshirani, R. (2013). Model selection and validation 1: Cross-validation.

Wu, T., Chen, Y., Hastie, T., Sobel, E., and Lange, K. (2009). Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721.