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 Cho,
Edwin K. Silverman, and Christoph Lange

Abstract

High dimensional linear regression problems are often fitted using LASSO-type 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, leading to the following threefold contribution of this work: (1) We introduce a closed-form smoothed LASSO which preserves the convexity of the LASSO objective 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; (2) we prove that the estimates obtained for the smoothed LASSO problem can be made arbitrarily close to the ones of the original (unsmoothed) problem and provide explicit bounds on the accuracy of our obtained estimates; and (3) we propose an iterative algorithm to progressively smooth the LASSO objective function which increases accuracy and is virtually free of tuning parameters. Using simulation studies for polygenic risk scores based on 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, in particular the proposed progressive smoothing algorithm, provides estimates with equal or higher accuracy than the gold standard while guaranteeing a bound on its error. The computation time of our initial implementation of the progressive smoothing approach increases only by a constant factor in comparison to FISTA.

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 $X\beta = y$. 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. 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. The LASSO 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 $\|\beta\|_1$ 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 depend only on the smoothing parameter μ.

The contributions of our article are threefold: (1) We introduce a closed-form smoothed LASSO which allows for fast and efficient minimization with the help of explicit gradients; (2) we prove explicit error bounds on the difference between the minima of the smoothed and unsmoothed LASSO objective functions, and we show that the smoothed regression estimates can be made arbitrarily close to the ones of the unsmoothed LASSO; (3) starting with a high degree of smoothness, an iterative algorithm is proposed to progressively approximate the minimum of the unsmoothed LASSO objective function which facilitates minimization and yields superior accuracy in our simulation experiments. Since setting the smoothing parameter does not play a major role in the performance of the latter approach, it is virtually free of tuning parameters. We evaluate our algorithms in a detailed simulation study with respect to both accuracy and runtime.

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.

This article is structured as follows. We first provide a detailed literature review in Section 1.1 to highlight previous work, distinguish it from ours and emphasise the contribution of our article. Section 2 applies Nesterov smoothing to the LASSO objective function. We refine our approach in Section 3 by proposing a progressive and virtually tuning-free smoothing procedure for the LASSO, as well as by deriving guarantees of correctness on the obtained LASSO estimates. Using polygenic risk scores as an example, we evaluate the proposed methodology in simulation study in Section 4.

The article concludes with a discussion in Section 5. Details of Nesterov smoothing and all proofs are provided in the appendix.

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.

1.1 Literature review

Since the seminal publication of the LASSO in Tibshirani (1996), numerous approaches have focused on (smoothing) approaches to facilitate the minimization of the LASSO objective function. The following publications differ from our work in that they do not consider the same bounds on the accuracy of the unsmoothed and smoothed LASSO objective functions and their resulting minimizers which we present. Moreover, no progressive smoothing procedure yielding stable regression estimates is derived.

Fan and Li (2001) consider smoothing approaches for the LASSO L_1 penalty, the SCAD (Smoothly Clipped Absolute Deviation) penalty, and hard thresholding penalties. However, their smoothing approaches are not based on the Nesterov (2005) framework. Instead, the authors employ a quadratic approximation at the singularity of the penalties to achieve a smoothing effect, and they propose a one-step shooting algorithm for minimization. However, their main focus is on root-n consistency results of the resulting estimators and asymptotic normality results for the SCAD penalty, results which the authors state do not all apply to the LASSO.

Some smoothing approaches (Belloni et al., 2011; Chen et al., 2010a; Banerjee et al., 2008) build upon the first-order accelerated gradient descent algorithm of Nesterov (2005). Those variants of Nesterov's algorithm are iterative methods which are unrelated to our adaptive smoothing procedure. A detailed overview of several variants of the first-order accelerated gradient descent algorithm can be found in Becker and Candès (2011).

Beck and Teboulle (2012) can be regarded as an extension of the work of Nesterov (2005). The authors consider a more general smoothing framework which, as a special case, includes the same smoothing we establish for the absolute value in the L_1 penalty of the LASSO (though without the

guarantees we derive).

Haselimashhadi and Vinciotti (2016) smooth the absolute value in the L_1 penalty of the LASSO using Nesterov's technique in the same way as we do, and they state the same bound on the difference between the unsmoothed and smoothed objective functions taken from Nesterov's results. However, no results on the accuracy of the obtained minimizers are given. Importantly, Haselimashhadi and Vinciotti (2016) deviate from our work in that they enforce that the smoothed LASSO penalty passes through zero, leading the focus of their article to be on another smoothed LASSO approach which is based on the error function of the normal distribution.

Further work available in the literature employs Nesterov's smoothing techniques for a variety of specialized LASSO objective functions. For instance, Chen et al. (2010b) consider the group LASSO and employ Nesterov's formalism to smooth the LASSO penalty using the squared error proximity function which we also consider. Nevertheless, they focus on adapting Nesterov's first-order accelerated gradient descent algorithm in order to compute the LASSO regression estimate, whereas we focus on adaptive smoothing. Chen et al. (2012) also consider the group LASSO, separate out the simple nonsmooth L_1 penalty from the more complex structure-inducing penalties, and only smooth the latter. This leaves the L_1 norm on the parameters unchanged, thus still enforcing individual feature level sparsity.

The joint LASSO is considered in Dondelinger and Mukherjee (2020) who state an iterative minimization procedure which smoothes the LASSO penalty using Nesterov's techniques. The authors state closed form derivatives for the minimization, but no other theoretical results are given.

One variant of the original LASSO which has recently gained attention is the concomitant LASSO. The concomitant LASSO augments the original LASSO with a term $\sigma/2$ for which a second regularization parameter σ is introduced (Ndiaye et al., 2017). The parameter σ is meant to be decreased to zero. Smoothing the concomitant LASSO has the advantage that Nesterov's techniques do not need to be applied to the L_1 penalty. Instead, the smooth concomitant LASSO has a closed form expression which is different from the smoothed LASSO approaches we consider (Ndiaye et al., 2017), and results in the literature (Massias et al., 2018) are only named in analogy to the smoothing terminology introduced in Nesterov (2005).

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 A. In Section 2.1, 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 results of this section will be used in Section 3 to develop an adaptive procedure which iteratively smoothes the LASSO, yielding more stable estimates for linear regression than the approaches in this section, and to provide theoretical guarantees on the smoothed LASSO.

We briefly summarize the results of Section A. 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 $f(z) = \max_{i=1,\dots,k} \left(A[z,1]^\top\right)_i$, 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, f is replaced by an approximation f^μ which is both uniformly close to f and smooth (see Section A.1). 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 considered in Section A.2. Smoothing with the so-called entropy prox function results in the smooth approximation f_e^μ having a closed-form expression given by

$$f_e^\mu(z) = \mu \log\left(\frac{1}{k}\sum_{i=1}^{k} e^{\frac{\left(A[z,1]^\top\right)_i}{\mu}}\right), \tag{2}$$

which satisfies the uniform bound

$$\sup_{z\in\mathbb{R}^q} |f(z) - f_e^\mu(z)| \leq \mu \log(k). \tag{3}$$

Similarly, the smooth approximation of f with the help of the squared error prox function $\rho_s(w) = \frac{1}{2}\sum_{i=1}^{k}\left(w_i - \frac{1}{k}\right)^2$ for $w \in \mathbb{R}^k$ can be written as

$$f_s^{\mu}(z) = \langle \hat{c}(z), A[z,1]^{\top} \rangle - \mu\rho_s(\hat{c}(z)), \tag{4}$$

where $\hat{c}(z) \in \mathbb{R}^k$ 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 (see Section A.2.2). The approximation f_s^{μ} via squared error prox function satisfies the uniform bound

$$\sup_{z \in \mathbb{R}^q} |f(z) - f_s^{\mu}(z)| \leq \mu\left(1 - \frac{1}{k}\right). \tag{5}$$

2.1 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

$$L(\beta) = \frac{1}{n}\|X\beta - y\|_2^2 + \lambda\|\beta\|_1 \tag{6}$$

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

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

2.1.1 Entropy prox function

For the entropy prox function, eq. (2) with A as in Section 2.1 simplifies to

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

which according to eq. (3) satisfies the approximation bound

$$\sup_{z\in\mathbb{R}} |f(z) - f_e^\mu(z)| \leq \mu \log(2). \tag{7}$$

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. (6) with the entropy prox function results in

$$\begin{aligned}
L_e^\mu(\beta) &= \frac{1}{n}\|X\beta - y\|_2^2 + \lambda \sum_{i=1}^p f_e^\mu(\beta_i), \\
\frac{\partial L_e^\mu}{\partial \beta_i}(\beta) &= -\frac{2}{n}\langle y - X\beta, X_{\cdot,i}\rangle + \lambda g_e^\mu(\beta_i), \\
\frac{\partial^2 L_e^\mu}{\partial \beta_i \partial \beta_j}(\beta) &= \frac{2}{n}(X^\top X)_{ij} + \mathbb{I}(i=j)\cdot \lambda h_e^\mu(\beta_i),
\end{aligned} \tag{8}$$

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. (8) 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 $\partial L_e^\mu / \partial \beta_i$.

2.1.2 Squared error prox function

Similarly, eq. (4) with A as in Section 2.1 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. (5), we obtain the approximation bound

$$\sup_{z \in \mathbb{R}} |f(z) - f_s^\mu(z)| \leq \frac{1}{2}\mu. \tag{9}$$

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. (6) with the squared error prox function results in

$$\begin{aligned} L_s^\mu(\beta) &= \frac{1}{n}\|X\beta - y\|_2^2 + \lambda \sum_{i=1}^{p} f_s^\mu(\beta_i), \\ \frac{\partial L_s^\mu}{\partial \beta_i}(\beta) &= -\frac{2}{n}\langle y - X\beta, X_{\cdot,i} \rangle + \lambda g_s^\mu(\beta_i), \\ \frac{\partial^2 L_s^\mu}{\partial \beta_i \partial \beta_j}(\beta) &= \frac{2}{n}(X^\top X)_{ij} + \mathbb{I}(i = j) \cdot \lambda \left. \frac{\partial^2 f_s^\mu}{\partial z^2} \right|_{z = \beta_i} \end{aligned} \tag{10}$$

where as before, the squared error prox smoothed LASSO 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 2.1.1 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

Algorithm 1: Progressive smoothing

input: $X \in \mathbb{R}^{n \times p}$, $y \in \mathbb{R}^n$, $\lambda > 0$, $\mu_0 > 0$, $N \in \mathbb{N}$;

Set $\hat{\beta}_{N+1} \in \mathbb{R}^p$ randomly;

for $i = N, \ldots, 0$ **do**

  $\mu \leftarrow 2^i \mu_0$;

  $\hat{\beta}_i \leftarrow \arg\min_\beta L^\mu(\beta)$ with starting value $\hat{\beta}_{i+1}$, where L^μ is either L_e^μ or L_s^μ which both depend on X, y, and λ;

end

return $\hat{\beta}_0$;

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.

3 Progressive smoothing and theoretical guarantees of correctness

This section proposes an adaptive smoothing technique for the LASSO in Section 3.1 which will be shown in the simulations of Section 4 to yield stable estimators for linear regression. Moreover, theoretical guarantees are derived in Section 3.2. The results bound the error of the unsmoothed to the smoothed LASSO objective functions, as well as the L_2 distance between the estimators obtained from either the unsmoothed on smoothed LASSO, thus giving additional validity to our approach.

3.1 Progressive smoothing

Instead of solving the smoothed LASSO problem $\hat{\beta} = \arg\min_\beta L^\mu(\beta)$ directly for some $\mu > 0$, where L^μ denotes either L_e^μ or L_s^μ, we employ a progressive smoothing procedure along the following rationale: We start with a large value of the smoothing parameter μ to facilitate the minimization. After computing $\hat{\beta}$, we decrease the smoothing parameter and repeat the minimization using the previously found minimizer as the new starting value. This approach is based on the heuristic idea that as μ decreases and the smoothed LASSO objectives L_e^μ or L_s^μ approach L (see Proposition 1 below), the found minimizers in each iteration remain close to each other and converge to the minimizer of L.

Algorithm 1 formalizes our approach. The input of the algorithm are the input matrix $X \in \mathbb{R}^{n \times p}$, the response $y \in \mathbb{R}^n$, and the LASSO parameter $\lambda > 0$ which are implicitly used in the

smoothed LASSO functions L_e^μ or L_s^μ. We also specify a target smoothing parameter $\mu_0 > 0$ and a number of smoothing steps $N \in \mathbb{N}$.

After initializing a random starting value $\hat{\beta}_{N+1} \in \mathbb{R}^p$ for the first minimization, we gradually decrease the degree of smoothness according to $\mu = 2^i \mu_0$ from $i = N$ to the target level μ_0 at $i = 0$. In each iteration i, we compute a new estimate $\hat{\beta}_i$ using the current smoothing level μ and the previous estimate $\hat{\beta}_{i+1}$ as the starting value. The output of the algorithm is $\hat{\beta}_0$, the LASSO parameter estimate corresponding to the target smoothing degree μ_0.

Importantly, the advantage of Algorithm 1 consists in the fact that the precise specification of the smoothing parameter does not play a major role. It suffices to start with any sufficiently large value (that is, $2^N \mu_0 \gg 1$) and to end the iteration with any sufficiently small value μ_0, for instance of the order of the machine precision or of the square root of the machine precision. This effectively makes Algorithm 1 free of tuning parameters. The choice of the LASSO regularization parameter λ remains problem specific and is thus left to the user.

3.2 Theoretical guarantees

The bounds on both f_e^μ in eq. (7) and f_s^μ in eq. (9) carry over to a bound on the overall approximation of the LASSO objection function of eq. (6):

Proposition 1. *The entropy and squared error prox smoothed LASSO objective functions in eqs.* (8) *and* (10) *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 $\mu \to 0$.

Following (Seijo and Sen, 2011, Lemma 2.9), the following proposition shows that the uniform proximity (in the supremum norm) of the unsmoothed and smoothed LASSO objective functions

implies that their global minimizers also converge to each other in the supremum norm metric.

Proposition 2. *Let $f_1 : \mathbb{R}^s \to \mathbb{R}$ be continuous and strictly convex for $s \in \mathbb{N}$. Then $x_1 = \arg\min_{x\in\mathbb{R}^s} f_1(x)$ is continuous at f_1 with respect to the supremum norm.*

Proposition 2 states that $\sup_{\beta\in\mathbb{R}^p} |L_e^\mu(\beta) - L(\beta)| \to 0$ for $\mu \to 0$ implies that the minimizers of L and L_e^μ converge to each other in the supremum norm. Similarly, the same result holds true for L_s^μ. This result is stronger than the one of (Beck and Teboulle, 2009, Theorem 4.4), who prove that the FISTA method finds a minimizer which is of similar quality than the true minimizer.

Although Proposition 2 shows convergence, it does not give an explicit error bound on the distance between the two minimizers. This is done in the next result.

Proposition 3. *Let $s \in \mathbb{N}$ and $\epsilon > 0$. Let $f_1 : \mathbb{R}^s \to \mathbb{R}$ be differentiable and strictly convex. Let $f_2 : \mathbb{R}^s \to \mathbb{R}$ be such that $\sup_{x\in\mathbb{R}^s} |f_1(x) - f_2(x)| \leq \epsilon$. Let $x_i = \arg\min_{x\in\mathbb{R}^s} f_i(x)$ be the two minimizers for $i \in \{1, 2\}$. Then for any $\delta > 0$ and any $y_1 \in \mathbb{R}^s$ satisfying $y_1 \neq x_1$ and $\|y_1 - x_1\|_2 \leq \delta$, there exist two constants $C_\delta > 0$ and $L_\delta > 0$ independent of x_2 such that*

$$\|x_1 - x_2\|_2 \leq C_\delta \left[\|\nabla f_1(y_1)\|_2^{-1} (\delta L_\delta + 2\epsilon) + \delta\right].$$

Note that Proposition 3 does not generalize to non strictly convex functions. Applying Proposition 3 with f_1 taken to be the differentiable and strictly convex $L_e^\mu(\beta)$ and f_2 taken to be $L(\beta)$ immediately gives an explicit bound on the distance between the two minimizers. As before, the same result holds true when taking f_1 to be L_s^μ.

4 Simulation studies for polygenic risk scores for COPD

In this section, we evaluate four approaches to compute LASSO estimates using simulated data (Section 4.1) 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 4.2). Of the four approaches we consider, the first two utilize existing methodology, while the last two approaches implement the methodologies 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);

3. we minimize the smoothed LASSO objective function of eq. (8) using its explicit gradient;

4. we employ the progressive smoothing approach of Section 3.1. As suggested at the end of Section 3.1, we set the target smoothing parameter to $\mu_0 = 2^{-6}$ and employ $N = 9$ smoothing steps (thus implying an initial value of the smoothing parameter of $\mu = 2^{-6} \cdot 2^9 = 8$).

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. (6). Additionally, we employ a uniform random starting value as done for our own approaches (unsmoothed and smoothed LASSO, as well as progressive smoothing). 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 choice of the LASSO regularization parameter λ varies in each experiment and is given individually.

4.1 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 variance σ^2. The smaller the variance, the easier the recovery of β will be. We will employ $\sigma^2 = 0.1$ in our 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.

The regularization parameter of the LASSO was chosen as $\lambda = 1$ for the experiments in this subsection.

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. Both FISTA and the progressive smoothing approach yield stable estimates for all n. Progressive smoothing achieves better estimates for $n < p$, although FISTA slightly outperforms it for $n > p$. As expected, all methods become more accurate as the number of data points n increases. The smoothed LASSO approach, the progressive smoothing algorithm and FISTA roughly draw equal in accuracy for large n.

Figure 1 (right) shows that the unsmoothed and smoothed LASSO approaches have an almost identical runtime. The progressive smoothing approach essentially calls the smoothed LASSO algorithm a fixed number of times, and is thus a constant factor slower than the other approaches.

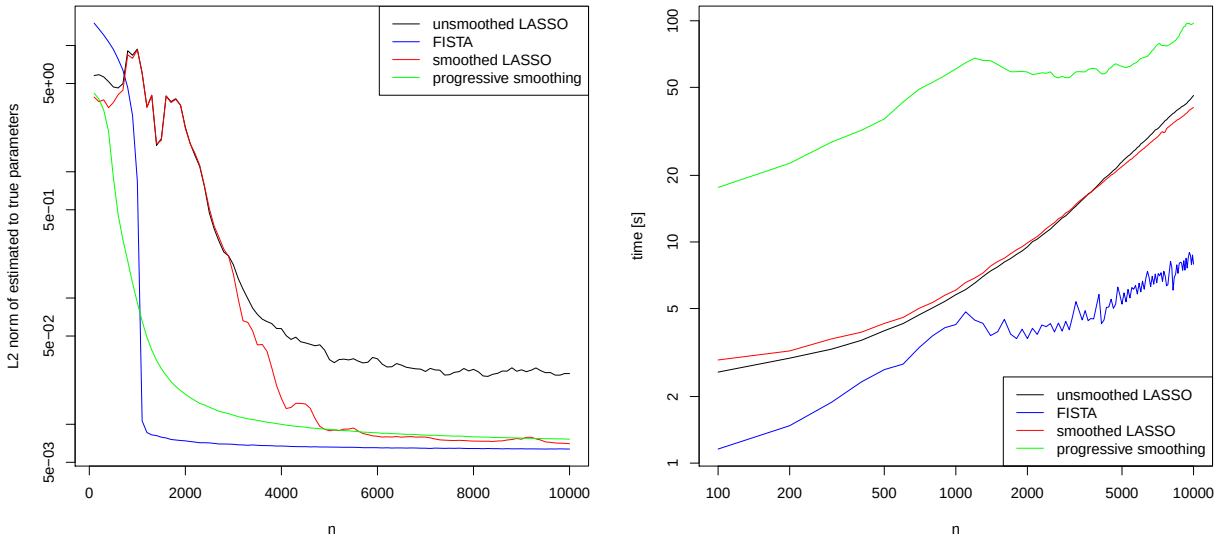

Figure 1: L_2 norm of parameter estimate to truth (left) and runtime in seconds (right) as a function of $n \in [1, 10000]$ while $p = 1000$. Logarithmic scale on the y-axes.

This seems to be a reasonable trade-off for the considerably improved accuracy that the progressive smoothing provides. We find that although highly optimized, FISTA is only a low multiple factor faster than our approaches.

Similarly to the previous experiment, in Figure 2 (left) we keep $n = 1000$ fixed and investigate the dependence of all four approaches on $p \in [1, 5000]$. Here, the unsmoothed and smoothed LASSO approaches seem to suffer from numerical instabilities, while the progressive smoothing approach finds good quality solutions much more reliably. Notably, progressive smoothing seems to outperform FISTA for $p > n$. As expected, while keeping the data size n fixed, estimation becomes more challenging for all methods as p increases.

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 as expected, the progressive smoothing approach is roughly a constant factor slower. FISTA is again slightly faster than the other approaches. Importantly, the scaling of the runtimes seems to be roughly equal for all four methods.

Since the unsmoothed LASSO does not come with the guarantees we established for our smoothing approach (see Section 3.2) and never outperforms any of the two smoothing approaches in

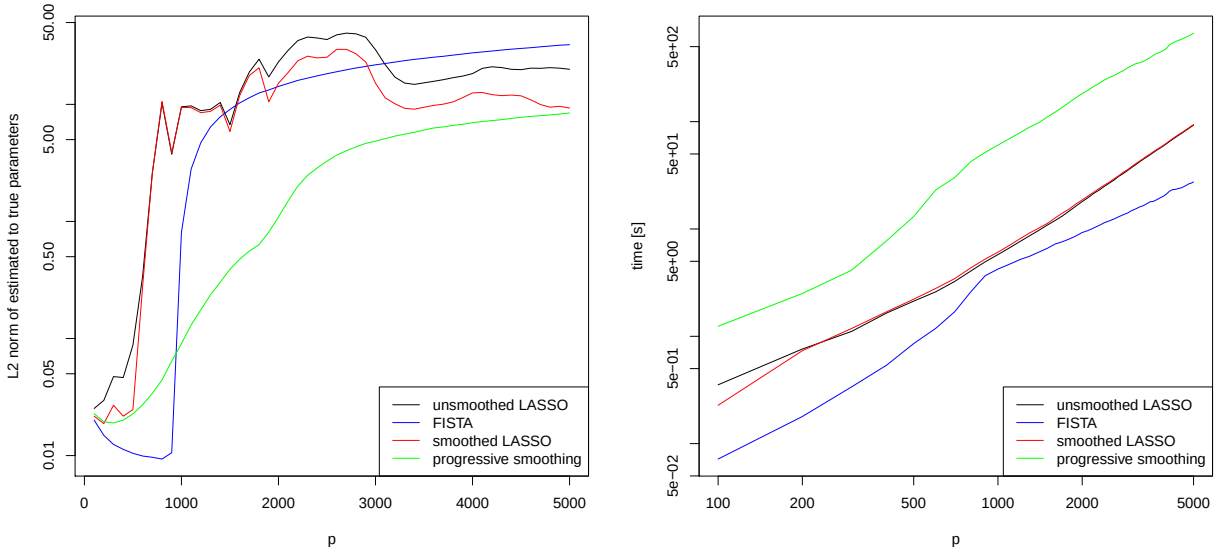

Figure 2: L_2 norm of parameter estimate to truth (left) and runtime in seconds (right) as a function of $p \in [1, 5000]$ while $n = 1000$. Logarithmic scale on the y-axes.

the previous experiments, we will focus in the remainder of the simulations on FISTA, smoothed LASSO, and progressive smoothing.

4.2 Application to polygenic risk scores

We evaluate the smoothed LASSO approaches 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.

method	L_2 norm	runtime [s]
FISTA	284.6	66.6
smoothed LASSO	26.3	116.6
progressive smoothing	33.6	373.6

Table 1: 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.

The regularization parameter of the LASSO was chosen as $\lambda = 0.05$ for the following two experiments.

4.2.1 Results from a single run

We solve $y = X\beta$ for the given X and y using our smoothed LASSO approach of eq. (8), as well as the progressive smoothing approach of Section 3.1, and compare both to the FISTA algorithm.

Table 1 shows results for a single application of the three 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 LASSO and progressive smoothing. However, in this experiment it turns out that a single application of the smoothed LASSO is actually advantageous over the progressive smoothing approach. Not surprisingly, progressive smoothing takes a constant factor longer due to its repeated application of the smoothed LASSO, see Section 4.1. The FISTA method beats our approaches in terms of runtime.

4.2.2 Cross validation

To quantify the accuracy of our approaches 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 standard choices of its tuning parameters seems to have difficulties to converge to the minimum

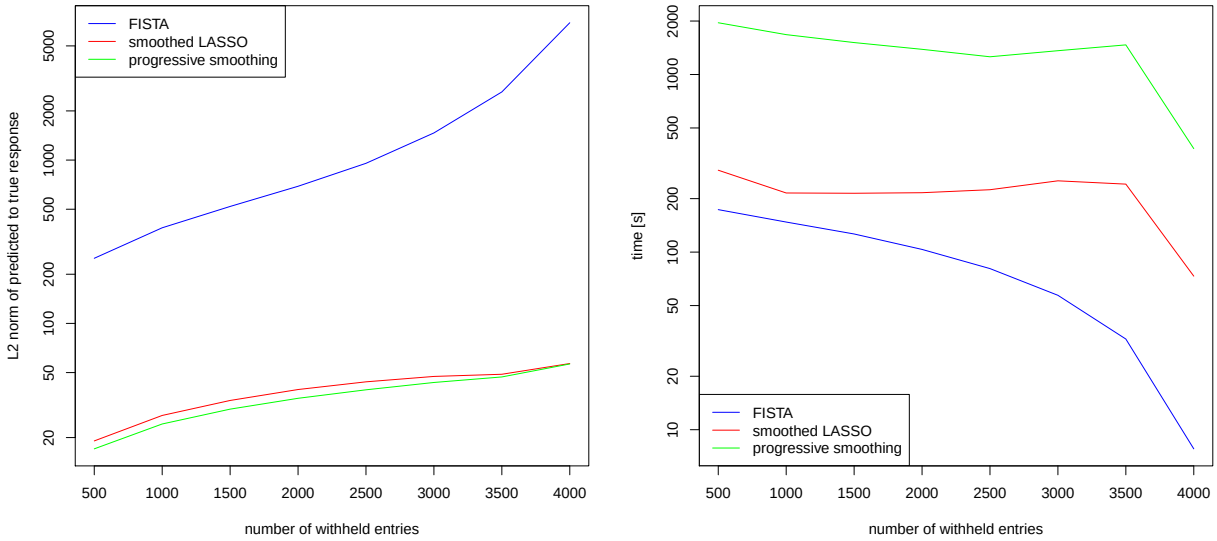

Figure 3: L_2 norm of predicted to withheld data in cross-validation (left) and runtime in seconds (right) as a function of the number of withheld entries. Dataset of polygenic risk scores of X. Logarithmic scale on the y-axes.

of the LASSO objective function, whereas smoothed LASSO and progressive smoothing perform better. Not surprisingly, the quality of the prediction becomes worse in general for any method as the number of withheld entries increases, since predictions are based on fewer and fewer datapoints.

Interestingly, the progressive smoothing approach is not as powerful here as it was in Section 4.1, as the simple smoothed LASSO and progressive smoothing basically draw equal in this experiment.

Finally, Figure 3 (right) displays runtime measurements for all three approaches. We observe that our two smoothing approaches seem to be rather insensitive to the number of withheld entries apart for a very large number of withheld entries. As usual, progressive smoothing is a constant factor slower than the smoothed LASSO. The FISTA algorithm is the fastest method, and moreover it exhibits a greater sensitivity to the number of withheld entries, that is the size of the estimation problem.

4.3 Application to synthetic polygenic risk scores

We aim to extend the simulations of Section 4.2 in order to vary the sparsity of the parameter estimate $\hat{\beta}$. To this end, we change the simulation setting as follows. Leaving X unchanged, we simulate a parameter vector $\beta \in \mathbb{R}^p$ in which $nz \in \{0, \dots, p\}$ entries are drawn from a Beta

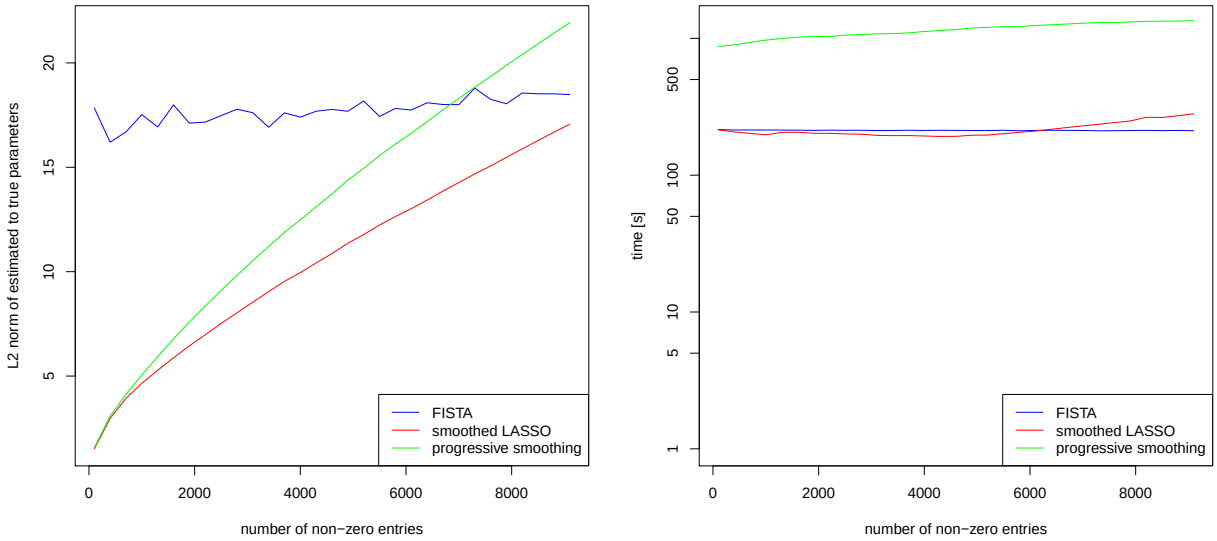

Figure 4: L_2 norm of parameter estimate to truth (left) and runtime in seconds (right) as a function of the number of non-zero entries in the simulated true parameter vector. Dataset of polygenic risk scores of X. Logarithmic scale on the y-axis of the time plot.

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, smoothed LASSO, and progressive smoothing 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 smoothed LASSO and progressive smoothing yield considerably more stable estimates than FISTA (as expressed through a lower deviation in L_2 norm). Only for very dense vectors (having a number of non-zero entries of more than 7000 out of $p = 9153$), we observe that FISTA draws equal with the smoothed LASSO approaches. Surprisingly, progressive smoothing performs less well than the simple smoothed LASSO in this experiment.

Figure 4 (right) shows runtime results in seconds for all three approaches. The runtime scalings of all methods seem to be rather insensitive to the simulation scenario. Interestingly, smoothed LASSO draws equal in speed with the FISTA algorithm, while progressive smoothing is again a

constant factor slower.

Overall, we conclude from the simulations that the smoothed LASSO and the progressive smoothing approach of Section 3.1 yield stable LASSO estimates which often outperform FISTA, exhibit roughly the same runtime scaling as FISTA, and come with a guarantee on their accuracy.

We suggest employing the progressive smoothing approach with the target smoothing parameter $\mu_0 = 2^{-N}$ set to a small value, for instance of the order of the machine precision or the square root of the machine precision, and N chosen such that the initial value of the smoothing parameter is sufficiently large, meaning $2^N \mu_0 \gg 1$. This will make the progressive smoothing algorithm essentially independent of the choice of its smoothing parameter and thus free of tuning parameters. Either the entropy or the squared error prox function can be employed within progressive smoothing.

5 Discussion

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

Most importantly, the presented approach comes with two guarantees. First, a uniform bound on the distance between the unsmoothed and smoothed LASSO functions is guaranteed. This distance can be made arbitrarily small. Second, we show that the uniform closeness of the unsmoothed and smoothed objective functions translates to an explicit bound on the norm between the minimizers of the unsmoothed and smoothed LASSO objective functions. Since we can carry out the latter optimization efficiently, our approach yields easily computable LASSO regression estimates which are guaranteed to be close to the actual estimates obtained had we minimized the original LASSO objective.

Simulations show that our proposed progressive smoothing algorithm yields equally reliable or more reliable estimates than the gold standard in the literature, the FISTA algorithm of Beck and Teboulle (2009), while (a) being essentially free of tuning parameters, (b) having roughly the same runtime scaling, and (c) coming with a guarantee on the accuracy of its regression estimates.

A Nesterov smoothing

This section follows (Nesterov, 2005, Sections 2 and 4). It introduces the basic formalism of Nesterov smoothing in Section A.1 and concretizes the approach in Section A.2.

A.1 Description 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). The function f can be expressed as

$$f(z) = \max_{i=1,\dots,k} \left(A[z,1]^\top \right)_i , \tag{11}$$

where in the remainder of the section, $A \in \mathbb{R}^{k\times(q+1)}$ is a matrix whose rows contain the linear coefficients for each of the k 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 $\|\cdot\|_k$ be a norm on $\mathbb{R}^k$ and $\langle\cdot,\cdot\rangle$ be the Euclidean inner product. Define the unit simplex $Q_k \subseteq \mathbb{R}^k$ as

$$Q_k = \left\{ w = (w_1,\dots,w_k) \in \mathbb{R}^k : \sum_{i=1}^k w_i = 1, \text{ and } w_i \geq 0 \text{ for all } i = 1,\dots,k \right\}.$$

To introduce the smoothing procedure, Nesterov (2005) first defines a proximity function, or prox function, on Q_k. A prox function ρ is any nonnegative, continuously differentiable, and strongly convex function (with respect to the norm $\|\cdot\|_k$). The latter means that ρ satisfies

$$\rho(s) \geq \rho(t) + \langle \nabla\rho(t), t-s\rangle + \frac{1}{2}\|t-s\|_k$$

for all $s,t \in Q_k$.

For any $\mu > 0$, consider the function

$$f^\mu(z) = \max_{w\in Q_k} \left\{ \langle A[z,1]^\top, w\rangle - \mu\rho(w) \right\}. \tag{12}$$

According to (Nesterov, 2005, Theorem 1), the function f^μ defined in eq. (12) is convex and everywhere differentiable in z for any $\mu > 0$. The function f^μ depends only on the parameter μ controlling the degree of smoothness. For $\mu = 0$, we recover the original unsmoothed function since $f^0(z) = \max_{w \in Q_k} \left\{ \langle A[z, 1]^\top, w \rangle \right\} = f(z)$. The gradient $z \mapsto \frac{\partial}{\partial z} f^\mu(z)$ is Lipschitz continuous with a Lipschitz constant that is proportional to μ^{-1}. A closed form expression of both the gradient and the Lipschitz constant are given in (Nesterov, 2005, Theorem 1).

Importantly, the function f^μ is a uniform smooth approximation of $f = f^0$ since

$$f^0(z) - \mu \sup_{w \in Q_k} \rho(w) \leq f^\mu(z) \leq f^0(z) \tag{13}$$

for all $z \in \mathbb{R}^q$, meaning that the approximation error is uniformly upper bounded by

$$\sup_{z \in \mathbb{R}^q} |f(z) - f^\mu(z)| \leq \mu \sup_{w \in Q_k} \rho(w) = O(\mu). \tag{14}$$

Indeed, eq. (13) holds true since for all $z \in \mathbb{R}^q$,

$$\begin{aligned} f^\mu(z) &\geq \sup_{w \in Q_k} \langle A[z, 1]^\top, w \rangle - \mu \sup_{w \in Q_k} \rho(w) = f^0(z) - \mu \sup_{w \in Q_k} \rho(w), \\ f^\mu(z) &= \sup_{w \in Q_k} \Big\{ \langle A[z, 1]^\top, w \rangle - \mu \rho(w) \Big\} \leq \sup_{w \in Q_k} \langle A[z, 1]^\top, w \rangle = f^0(z), \end{aligned}$$

where it was used that both the function ρ and the parameter μ are nonnegative.

A.2 Two choices for the proximity function

We consider two choices of the prox function ρ.

A.2.1 Entropy prox function

The entropy prox function $\rho_e : \mathbb{R}^k \to \mathbb{R}$ is given by

$$\rho_e(w) = \sum_{i=1}^{k} w_i \log(w_i) + \log(k)$$

for $w \in \mathbb{R}^k$.

Setting the norm $\|\cdot\|_k$ as the L_1 norm in $\mathbb{R}^k$, Nesterov (2005) shows that ρ_e is strongly convex with respect to the L_1 norm and satisfies $\sup_{w\in Q_k} \rho_e(w) = \log(k)$, see (Nesterov, 2005, Lemma 3). Using eq. (14), we obtain the uniform bound

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

for the entropy smoothed function f_e^μ obtained by using ρ_e in eq. (12). Interestingly, smoothing with the entropy prox function admits a closed-form expression of f_e^μ given by

$$f_e^\mu(z) = \max_{w\in Q_k} \left\{ \sum_{i=1}^k w_i \left(A[z,1]^\top\right)_i - \mu \left(\sum_{i=1}^k w_i \log(w_i) + \log(k) \right) \right\} = \mu \log \left(\frac{1}{k} \sum_{i=1}^k e^{\frac{\left(A[z,1]^\top\right)_i}{\mu}} \right),$$

see (Nesterov, 2005, Lemma 4).

A.2.2 Squared error prox function

The squared error prox function is given by

$$\rho_s(w) = \frac{1}{2} \sum_{i=1}^k \left(w_i - \frac{1}{k} \right)^2.$$

Mazumder et al. (2019) show that the optimization in eq. (12) with squared error prox function is equivalent to the convex program

$$f_s^\mu(z) = \min_{w\in Q_k} \left(\frac{1}{k} \sum_{i=1}^k w_i^2 - \sum_{i=1}^k w_i c_i(z) \right), \tag{15}$$

where $c_i(z) = 1/\mu \cdot \left(A[z,1]^\top\right)_i - 1/k$ depends on A and μ and is defined for any $i \in \{1,\ldots,k\}$. The problem in eq. (15) is equivalent to finding the Euclidean projection of the vector $c(z) = (c_1(z),\ldots,c_k(z))$ onto the k-dimensional unit simplex Q_k. This projection can be carried out efficiently using the algorithm of Michelot (1986), for which a computationally more efficient version was proposed in Wang and Carreira-Perpiñán (2013) that we use in our implementations. Denoting the Euclidean projection of the vector $c(z)$ onto Q_k as vector $\hat{c}(z)$, the squared error prox

approximation of f can be written as

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

As $\sup_{w \in Q_k} \rho_s(w) = 1 - \frac{1}{k}$ (Nesterov, 2005, Section 4.1), we obtain the uniform bound

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

for the squared error smoothing approach.

B Proofs

Proof of Proposition 1. The bounds on L_e^μ and L_s^μ follow from eq. (7) and eq. (9) after a direct calculation.

Since both f_e^μ and f_s^μ are convex according to (Nesterov, 2005, Theorem 1), see also Section A.1, it follows that both L_e^μ and L_s^μ remain convex. To be precise, the second derivative of the entropy smoothed absolute value (Section 2.1.1) is given by

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

and hence always positive, thus making f_e^μ strictly convex. From the LARS objective function (Efron et al., 2004) we know that the part $\frac{1}{2}\|X\beta - y\|_2^2$ is strictly convex as well, thus making eq. (8) in fact strictly convex. Similar arguments show that eq. (10) is strictly convex. □

Proof of Proposition 2. Since f_1 is continuous and strictly convex, it lays in the Skorohod topology $\mathcal{D}_K$ as defined in (Seijo and Sen, 2011, Definition 2.2). According to (Seijo and Sen, 2011, Lemma 2.9), the argmax functional is continuous at f_1 with respect to the supremum norm metric. □

Proof of Proposition 3. Since f_1 is differentiable, we know that ∇f_1 exists. Since f_1 is strictly convex, the minimum x_1 is unique and $\nabla f_1(y_1) \neq 0$ as $y_1 \neq x_1$. Since f_1 is also convex, the tangent

at every point stays below the function. Thus considering the tangent at y_1 we have for all z_0 that

$$f_1(y_1) + \nabla f_1(y_1)^\top (z_0 - y_1) \leq f_1(z_0),$$

and thus we can bound $f_1 - \epsilon$ from below as

$$f_1(y_1) + \nabla f_1(y_1)^\top (z_0 - y_1) - \epsilon \leq f_1(z_0) - \epsilon.$$

Observe that at x_1 we have $f_2(x_1) \in [f_1(x_1) - \epsilon, f_1(x_1) + \epsilon]$, and similarly at x_2 we have $f_2(x_2) \in [f_1(x_2) - \epsilon, f_1(x_2) + \epsilon]$. Thus for any z satisfying

$$f_1(y_1) + \nabla f_1(y_1)^\top (z - y_1) - \epsilon = f_1(x_1) + \epsilon, \tag{16}$$

we know that the minimum x_2 of f_2 cannot be further away from x_1 than z, thus $\|x_1 - x_2\|_2 \leq \|x_1 - z\|_2$. The quantity z satisfying eq. (16) is not unique, and thus without loss of generality we choose z such that $\nabla f_1(y_1)$ and $z - y_1$ are not orthogonal. Rewriting $z - y_1$ in eq. (16) as $z - x_1 + x_1 - y_1$ and rearranging terms yields

$$\nabla f_1(y_1)^\top (z - x_1) = f_1(x_1) - f_1(y_1) + 2\epsilon - \nabla f_1(y_1)^\top (x_1 - y_1).$$

Rewriting the non-zero scalar product on the left hand side as $\|\nabla f_1(y_1)\|_2 \cdot \|z - x_1\|_2 \cdot \cos(\theta)$ for some $\theta \in [0, \pi/2)$ and applying the L_2 norm on both sides yields, after applying the triangle inequality on the right hand side,

$$\|\nabla f_1(y_1)\|_2 \cdot \|z - x_1\|_2 \cdot |\cos(\theta)| \leq \|f_1(x_1) - f_1(y_1)\|_2 + 2\epsilon + \|\nabla f_1(y_1)\|_2 \cdot \|x_1 - y_1\|_2,$$

which after rearranging yields

$$\|z - x_1\|_2 \leq \frac{\|f_1(x_1) - f_1(y_1)\|_2 + 2\epsilon + \|\nabla f_1(y_1)\|_2 \cdot \|x_1 - y_1\|_2}{|\cos(\theta)| \cdot \|\nabla f_1(y_1)\|_2}.$$

We write $|\cos(\theta)|^{-1} = C_\delta$ and note that θ is determined by z and y_1 but independent of x_2. Since x_1 and y_1 are fixed, and f_1 is differentiable, it is also locally Lipschitz in a ball around x_1 that

includes y_1 (note that the Lipschitz parameter is independent of x_2). Thus there exists $L_\delta > 0$ such that $\|f_1(x_1) - f_1(y_1)\|_2 \leq L_\delta \|x_1 - y_1\|_2$. Using that $\|x_1 - y_1\|_2 \leq \delta$ by construction of y_1, we obtain

$$\|z - x_1\|_2 \leq C_\delta \left[\|\nabla f_1(y_1)\|_2^{-1} (\delta L_\delta + 2\epsilon) + \delta \right].$$

Since $\|x_1 - x_2\|_2 \leq \|x_1 - z\|_2$, the result follows. □

References

Banerjee, O., Ghaoui, L. E., and d'Aspremont, A. (2008). Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data. *Journal of Machine Learning Research*, 9:485–516.

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

Beck, A. and Teboulle, M. (2012). Smoothing And First Order Methods: A Unified Framework. *Siam J Optim*, 22(2):557–580.

Becker, S. R. and Candès, E. J. (2011). Templates for convex cone problems with applications to sparse signal recovery. *Math Prog Comp*, 3:165–218.

Belloni, A., Chernozhukov, V., and Wang, L. (2011). Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806.

Chen, X., Kim, S., Lin, Q., Carbonell, J. G., and Xing, E. P. (2010a). Graph-Structured Multi-task Regression and an Efficient Optimization Method for General Fused Lasso. *arXiv:1005.3579*, pages 1–21.

Chen, X., Lin, Q., Kim, S., Carbonell, J. G., and Xing, E. P. (2010b). An efficient proximal gradient method for general structured sparse learning. *Journal of Machine Learning Research*, 11.

Chen, X., Lin, Q., Kim, S., Carbonell, J. G., and Xing, E. P. (2012). Smoothing proximal gradient method for general structured sparse regression. *Ann Appl Stat*, 6(2):719–752.

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.

Dondelinger, F. and Mukherjee, S. (2020). The joint lasso: high-dimensional regression for group structured data. *Biostatistics*, 21:219–235.

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. http://www.cantab.net/users/ghahn/preprints/PhaseRegMultiDim.pdf.

Haselimashhadi, H. and Vinciotti, V. (2016). A Differentiable Alternative to the Lasso Penalty. *arXiv:1609.04985*, pages 1–12.

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

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.

Massias, M., Fercoq, O., Gramfort, A., and Salmon, J. (2018). Generalized Concomitant Multi-Task Lasso for Sparse Multimodal Regression. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain*, volume 84. PMLR.

Mazumder, R., Choudhury, A., Iyengar, G., and Sen, B. (2019). A Computational Framework for Multivariate Convex Regression and Its Variants. *J Am Stat Assoc*, 114(525):318–331.

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.

Ndiaye, E., Fercoq, O., Gramfort, A., Leclère, V., and Salmon, J. (2017). Efficient Smoothed Concomitant Lasso Estimation for High Dimensional Regression. In *7th International Conference on New Computational Methods for Inverse Problems*.

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.

Seijo, E. and Sen, B. (2011). A continuous mapping theorem for the smallest argmax functional. *Electron J Stat*, 5:421–439.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J Roy Stat Soc B Met*, 58(1):267–288.

Wang, W. and Carreira-Perpiñán, M. (2013). Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv:1309.1541*, pages 1–5.

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.