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 and Christoph Lange

March 2020

Abstract

We consider solving a high dimensional linear regression problem, using LASSO to account for sparsity. Though 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 suggest to use Nesterov smoothing of the LASSO objective function which enables us to compute closed form derivatives for efficient and fast minimization. The contribution of this work is threefold: (1) We propose an algorithm to efficiently compute estimates of the LASSO regression parameters; (2) we prove explicit bounds on the accuracy of the obtained estimates which show that the estimates obtained through the smoothed problem can be made arbitrary close to the ones of the original (unsmoothed) LASSO problem; (3) we propose an iterative procedure of the proposed approach to progressively smooth the objective function, which facilitates minimization and increases accuracy. Using simulation studies for polygenic risk scores based on genetic data from a genome-wide association study (GWAS) for COPD, we evaluate accuracy and runtime of the proposed approaches and compare them with the current standard in the literature. The results of the simulation studies suggest that the proposed methodology, progressive smoothing, provides estimates with higher accuracy than standard approaches that rely on GLMNET and that the differences are of practical relevance, while the computation time for progressive smoothing increases only be a constant factor, compared to GLMNET.

1 Introduction

Many substantive research questions in health, economic and social science require solving a classical linear regression problem $X\beta = y$, where 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, i.e. the least squares estimate $\arg\min_\beta \|X\beta - y\|_2^2$. However, their lack of robustness as well as the lack of sparsity in high dimensional settings with $p \gg n$, are two problems that have led to

the development of alternative estimation approaches, e.g. *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

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

where $\|\cdot\|_2$ is the Euclidean norm, $\|\cdot\|_1$ is the L_1 norm, and $\lambda > 0$ is a tuning parameter 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, which 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 exploring the idea of using smoothing. We transform the LASSO objective function into a function that is differentiable everywhere. We apply Nesterov smoothing (Nesterov, 2005) to the non-differentiable $\|\beta\|_1$ term in eq. (1), which will result in an approximation of the LASSO objective function that has the desired features. The Nesterov formalism depends on a smoothing parameter μ that controls the smoothness level. The 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 which we will use for a gradient descent algorithm; and (3) it provides uniform error bounds on the discrepancy between the smoothed and the original objective functions that depend only on μ.

The contributions of our article are threefold: (1) Using the uniform Nesterov bound, we propose an algorithm to efficiently compute a LASSO regression estimate with the help of explicit gradients; (2) we prove explicit error bounds on the minimizer of the smoothed LASSO objective function (compared with the minimizer of the unsmoothed LASSO objective function), showing that the smoothed regression estimate can be made arbitrary close to the one of the unsmoothed LASSO; (3) starting with a high degree of smoothness, an iterative procedure is proposed to progressively approximate the minimum of the unsmoothed function, which facilitates minimization and yields superior accuracy in our experiments. We evaluate our algorithms in a detailed simulation study with respect to both accuracy and runtime. Additionally, we provide guidance on how to set the tuning parameter μ.

In Fan and Li (2001), the authors likewise consider smoothing approaches for the LASSO L_1, SCAD and hard thresholding penalities. They employ a quadratic approximation at the singularity of the objective functions for smoothing and propose a "one-step" shooting algorithm for minimization. Root-n consistency of the resulting estimators is proven. In contrast to our work, the authors do not consider bounds on the accuracy of the smoothed minimizer, and no progressive smoothing

procedure is derived.

This article is structured as follows. Section 2 provides details on Nesterov smoothing and its application to the LASSO objective function. We refine our approach in Section 3 by proposing an adaptive smoothing procedure for the LASSO and by deriving guarantees of correctness on the obtained LASSO estimates. Using polygenic risk scores for COPD as the toy example, we evaluate the proposed methodology in simulation study in Section 4. To be precise, the simulations evaluate accuracy and runtime of all methods on real genetic data from a genome-wide association study for COPD. The article concludes with a discussion in Section 5. All proofs are provided in the appendix. Throughout the article, X_i denotes the ith column of a matrix X, $X_{I,\cdot}$ (and y_I) denote the submatrix (subvector) consisting of all rows (entries) indexed in the set I, $X_{-I,\cdot}$ (and y_{-I}) denote the submatrix (subvector) consisting of all rows (entries) except for those indexed in the set I, $|\cdot|$ denotes the absolute value, and $\|\cdot\|_2$ as well as $\|\cdot\|_\infty$ denote the L_2 and supremum norms, respectively.

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) (see Section 1). To be precise, we propose to replace the L_1 penalty term $\|\beta\|_1 = \sum_{i=1}^p |\beta_i|$ with a substitute having two properties: the substitute is uniformly close to the original L_1 penalty, and it is everywhere differentiable. The L_2 term $\frac{1}{2}\|X\beta - y\|_2^2$ in eq. (1) is smooth and will stay unchanged. Together, this will yield a smoothed LASSO substitute with two desirable properties: the substitute is everywhere differentiable, thus allowing for fast minimization using gradients, and it is uniformly close to the unsmoothed LASSO, meaning that we can use it in lieu of the original LASSO while being able to guarantee that any linear regression fit obtained with the smoothed LASSO is not too different from the one obtained had we minimized the original LASSO objective function.

The substitute of the term $\|\beta\|_1 = \sum_{i=1}^p |\beta_i|$ will be computed with the help of a technique called Nesterov smoothing. The Nesterov smoothing approach is in fact applicable to a large class of functions, and its general idea will be introduced in Section 2.1. After some technicalities in Section 2.2, we simplify the general Nesterov smoothing approach for the particular case of the LASSO penalty in Section 2.3 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.

The following two subsections (Sections 2.1 and 2.2) follow (Nesterov, 2005, Sections 2 and 4).

2.1 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. We assume that f is composed of $k \in \mathbb{N}$ linear pieces (components). The function f can be expressed as

$$f(z) = \max_{i=1,\dots,k} \left(A[z,1]\right)_i , \tag{2}$$

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

$$f^\mu(z) = \max_{w\in Q_k} \left\{ \langle A[z,1], w\rangle - \mu\rho(w) \right\}. \tag{3}$$

According to (Nesterov, 2005, Theorem 1), the function f^μ defined in eq. (3) 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 $f^0(z) = \max_{w\in Q_k}\{\langle A[z,1], w\rangle\} = f(z)$. 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

$$f^0(z) - \mu \sup_{w\in Q_k} \rho(w) \leq f^\mu(z) \leq f^0(z) \tag{4}$$

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

$$\sup_{z\in\mathbb{R}^q} |f(z) - f^\mu(z)| \leq \mu \sup_{w\in Q_k} \rho(w) = O(\mu). \tag{5}$$

Indeed, eq. (4) holds true since for all $z \in \mathbb{R}^q$,

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

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

2.2 Two choices for the proximity function

We consider two choices of the prox function ρ.

2.2.1 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. (5), we obtain the uniform bound

$$\sup_{z \in \mathbb{R}^q} |f(z) - f_e^\mu(z)| \leq \mu \log(k) \tag{6}$$

for the entropy smoothed function f_e^μ obtained by using ρ_e in eq. (3). Interestingly, smoothing with the entropy prox function admits a closed-form expression of f_e^μ given by

$$\begin{aligned} f_e^\mu(z) &= \max_{w \in Q_k} \left\{ \sum_{i=1}^{k} w_i A_i^\top [z,1] - \mu \left(\sum_{i=1}^{k} w_i \log(w_i) + \log(k) \right) \right\} \\ &= \mu \log \left(\frac{1}{k} \sum_{i=1}^{k} e^{\frac{A_i^\top [z,1]}{\mu}} \right), \end{aligned} \tag{7}$$

see (Nesterov, 2005, Lemma 4).

2.2.2 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. (3) with squared error prox function is equivalent to the convex program

$$f_s^\mu(z) = \min_{w \in Q_k} \left(\frac{1}{k} \sum_{i=1}^k w_i^2 - \sum_{i=1}^k w_i c_i(z) \right), \tag{8}$$

where $c_i(z) = A_i^\top [z, 1]/\mu - 1/k$ depends on A_i and μ and is defined for any $i \in \{1, \ldots, k\}$. The problem in eq. (8) 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). 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

$$f_s^\mu(z) = \sum_{i=1}^k (\hat{c}(z))_i \cdot \left(A_i^\top [z, 1] \right) - \mu \rho_s(\hat{c}(z)). \tag{9}$$

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

$$\sup_{z \in \mathbb{R}^q} |f(z) - f_s^\mu(z)| \leq \mu \left(1 - \frac{1}{k} \right) \tag{10}$$

for the squared error smoothing approach.

2.3 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

$$L(\beta) = \frac{1}{2} \|X\beta - y\|_2^2 + \lambda \|\beta\|_1 \tag{11}$$

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. In the remainder of this section, we assume that $z \in \mathbb{R}$ is a scalar.

Though the convexity of the unsmoothed LASSO objective function conceptually makes it easy to find the minimizer (even without closed-form gradients), the Lipschitz continuous gradients in our approach allow us to find the minimizer both more reliably and more computationally efficient (see Section 4).

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 $f(z) = \max\{-z, z\} = \max_{i=1,2}(A[z, 1])_i$.

2.3.1 Entropy prox function

For the entropy prox function, eq. (7) with A as in Section 2.3 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. (6) satisfies the approximation bound

$$\sup_{z\in\mathbb{R}} |f(z) - f_e^\mu(z)| \leq \mu \log(2). \tag{12}$$

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. (11) with the entropy prox function results in

$$\begin{aligned}
L_e^\mu(\beta) &= \frac{1}{2}\|X\beta - y\|_2^2 + \lambda \sum_{i=1}^p f_e^\mu(\beta_i), \\
\frac{\partial L_e^\mu}{\partial \beta_i}(\beta) &= -\frac{2}{n}\langle y - X\beta, X_i\rangle + \lambda g_e^\mu(\beta_i), \\
\frac{\partial^2 L_e^\mu}{\partial \beta_i \partial \beta_j}(\beta) &= \frac{2}{n}(X^\top X)_{ij} + \mathbb{I}(i=j)\cdot \lambda h_e^\mu(\beta_i),
\end{aligned} \tag{13}$$

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 λh_e^μ 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 λh_e^μ 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. (13) will be singular except for a few artificial cases and the second order Newton-Raphson method will not applicable. In the simulations we therefore focus on quasi-Newton methods which involve only L_e^μ and its gradient $\partial L_e^\mu / \partial \beta_i$.

2.3.2 Squared error prox function

Similarly, eq. (9) with A as in Section 2.3 simplifies to

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

where $\hat{c}(z) \in \mathbb{R}^2$ is the Michelot (1986) projection vector of $c(z)$ (see Section 2.2.2) onto Q_2. According to eq. (10), the squared error prox function satisfies the approximation bound

$$\sup_{z \in \mathbb{R}} |f(z) - f_s^\mu(z)| \leq \frac{1}{2}\mu. \tag{14}$$

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., 2016, 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. (11) with the squared error prox function results in

$$\begin{aligned} L_s^\mu(\beta) &= \frac{1}{2}\|X\beta - y\|_2^2 + \lambda \sum_{i=1}^p f_s^\mu(\beta_i), \\ \frac{\partial L_s^\mu}{\partial \beta_i}(\beta) &= -\frac{2}{n}\langle y - X\beta, X_i \rangle + \lambda g_s^\mu(\beta_i), \\ \frac{\partial^2 L_s^\mu}{\partial \beta_i \partial \beta_j}(\beta) &= \frac{2}{n}(X^\top X)_{ij} + \mathbb{I}(i=j) \cdot \lambda \left. \frac{\partial^2 f_s^\mu}{\partial z^2} \right|_{z=\beta_i} \end{aligned} \tag{15}$$

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 2.3.1 we observe that the Hessian matrix is singular due to $X^\top X$ being singular, and that the additional diagonal entries stemming from $\partial^2 f_s^\mu / \partial z^2$ 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.

3 Progressive smoothing and theoretical guarantees of correctness

This section proposes an adaptive smoothing technique for the LASSO (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 which quantify the similarity of both the unsmoothed and smoothed LASSO objective functions, as well as the estimators obtained from either the unsmoothed on smoothed LASSO, thus giving additional validity to our approach.

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

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

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

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

end

return $\hat{\beta}_0$;

3.1 Progressive smoothing

Instead of solving the smoothed LASSO problem $\hat{\beta} = \arg\min_\beta L^\mu(\beta)$ directly for some $\mu > 0$, were 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 the estimate of β, we decrease the smoothing parameter and use 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 according to Proposition 1, 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 last 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.

3.2 Theoretical guarantees

The bounds on both f_e^μ in eq. (12) and f_s^μ in eq. (14) carry over to a bound on the overall approximation of the LASSO objection function of eq. (11):

Proposition 1. *The smoothed entropy and squared error prox functions lead to the following bounds on the LASSO objective function estimates*

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

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

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 $\mu \to 0$. The smoothed approximation and its gradient given in eq. (13) and eq. (15), respectively, remain valid for any $\mu > 0$.

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 for the *FISTA* algorithm 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 bound on the distance between the two minimizers. This is done in the next result giving an explicit bound.

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 two minimizers, $i \in \{1, 2\}$. Then for any $\delta > 0$ and any $y_1 \in \mathbb{R}^s$ with $y_1 \neq x_1$, $\|y_1 - x_1\|_2 \leq \delta$ 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 LASSO approaches 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). We compare the following four approaches, of which the first two approaches 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 $\partial L/\partial \beta_i = -2/n \cdot \langle y - X\beta, X_i \rangle + \lambda \text{sign}(\beta_i)$. This approach will be referred to as *unsmoothed LASSO*;

2. We use the *glmnet* function provided in the *R* package *glmnet*, available on *The Comprehensive R Archive Network* (R Core Team, 2014);

3. We minimize the smoothed LASSO objective function of eq. (13) (or eq. (15)) using their explicit gradients;

4. We employ the progressive smoothing approach of Section 3.1 with target smoothing parameter $\mu_0 = 2^{-5}$ and $N = 5$ smoothing steps;

In the simulation studies, we apply glmnet with standardization and parameter *alpha=1*. In order to obtain a parameter estimate from glmnet, we evaluate the regularization path with the function *coef*, where we set its parameter s to the choice of λ used in the other approaches.

All results are averages over 100 repetitions. The choice of the LASSO regularization parameter λ varies in each experiment and is given individually.

Since the *R* package *SIS* accompanying Fan and Li (2001) does not contain methodology for smoothing the LASSO, we omit its comparison with our approach in this section.

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, 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 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 $\epsilon = N_p(0, \sigma^2)$ is a noise vector with entries generated from a Normal distribution with some mean zero and 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%, 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 glmnet and the progressive smoothing approach yield

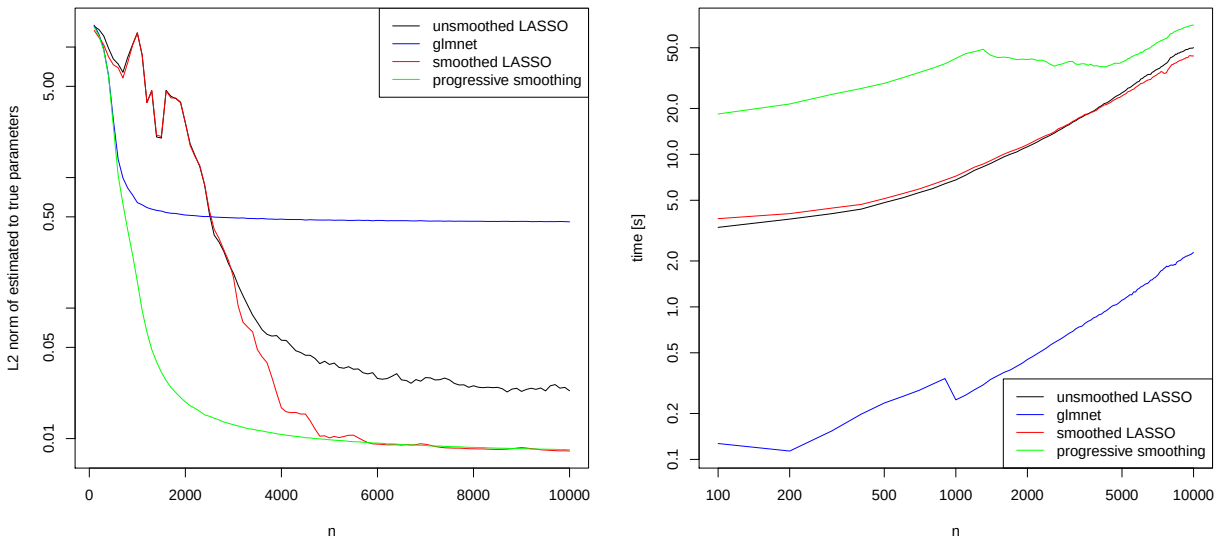

Figure 1: L_2 norm of parameter estimate to truth (left) and runtime in seconds (right) as a function of $n \in [1, 10000]$ while $p = 1000$. Logarithmic scale on the y-axes.

stable estimates for all n, although the estimates of the progressive smoothing approach are of higher quality than the one of glmnet. As expected, all methods become more accurate as the number of data points n increases. Surprisingly, the smoothed approach draws equal in accuracy with the progressive smoothing approach 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. This seems to be a reasonable trade-off for the considerably improved accuracy that the progressive smoothing provides. We find that glmnet is highly optimized and thus faster than the other approaches.

Similarly to the previous experiment, in Figure 2 (left) we keep $n = 1000$ fixed and investigate the dependence of all four approaches on $p \in [1, 5000]$. Results are qualitatively similar to the ones of Figure 1: The unsmoothed and smoothed LASSO approaches seem to suffer from numerical instabilities, while the progressive smoothing approach and glmnet find good quality solutions much more reliably. As observed before, progressive smoothing seems to outperform glmnet, even though both approaches draw equal for large p. As expected, while keeping the data size n fixed, estimation becomes more challenging 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. Glmnet is again faster than the other approaches. However, since runtimes in the neighborhood of hundreds of seconds for the

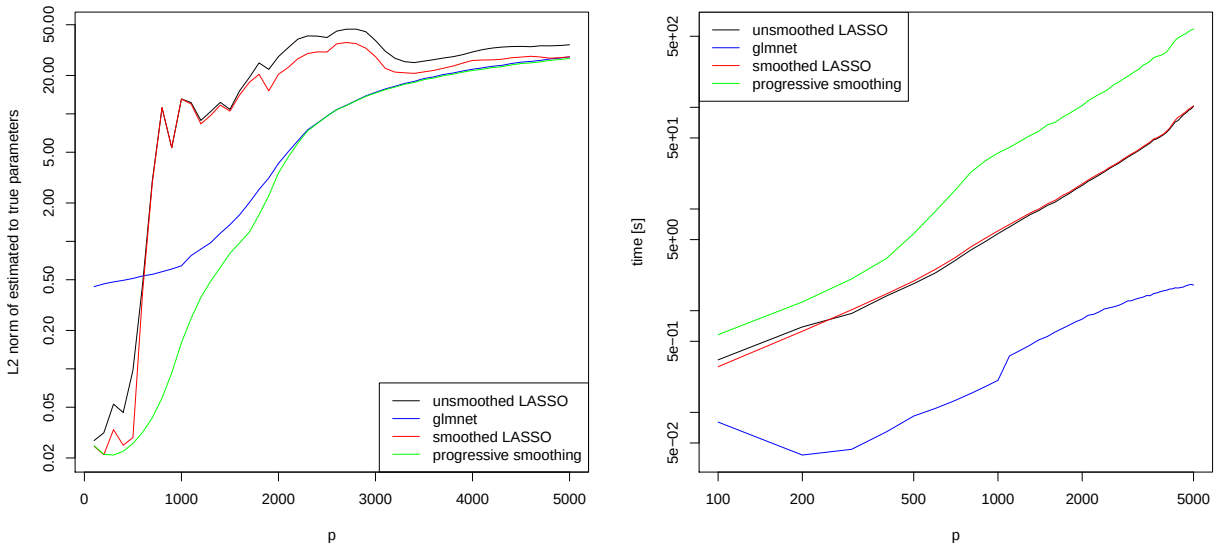

Figure 2: L_2 norm of parameter estimate to truth (left) and runtime in seconds (right) as a function of $p \in [1, 5000]$ while $n = 1000$. Logarithmic scale on the y-axes.

progressive smoothing approach seem acceptable for practical applications, we conclude that the superior accuracy of the progressive smoothing comes at an acceptable increase in runtime.

Since the unsmoothed LASSO is slower to compute, does not come with the guarantees we established for our smoothing approach (see Section 3.2), and gives worse results than the smoothing methods in the previous experiments we will focus in the remainder of the simulations on glmnet, smoothed LASSO, and progressive smoothing.

4.2 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). 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 is 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.

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

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

method	L_2 norm	runtime [s]
glmnet	44.5	19.9
smoothed LASSO	14.9	120.2
progressive smoothing	22.6	526.2

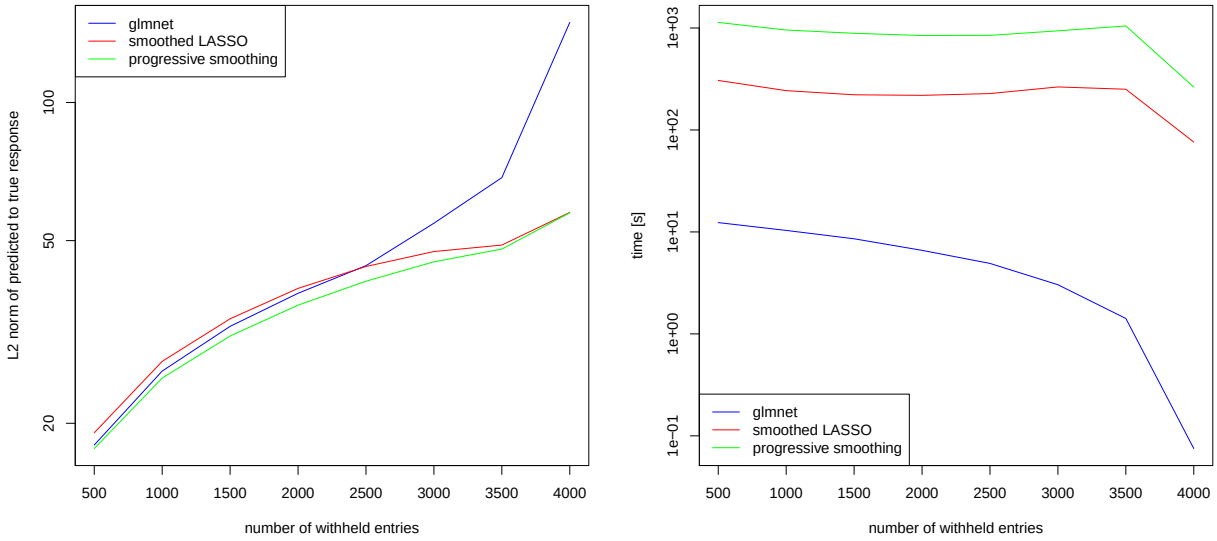

Figure 3: L_2 norm of predicted to withheld data in cross-validation (left) and runtime in seconds (right) as a function of the number of withheld entries. Dataset of polygenic risk scores of X. Logarithmic scale on the y-axes.

4.2.1 Results from a single run

We fit the response y to the data X using our smoothed LASSO approach of eq. (13) (or eq. (15)), as well as the progressive smoothing approach of Section 3.1, and compare both to the glmnet function in R.

Table 1 shows results for a single application of the three algorithms. After computing the LASSO 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 analogously to Section 4.1, glmnet performs worse than smoothed LASSO and progressive smoothing. However, in this experiment it turns out that a single application of 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 glmnet methods 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 $X_I\hat{\beta}$. We evaluate the quality of the prediction by computing the L_2 norm $\|X_I\hat{\beta} - y_I\|_2$ between predicted and withheld data.

Figure 3 (left) shows results of this cross-validation experiment. We observe that for up to a proportion of roughly two thirds of withheld entries, all approaches yield almost equal deviations between predicted and withheld data. As the number of withheld entries increases further, the accuracy of both smoothed LASSO and progressive smoothing somewhat stabilizes, whereas glmnet is considerably worse for large numbers of withheld entries. This shows that the glmnet estimates become less reliable especially when based on little data. 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 data.

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 glmnet 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 β 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 a slight perturbation of $\epsilon = N_p(0, 0.1)$ is added, that is the entries of ϵ are drawn independently from a Normal distribution with mean 0 and standard deviation 0.1.

After generating a vector β in this fashion having $nz \in \{0, \ldots, p\}$ non-zero entries, and after having calculating the perturbed y, we again use glmnet, 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

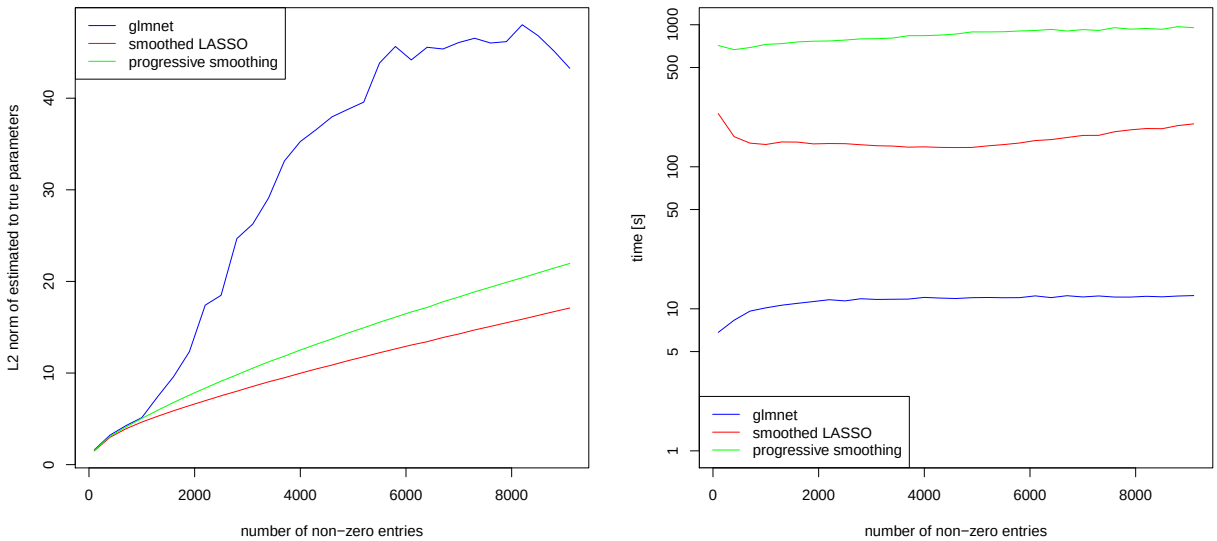

Figure 4: L_2 norm of parameter estimate to truth (left) and runtime in seconds (right) as a function of the number of non-zero entries in the simulated true parameter vector. Dataset of polygenic risk scores of X. Logarithmic scale on the y-axis of the time plot.

stable estimates than glmnet (as expressed through a lower deviation in L_2 norm). Only for very sparse vectors (i.e., a number of non-zero entries in β ranging from zero to roughly 1000), we observe that glmnet draws equal with the smoothed LASSO approaches.

The improved accuracy of our smoothed LASSO approach again comes at an increased runtime. Figure 4 (right) shows runtime results in seconds for all three approaches. We observe that our smoothed LASSO approach is roughly a factor ten slower than glmnet, and that as usual, progressive smoothing is a factor five slower than smoothed LASSO. Nevertheless, depending on the application, a roughly constant increase in runtime could be worth the considerably increase in accuracy.

Overall, we conclude from the simulations that the progressive smoothing approach of Section 3.1 yields stable LASSO estimates which often outperform glmnet and can be computed at a constant multiple of the runtime of the smoothed LASSO. We suggest employing the progressive smoothing approach with the target smoothing parameter $\mu_0 = 2^{-N}$ and a low number of e.g. $N = 5$ or $N = 10$ steps. 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 objective function of the LASSO.

Most importantly, the presented approach comes with two guarantees. First, we show that the

utilized Nesterov smoothing enables us to obtain explicit gradients for the LASSO while guaranteeing a uniform bound on the distance between the unsmoothed and smoothed LASSO functions. Using the Nesterov smoothing parameter, this distance can be made arbitrarily small. Second, we show that the unifrom closeness of unsmoothed and smoothed objective functions translates to an explicit bound on the distance 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 close to the actual estimates obtained if we minimized the original LASSO objective.

Simulations showed that in our smoothed LASSO approach yields more reliable estimates than the glmnet algorithm in an real data application of polygenic risk scores. The progressive smoothing procedure we propose in Section 3.1 yields stable LASSO estimators at only a constant increase in runtime. We propose to use the progressive smoothing approach as a default method, in connection with either the entropy or squared error prox function, applied with target smoothing $\mu_0 = 2^{-N}$ and small number of e.g. $N = 5$ or $N = 10$ smoothing steps depending on the application.

A Proofs

Proof of Proposition 1. The bounds on L_e^μ and L_s^μ follow from eq. (12) and eq. (14) after a direct calculation.

Since both f_e^μ and f_s^μ are convex according to (Nesterov, 2005, Theorem 1), see also Section 2.1, 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 2.3.1) 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. (13) in fact strictly convex. Similar arguments show that eq. (15) 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 through 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

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

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.

Hahn, G., Banerjee, M., and Sen, B. (2016). Parameter Estimation and Inference in a Continuous Piecewise Linear Regression Model. *Manuscript draft*, pages 1–38.

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.

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.

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

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.