CONVERGENCE RATE ANALYSIS OF SMOOTHED LASSO

Subhadip Mukherjee and Chandra Sekhar Seelamantula

Department of Electrical Engineering, Indian Institute of Science, Bangalore 560012, India
Emails: subhadip@ee.iisc.ernet.in, chandra.sekhar@ieee.org

ABSTRACT

The LASSO regression has been studied extensively in the statistics and signal processing community, especially in the realm of sparse parameter estimation from linear measurements. We analyze the convergence rate of a first-order method applied on a smooth, strictly convex, and parametric upper bound on the LASSO objective function. The upper bound approaches the true non-smooth objective as the parameter tends to infinity. We show that a gradient-based algorithm, applied to minimize the smooth upper bound, yields a convergence rate of $\mathcal{O}\left(\frac{1}{K}\right)$, where K denotes the number of iterations performed. The analysis also reveals the optimum value of the parameter that achieves a desired prediction accuracy, provided that the total number of iterations is decided a priori. The convergence rate of the proposed algorithm and the amount of computation required in each iteration are same as that of the *iterative soft thresholding* technique. However, the proposed algorithm does not involve any thresholding operation. The performance of the proposed technique, referred to as *smoothed LASSO*, is validated on synthesized signals. We also deploy smoothed LASSO for estimating an image from its blurred and noisy measurement, and compare the performance with the *fast iterative shrinkage thresholding algorithm* for a fixed run-time budget, in terms of the reconstruction peak signal-to-noise ratio and structural similarity index.

***Index Terms*—** Sparse estimation, LASSO regression, Smoothing, Accelerated gradient descent.

1. INTRODUCTION

In recent years, considerable amount of research attention has been devoted to the problem of sparse recovery from noisy random linear projections, popularly known as *compressive sensing* (CS) [1–3], which finds applications in a variety of fields including, but not limited to, medical imaging [4], computational biology [5], radar imaging [6], etc. The principal objective of CS is to recover a sparse signal $\mathbf{x} \in \mathbb{R}^n$ from a lower-dimensional noisy measurement vector $\mathbf{y} \in \mathbb{R}^m$, $m < n$, given by $\mathbf{y} = \Phi\mathbf{x} + \mathbf{w}$, by solving

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ subject to } \|\mathbf{y} - \Phi\mathbf{x}\|_2 \leq \epsilon, \tag{1}$$

where the ℓ_0-norm of $\mathbf{x}$, denoted by $\|\mathbf{x}\|_0$, counts the number of non-zero coordinates in $\mathbf{x}$. The optimization in (1) is NP-hard and computationally intractable, in general. The algorithms developed for CS can be broadly classified into two categories: greedy algorithms and convex relaxation-based approaches. In the greedy algorithms, one first identifies the locations of the non-zero entries, and subsequently the magnitudes are obtained by solving a least squares minimization problem. Algorithms, such as *orthogonal matching pursuit* (OMP) [7], CoSaMP [8], etc. fall under this category. In the convex relaxation techniques, the ℓ_0-norm minimization in (1) is replaced by ℓ_1-norm minimization [9] given by

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ subject to } \|\mathbf{y} - \Phi\mathbf{x}\|_2 \leq \epsilon. \tag{2}$$

The problem (2) is a convex program and one can design polynomial-time algorithms to solve it. It is shown in [10] that it is possible to recover any sparse signal $\mathbf{x}$, with $\|\mathbf{x}\|_0 \leq s$, by solving (2), provided $\|\mathbf{w}\|_2 \leq \epsilon$ and the sensing matrix Φ satisfies $\delta_{3s} + 3\delta_{4s} < 2$, where δ_s denotes the *restricted isometry constant* of order s. The formulation in (2) is also famously known as the *basis pursuit denoising* (BPDN) problem.

An alternative to solving (2) for sparse recovery is to minimize an ℓ_1-constrained quadratic penalty of the form

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \Phi\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1, \tag{3}$$

which is known as LASSO regression [11, 12] in the statistics literature. The parameter λ trades-off the sparsity in the estimate with the data fidelity criterion. Necessary and sufficient conditions for reliable support recovery using LASSO are developed in [13], in a high-dimensional and noisy measurement setting. Knight et al. [14] has analyzed the asymptotic behavior of the optimal solution to (3) in a more general setup, where one replaces the ℓ_1 penalty term in (3) with an ℓ_q penalty, for $0 < q \leq 2$. In this paper, we consider the smooth approximation to the LASSO cost function proposed in [15], by using the integral of the sigmoid function. We propose to apply an accelerated first-order technique on the resulting approximation and derive the convergence rate of the overall algorithm. Performance validation is carried out on synthesized signals and real images.

1.1. Our contribution

We propose to use a smooth approximation to the ℓ_1-norm [15] and solve the resulting minimization problem by applying an accelerated gradient descent technique [16, 17] on the smooth objective. A smooth function $\phi_\alpha(u)$, defined as

$$\phi_\alpha(u) = \frac{2}{\alpha}\log\left(1 + e^{\alpha u}\right) - u, \ \alpha > 0,$$

is used to approximate $|u|$ and, in turn, yields an approximation of $\|\mathbf{x}\|_1$ for any vector $\mathbf{x}$, where one uses $\phi_\alpha(\mathbf{x}_i)$ to smoothly approximate $|\mathbf{x}_i|$, for every i. We refer to the resulting technique as *smoothed LASSO*, and prove that it generates estimates of the

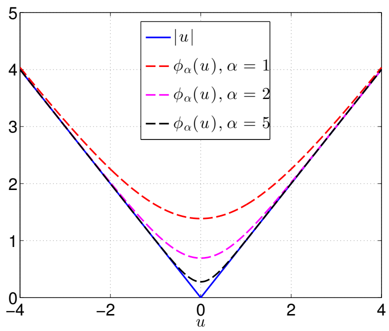

Fig. 1. (Color online) The modulus function and its smooth approximation $\phi_\alpha(u)$, for different α.

underlying sparse signal iteratively, leading to a convergence rate of $\mathcal{O}\left(\frac{1}{K}\right)$ of the actual non-smooth objective, where K denotes the number of iterations. Our analysis also indicates the value of the smoothing parameter α that one needs to choose, for a fixed K, to achieve the tightest upper-bound on the rate of convergence. Performance validation and a comparative analysis of the proposed smoothed LASSO approach with state-of-the-art techniques are also provided, on synthesized signals as well as for an image deblurring exercise. Note that while deriving the convergence rate, we confine ourselves within the realm of first-order techniques, that use no more than $\mathcal{O}\left(n^2\right)$ computations in every iteration. However, the convergence rate established in the paper might be improved if one uses a higher-order technique, such as the *Newton's method*, to minimize the smooth approximation, albeit at the cost of increased computational complexity per iteration.

2. SMOOTHED LASSO AND ITS CONVERGENCE ANALYSIS

We develop the smoothed LASSO algorithm and establish an upper bound on its rate of convergence, in a step-by-step manner.

2.1. Smooth approximation function and its properties

We begin by establishing the properties of the function $\phi_\alpha(u) = \frac{2}{\alpha}\log\left(1+e^{\alpha u}\right) - u$, that smoothly approximates $|u|$ from above. To facilitate visual comparison, we plot $|u|$ and $\phi_\alpha(u)$ in Fig. 1.

Proposition 1 *The function* $\phi_\alpha(u) = \frac{2}{\alpha}\log\left(1+e^{\alpha u}\right) - u$*, where* $\alpha > 0$*, satisfies the following properties:*

1. $0 < \phi_\alpha(u) - |u| \le \frac{2\log 2}{\alpha}$*, for every* $u \in \mathbb{R}$*. Moreover, the inequality is satisfied with an equality if and only if* $u = 0$*.*
2. $\phi_\alpha(u)$ *is twice continuously differentiable and strictly convex in* u*, satisfying* $0 < \frac{\mathrm{d}^2\phi_\alpha(u)}{\mathrm{d}u^2} \le \frac{\alpha}{2}$*.*

Consequently, it is straightforward to see that $\lim_{\alpha\to\infty}\phi_\alpha(u) = |u|$.

Proof: (i) Since $\frac{2}{\alpha}\log\left(1+e^{\alpha u}\right) > 0$ for any u, we have $\phi_\alpha(u) > -u$. Moreover, $\frac{2}{\alpha}\log\left(1+e^{\alpha u}\right) > \frac{2}{\alpha}\alpha u = 2u$, thus implying that $\phi_\alpha(u) > u$ for all u. Combining $\phi_\alpha(u) > u$ and $\phi_\alpha(u) > -u$, we get $\phi_\alpha(u) > |u|$. Define $g_\alpha(u) = \phi_\alpha(u) - |u|$, and consider the cases $u \ge 0$ and $u \le 0$ separately. For $u \ge 0$, we have

$$g_\alpha(u) = \frac{2}{\alpha}\log\left(1+e^{\alpha u}\right) - 2u, \text{ and } \frac{\mathrm{d}g_\alpha(u)}{\mathrm{d}u} = \frac{-2}{1+e^{\alpha u}} < 0.$$

Algorithm 1 Smoothed LASSO: Accelerated gradient descent to minimize the smooth upper-bound $f_\alpha(\mathbf{x})$ on $f(\mathbf{x})$.

1. Initialization: Set iteration counter k to 1, tolerance $\epsilon > 0$, step-size $\mu = \frac{1}{L}$, where L satisfies $\nabla^2 f_\alpha(\mathbf{x}) \preceq L I_n$, and initialize $\mathbf{x}_{k-1}$.

2. Iterate the following steps K **times:**
$\mathbf{w}_k = \mathbf{x}_{k-1} + \frac{k-2}{k+1}\left(\mathbf{x}_{k-1} - \mathbf{x}_{k-2}\right)$, $\mathbf{x}_k = \mathbf{w}_k - \mu\nabla f_\alpha(\mathbf{w}_k)$, and $k \leftarrow k+1$

3. Output: The minimizer $\mathbf{x}_\alpha^* = \mathbf{x}_K$.

Likewise, for $u \le 0$ we have

$$g_\alpha(u) = \frac{2}{\alpha}\log\left(1+e^{\alpha u}\right), \text{ and } \frac{\mathrm{d}g_\alpha(u)}{\mathrm{d}u} = \frac{2e^{\alpha u}}{1+e^{\alpha u}} > 0.$$

Thus, we conclude that the function $g_\alpha(u)$ is monotonically increasing in the range $-\infty < u \le 0$ and decreasing in the range $0 \le u < \infty$, however, $\frac{\mathrm{d}g_\alpha(u)}{\mathrm{d}u}$ is not defined at $u = 0$. Combining these, we have $g_\alpha(u) \le g_\alpha(0) = \frac{2\log 2}{\alpha}$, for all u.
(ii) $\phi_\alpha(u)$ is strictly convex in u, since $\frac{\mathrm{d}^2\phi_\alpha(u)}{\mathrm{d}u^2} = \frac{2\alpha e^{\alpha u}}{(1+e^{\alpha u})^2} > 0$. Also, $\frac{\mathrm{d}^2\phi_\alpha(u)}{\mathrm{d}u^2} = \frac{2\alpha e^{\alpha u}}{(1+e^{\alpha u})^2} \le \frac{\alpha}{2}$, by virtue of the fact that $\frac{t}{(1+t)^2} \le \frac{1}{4}$, for all t. This completes the proof.

2.2. Convergence of the smooth approximation

Let us define

$$\begin{aligned}\mathbf{x}^* &= \arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{y} - \Phi\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1, \text{ and} \\ \mathbf{x}_\alpha^* &= \arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{y} - \Phi\mathbf{x}\|_2^2 + \lambda\sum_{i=1}^n \phi_\alpha(x_i)\end{aligned}$$

to be the minima of the LASSO objective and its smooth approximation, respectively. For convenience, we write

$$\begin{aligned}f(\mathbf{x}) &= \frac{1}{2}\|\mathbf{y} - \Phi\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1, \text{ and} \\ f_\alpha(\mathbf{x}) &= \frac{1}{2}\|\mathbf{y} - \Phi\mathbf{x}\|_2^2 + \lambda\sum_{i=1}^n \phi_\alpha(x_i).\end{aligned}$$

The properties established in Proposition 1 lead us to the following corollary, where we prove that as the parameter α tends to infinity, $f(\mathbf{x}_\alpha^*)$ gets infinitesimally closer to $f(\mathbf{x}^*)$.

Corollary 1 $\lim_{\alpha\to\infty} f(\mathbf{x}_\alpha^*) = f(\mathbf{x}^*)$.

Proof: We have $f(\mathbf{x}^*) \overset{(a)}{\le} f(\mathbf{x}_\alpha^*) \overset{(b)}{<} f_\alpha(\mathbf{x}_\alpha^*) \overset{(c)}{<} f_\alpha(\mathbf{x}^*)$, where (a) is a consequence of the fact that $\mathbf{x}^*$ minimizes $f(\mathbf{x})$, (b) follows from Proposition 1, and (c) holds as $\mathbf{x}_\alpha^*$ is the unique minimizer of $f_\alpha(\mathbf{x})$. Now, again by Proposition 1, $\lim_{\alpha\to\infty} f_\alpha(\mathbf{x}^*) = f(\mathbf{x}^*)$, and consequently, we have the result, since $f(\mathbf{x}_\alpha^*)$ is sandwiched between $f_\alpha(\mathbf{x}^*)$ and $f(\mathbf{x}^*)$. This result guarantees that the objective function evaluated at $\mathbf{x}_\alpha^*$, the minimizer of the smooth approximation $f_\alpha(\mathbf{x})$, approaches the true minimum for sufficiently large α.

2.3. Proposed algorithm

In order to minimize $f(\mathbf{x})$, we propose to employ the accelerated gradient descent algorithm [16] on the smooth approximation $f_\alpha(\mathbf{x})$. The steps are summarized in Algorithm 1. The accelerated version of the gradient descent algorithm entails same amount of computation as its regular version, but results in a better convergence rate, as shown in [16]. For completeness, we state the result on the convergence performance of the algorithm in Proposition 2. Note that for two matrices A and B, the notation $A \preceq B$ is used to indicate that the difference $(B-A)$ is positive semi-definite.

Proposition 2 *Let $h(\mathbf{x})$ be a smooth, twice-differentiable convex function satisfying $\nabla^2 h(\mathbf{x}) \preceq L I_n$, for some constant $L>0$, where I_n denotes the $n \times n$ identity matrix. Then, for any initialization $\mathbf{x}_0$ and a step-size $\mu = \frac{1}{L}$, the iterates $\mathbf{x}_k$ generated by the accelerated gradient method satisfies*

$$h(\mathbf{x}_k) - h(\mathbf{x}^*) \leq \frac{2L\|\mathbf{x}_0 - \mathbf{x}^*\|_2^2}{(k+1)^2}.$$

Thus, one can conclude from Proposition 2 that there exists a constant $c_1 > 0$ such that the iterates generated by the algorithm satisfies $h(\mathbf{x}_k) - h(\mathbf{x}^*) \leq \frac{c_1}{k^2}$, for any initialization $\mathbf{x}_0$. Moreover, the value of the constant c_1 is given by $c_1 = 2L\|\mathbf{x}_0 - \mathbf{x}^*\|_2^2$. This result plays a crucial role in establishing the main result of the paper.

2.4. Convergence rate of the smoothed LASSO algorithm

Before proving the main theorem, let us investigate whether the approximation function $f_\alpha(\mathbf{x})$ satisfies the smoothness condition required for Proposition 2 to hold. We observe that $f_\alpha(\mathbf{x})$ is twice continuously differentiable and the Hessian of $f_\alpha(\mathbf{x})$ is given by

$$\nabla^2 f_\alpha(\mathbf{x}) = \Phi^T\Phi + D,$$

where D is a diagonal matrix whose i^{th} entry is $d_i = \lambda \frac{\mathrm{d}^2\phi_\alpha(\mathbf{x}_i)}{\mathrm{d}\mathbf{x}_i^2}$. As a consequence of Proposition 1, we have $d_i \leq \frac{\lambda\alpha}{2}$ for every i. Therefore, we have

$$\nabla^2 f_\alpha(\mathbf{x}) \preceq \left(\sigma_{\max}^2(\Phi) + \frac{\lambda\alpha}{2}\right) I_n,$$

where $\sigma_{\max}(\Phi)$ denotes the largest singular value of the matrix Φ. Thus, $f_\alpha(\mathbf{x})$ satisfies the smoothness condition as required in Proposition 2, with a constant $L = \sigma_{\max}^2(\Phi) + \frac{\lambda\alpha}{2}$. Now, equipped with the results proved in the preceding sections, we are in a position to state and prove the main theorem on the convergence rate of the smoothed LASSO algorithm.

Theorem 1 *The output $\mathbf{x}_K$ of Algorithm 1, obtained after K iterations, satisfies $f(\mathbf{x}_K) - f(\mathbf{x}^*) \leq \mathcal{O}\left(\frac{1}{K}\right)$, yielding a linear rate of decrease of the actual LASSO objective.*

Proof: Observe that for any iteration index k, we have

$$\begin{aligned} f(\mathbf{x}_k) - f(\mathbf{x}^*) &= (f(\mathbf{x}_k) - f_\alpha(\mathbf{x}_k)) + (f_\alpha(\mathbf{x}_k) - f_\alpha(\mathbf{x}_\alpha^*)) \\ &+ (f_\alpha(\mathbf{x}_\alpha^*) - f(\mathbf{x}^*)), \end{aligned}$$

where $f(\mathbf{x}_k) - f_\alpha(\mathbf{x}_k) \leq 0$ by Proposition 1, $f_\alpha(\mathbf{x}_k) - f_\alpha(\mathbf{x}_\alpha^*) \leq \frac{c_1}{k^2}$ for some $c_1 > 0$, by Proposition 2, and

$$f_\alpha(\mathbf{x}_\alpha^*) - f(\mathbf{x}^*) < f_\alpha(\mathbf{x}^*) - f(\mathbf{x}^*) \leq \frac{2n\log 2}{\alpha},$$

by using Proposition 1 again. By combining the bounds on the three terms together, we obtain

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \quad \leq \quad \frac{c_1}{k^2} + \frac{2n\log 2}{\alpha},$$

where $c_1 = 2LR$, with $L = \sigma_{\max}^2(\Phi) + \frac{\lambda\alpha}{2}$ and $R = \|\mathbf{x}_0 - \mathbf{x}^*\|_2^2$. Assume that Algorithm 1 is terminated after a fixed number of iterations K. Consequently, we have that

$$f(\mathbf{x}_K) - f(\mathbf{x}^*) \leq \frac{2R\sigma_{\max}^2(\Phi)}{K^2} + \underbrace{\frac{\lambda R\alpha}{K^2} + \frac{2n\log 2}{\alpha}}_{q(\alpha)}.$$

We observe that the only parameter that can be controlled in the expression of the upper bound is α, in order to achieve a desirable rate of convergence. Thus, if the total number of iterations K is chosen a priori, one can tighten the upper bound on the difference $f(\mathbf{x}_K) - f(\mathbf{x}^*)$ by setting $\alpha = \left(\frac{2\log 2n}{\lambda R}\right)^{\frac{1}{2}} K$, which results in

$$f(\mathbf{x}_K) - f(\mathbf{x}^*) \leq \frac{2R\sigma_{\max}^2(\Phi)}{K^2} + 2\frac{\sqrt{2\lambda Rn\log 2}}{K},$$

indicating that $f(\mathbf{x}_K) - f(\mathbf{x}^*) \leq \mathcal{O}\left(\frac{1}{K}\right)$. The optimum value of α that yields the tightest upper-bound is obtained by differentiating $q(\alpha) = \frac{\lambda R\alpha}{K^2} + \frac{2n\log 2}{\alpha}$ with respect to α, and setting $q'(\alpha)$ to zero. It can be verified that $\frac{\mathrm{d}^2 q(\alpha)}{\mathrm{d}\alpha^2} > 0$, indicating that the solution to $\frac{\mathrm{d}q(\alpha)}{\mathrm{d}\alpha} = 0$ corresponds to a minimum. This completes the proof of the theorem. Depending on the accuracy one desires to achieve, one can choose the number of iterations K appropriately at the outset.

SNR (dB)	Sparsity	ISTA	FISTA	IRLS	CVX	Smoothed LASSO
30	$s=8$	0.055	0.044	0.037	0.044	0.043
	$s=16$	0.114	0.084	0.076	0.085	0.081
40	$s=8$	0.060	0.014	0.012	0.014	0.017
	$s=16$	0.123	0.025	0.025	0.027	0.027
$+\infty$	$s=8$	0.058	0.004	0.008	0.004	0.012
	$s=16$	0.121	0.006	0.012	0.006	0.014

Table 1. Reconstruction MSE values of the competing algorithms, for different values of the measurement SNR. The corresponding sparsity values are indicated in the table. The parameter α is taken as $\alpha = 100$. The MSE values are averaged over 50 independent noise realizations. The number of iterations for the FISTA, IRLS, and the smoothed LASSO algorithms is chosen to be $I = 50$. The value of λ is taken as $\lambda = 0.06$ and $\lambda = 0.02$, corresponding to input SNR values 30 dB and 40 dB, respectively. Also, for noiseless measurements (SNR $= +\infty$), we choose $\lambda = 0.02$.

3. EXPERIMENTAL RESULTS

We validate the proposed smoothed LASSO algorithm on synthesized signals and compare the performance with other competing algorithms. On the application front, we consider the problem of image deblurring, where the objective is to retrieve an image from its blurred and noisy measurement. The experiments are performed on the MATLAB 2011 platform, running on a system with 8 GB internal memory and 3.2 GHz core-i5 processor.

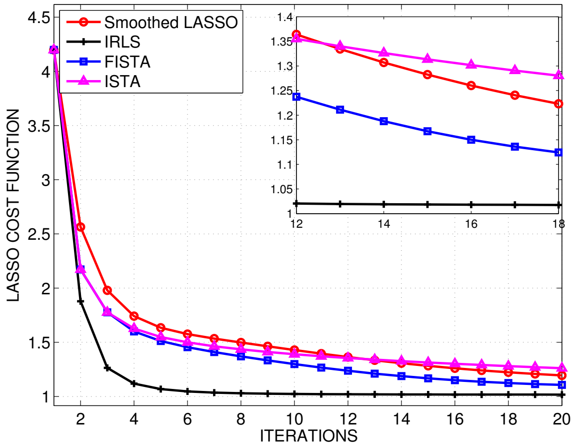

Fig. 2. (Color online) The variation of the LASSO objective function with respect to the number of iterations for different algorithms.

3.1. Experiments on synthesized signals

We consider the problem of recovering a sparse signal $\mathbf{x} \in \mathbb{R}^n$, where $n = 256$, from the $m = 128$ dimensional noisy measurement vector $\mathbf{y} = \Phi\mathbf{x} + \mathbf{w}$. Each entry of the sensing matrix Φ of size $m \times n$ is generated by drawing independent samples from the $\mathcal{N}(0,1)$ distribution, where $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian distribution with mean μ and variance σ^2. The locations of the s nonzero entries of the ground truth sparse signal $\mathbf{x}$ are chosen uniformly at random from n possible locations, and the values of the nonzero entries are generated by drawing independent samples from the $\mathcal{N}(0,1)$ distribution. The additive measurement noise $\mathbf{w}$ is assumed to follow the $\mathcal{N}(0, \sigma^2)$ distribution. The experiments are conducted for two different sparsity levels, namely $s = 8$ and $s = 16$. To study the effect of the measurement noise $\mathbf{w}$, we repeat the experiments for various noise strengths, controlled by appropriately choosing σ. The reconstruction mean-squared error (MSE), defined as $\text{MSE} = \left(\frac{1}{n}\sum_{i=1}^{n} |\mathbf{x}_i - \hat{\mathbf{x}}_i|_2^2\right)^{\frac{1}{2}}$, where $\hat{\mathbf{x}}$ and $\mathbf{x}$ denote the reconstructed signal and the ground truth, respectively, is used as the performance metric for evaluation and comparison.

We perform a comparative analysis of the proposed smoothed LASSO algorithm with four competing techniques for solving the LASSO regression. A brief descriptions of the techniques considered for comparison is provided in the following.

(i) **Iterative soft thresholding algorithm** (ISTA) [18]: Starting from an initial guess $\mathbf{x}_0$, one updates the estimate of the underlying sparse signal $\mathbf{x}$ as $\mathbf{x}_{k+1} = T_{\lambda\mu}(\mathbf{x}_k - \mu\nabla f(\mathbf{x}_k))$, where T_γ is the soft-thresholding operator with threshold γ, and μ is the step-size. The convergence rate of ISTA is given by $\mathcal{O}\left(\frac{1}{k}\right)$.

(ii) **Fast iterative shrinkage-thresholding algorithm** (FISTA): This algorithm is an improved version of the ISTA algorithm, with a convergence rate of $\mathcal{O}\left(\frac{1}{k^2}\right)$ [19]. The iterations of FISTA proceed by repeating the following steps:

$$\begin{aligned} \mathbf{x}_k &= T_{\lambda\mu}(\mathbf{y}_k - \mu\nabla f(\mathbf{y}_k)), t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}, \text{ and} \\ \mathbf{y}_{k+1} &= \mathbf{x}_k + \frac{t_k - 1}{t_{k+1}}(\mathbf{x}_k - \mathbf{x}_{k-1}). \end{aligned}$$

Similar to ISTA, one has to initialize the FISTA algorithm by choosing $\mathbf{x}_0$ and t_1 appropriately.

(iii) **Iteratively reweighted least squares** (IRLS) [20]: The fundamental idea behind this algorithm is to minimize the LASSO objective by solving a series of reweighted quadratic minimization problems iteratively. In the k^{th} iteration of the algorithm, one updates the estimate $\mathbf{x}_k$ of $\mathbf{x}$ by computing

$$\mathbf{x}_k = \left(\Phi^T\Phi + 2\lambda W_{k-1}\right)^{-1}\Phi^T\mathbf{y}, \tag{4}$$

where W_{k-1} is a diagonal weight matrix, whose j^{th} entry is updated as $W_{k-1}(j) = \frac{1}{|\mathbf{x}_{k-1}(j)|+\delta}$ using the previous estimate $\mathbf{x}_{k-1}$. A small positive constant δ is added to the denominator in order to ensure numerical stability of the algorithm.

(iv) **Minimization using the CVX package**: We use the CVX optimization package [21], that uses the SDPT3 solver, for minimizing the LASSO cost function and compare its output with the one produced by other algorithms. The solution produced by the CVX package is used to benchmark the performance of different algorithms in terms of reconstruction MSE.

The MSE of the reconstructed signals, obtained using different techniques, for various values of the sparsity parameter s and the input SNR are reported in Table 1. The entries in Table 1 are obtained by averaging over 50 independent trials of the experiment. The iterations of ISTA, FISTA, and the smoothed LASSO algorithms are repeated $I = 50$ times. The values of the smoothing parameter α and the regularization parameter λ, corresponding to various input SNR levels, are also indicated in Table 1. We observe that the smoothed LASSO algorithm consistently outperforms ISTA and performs on par with FISTA and IRLS. Note that the smoothed LASSO algorithm employs an update of the estimate by using only the first-order information about the cost, whereas the IRLS algorithm requires the inversion of an $n \times n$ matrix in every iteration. Consequently, the complexity of updating the estimate in the smoothed LASSO algorithm is considerably less compared to the IRLS algorithm.

3.1.1. Empirical behavior of the LASSO objective

We study the rate of decay of the LASSO objective for different algorithms as a function of the number of iterations, and the result is shown in Fig. 2. All algorithms are initialized with $\mathbf{x}_0 = 0$ to facilitate a fair comparison of the rate of decrement of the cost function. The plots in Fig. 2 are obtained by averaging over 50 independent trials of the experiment. The dimension n of the ground truth sparse signal and the number of measurements m are taken as $n = 256$ and $m = 128$, respectively. The sparsity level s is chosen to be $s = 16$ and the value of input SNR is 30 dB for this experiment. The value of the parameters λ and the smoothing parameter α are set at $\lambda = 0.06$ and $\alpha = 100$. The signal generation process is identical to that explained in Section 3.1. We observe that the IRLS algorithm results in the fastest decay of the objective function as compared to all other techniques, but the improvement in the decay-rate comes at the price of increased per-iteration complexity. Note that the update rule (4) of the IRLS algorithm requires the inversion of an $n \times n$ matrix in every iteration, requiring $\mathcal{O}(n^3)$ computations. On the contrary, the smoothed LASSO, FISTA, and ISTA entail $\mathcal{O}(n^2)$ complexity in each iteration, as one needs to compute only the gradients of the respective cost functions to obtain the updated estimates. As the number of iterations exceeds 12, the cost function for smoothed LASSO

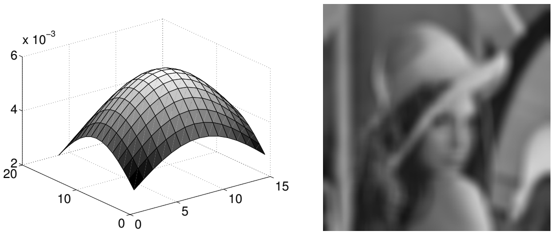

Fig. 3. The effect of a Gaussian blur kernel of size $p = 15$ and standard deviation $\sigma_0 = 8$ on the *Lenna* image.

becomes smaller than that obtained using ISTA and approaches that of FISTA. Thus, although the proposed algorithm has a theoretical upper bound of $\mathcal{O}\left(\frac{1}{K}\right)$ on the rate of convergence, it performs on par experimentally with FISTA, having a quadratic rate of convergence. The explanation of this phenomenon can be given by alluding to the fact that the upper bound on the convergence rate established in Theorem 1 is not tight, and consequently the actual convergence rate might be better than $\mathcal{O}\left(\frac{1}{K}\right)$, as suggested by the experiments.

3.2. Image deblurring

In this experiment, the objective is to recover an image from its blurred and noisy measurement. The degraded image is obtained by applying a Gaussian blur kernel of size $p \times p$ with standard deviation σ_0 on the image, followed by additive Gaussian noise corruption. In Fig. 3, we show the blurring effect obtained by applying a Gaussian kernel with $p = 15$ and $\sigma_0 = 8$ on the *Lenna* image. Increasing the size p and the standard deviation σ_0 of the kernel results in more intense blurring, due to a stronger averaging effect. The degraded image $\mathbf{Y}$ thus obtained can be expressed as

$$\mathbf{Y} = \mathcal{A}\left(\mathbf{X}\right) + \mathbf{W},$$

where $\mathcal{A}$ is a linear operator that represents the blurring operation, $\mathbf{X}$ is the original image, and $\mathbf{W}$ denotes the additive noise. In recovering $\mathbf{X}$ from its linearly degraded measurement $\mathbf{Y}$, we exploit the sparsity of natural images in the discrete cosine transform (DCT) domain. Let $\mathcal{D}$ denote the two-dimensional (2D) DCT operator. It is well-known that the 2D-DCT of an image $\mathbf{X}$, given by $\mathbf{X}_u = \mathcal{D}\left(\mathbf{X}\right)$, is sparse, that is, the most of the entries in $\mathbf{X}_u$ are small in magnitude and the energy is concentrated in only few coefficients. Thus, the 2D-counterpart of the LASSO regression that we intend to solve for recovering $\mathbf{X}$ from $\mathbf{Y}$ is given by

$$\hat{\mathbf{X}}_u = \arg\min_{\mathbf{X}_u} \frac{1}{2}\left\|\mathbf{Y} - \mathcal{A}\mathcal{D}^{-1}\left(\mathbf{X}_u\right)\right\|_F^2 + \lambda\left\|\mathbf{X}_u\right\|_1, \quad (5)$$

where $\mathcal{D}^{-1}$ denotes the inverse of the 2D-DCT operation. The estimate $\hat{\mathbf{X}}$ of the original image $\mathbf{X}$ is obtained by computing $\hat{\mathbf{X}} = \mathcal{D}^{-1}\left(\hat{\mathbf{X}}_u\right)$. Note that the symbol $\left\|\mathbf{X}_u\right\|_1$ in (5) denotes the sum of the absolute values of the entries in $\mathbf{X}_u$.

A comparative analysis of the proposed algorithm with FISTA is performed, in terms of two reconstruction quality metrics, namely, peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). We use the publicly available implementation of FISTA [22] for comparison of the deblurring performance. The blur kernel is applied on the images after the pixels are normalized to take values

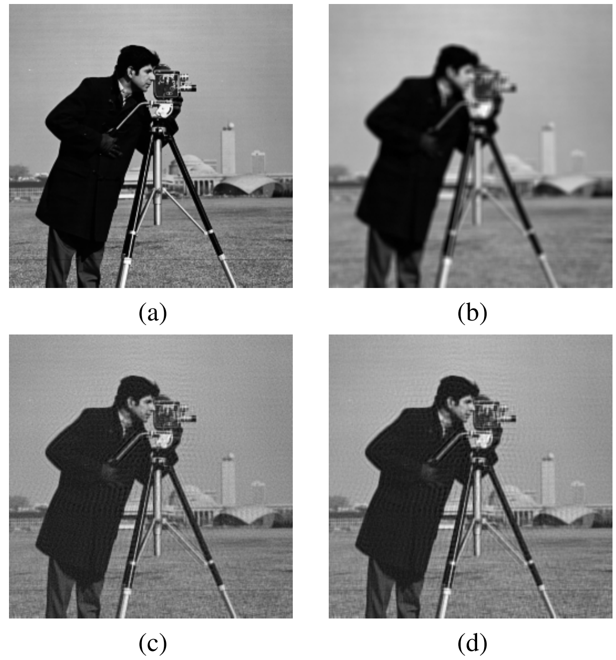

Fig. 4. Image deblurring experiment: (a) Ground-truth; (b) Blurred and noisy measurement, PSNR $=$ 23.17 dB, SSIM $=$ 0.75; (c) Deblurred using FISTA, PSNR $=$ 29.47 dB, SSIM $=$ 0.77; (d) Deblurred using the smoothed LASSO algorithm, PSNR $=$ 29.43 dB, SSIM $=$ 0.82. Execution times for (c) and (d) are 2.40 and 1.72 seconds, respectively, for 100 iterations. We choose $\lambda = 10^{-4}$ and $\alpha = 500$ in the experiment.

in the range $[0, 1]$. This process is followed by noise addition. For visual comparison, we show the reconstructed images using FISTA and the smoothed LASSO algorithm in Fig. 4, where a Gaussian kernel with $p = 5$ and $\sigma_0 = 4$ is used on the *Cameraman* image for blurring and the additive noise is assumed to be following a Gaussian distribution of zero mean and standard deviation $\sigma = 2 \times 10^{-3}$. The number of iterations for FISTA and the smoothed LASSO algorithms is chosen to be 100, resulting in run-times of 2.40 and 1.72 seconds, respectively. The faster execution of the smoothed LASSO algorithm is due to the absence of the soft-thresholding block, unlike FISTA. We observe that the PSNR and the SSIM values of the reconstructed images, as indicated in Fig. 4, are almost identical for both techniques.

In Table 2, we report the PSNR and SSIM values of the reconstructed images obtained using the FISTA and the smoothed-LASSO algorithms, for various values of p and σ_0. Each iteration of FISTA and smoothed LASSO consumes $t_{\text{sl}} = 0.017$ and $t_{\text{fista}} = 0.024$ seconds, respectively, on the specified platform. The execution time of FISTA is slightly higher owing to the soft-thresholding operation, which is not required for smoothed LASSO. For a fair comparison, the number of iterations of FISTA and smoothed LASSO is chosen to be $I_{\text{fista}} = 100$ and $I_{\text{sl}} = \frac{I_{\text{fista}} t_{\text{fista}}}{t_{\text{sl}}} \approx 140$, respectively, so that the total execution time is same both algorithms. The parameters α and λ are taken as $\alpha = 500$ and $\lambda = 10^{-4}$. The standard deviation of the additive Gaussian noise is $\sigma = 2 \times 10^{-3}$ and the entries in Table 2 are obtained by averaging over 10 independent noise realizations. We observe in Table 2 that the FISTA and the smoothed LASSO

Images	Blur kernel parameters (p, σ_0)	FISTA	Smoothed LASSO
	$p = 5, \sigma_0 = 4$	29.47 (0.7741)	29.47 (0.8155)
	$p = 5, \sigma_0 = 8$	30.31 (0.8343)	30.66 (0.8706)
	$p = 7, \sigma_0 = 4$	28.84 (0.8108)	28.79 (0.8229)

Table 2. Comparison of the image deblurring performance of FISTA and smoothed LASSO, in terms of PSNR (dB) and SSIM of the output images, on three different images for various blur kernels. The numbers inside the parentheses indicate the corresponding SSIM values. The input PSNR and SSIM values, corresponding to the first, second, and the third rows are given by $(23.18\,\text{dB}, 0.75)$, $(25.17\,\text{dB}, 0.78)$, and $(24.20\,\text{dB}, 0.69)$, respectively.

algorithms result in almost identical performance, consistently for different p and σ_0 values, in terms of PSNR. The SSIM values of the reconstructed images for smoothed LASSO is observed to be slightly better compared to FISTA. This behavior is consistent with the results shown in Fig. 4 and indicates that the smoothed LASSO algorithm is slightly better than FISTA in terms of SSIM, and on par with FISTA in terms of PSNR, for a fixed execution time budget.

4. CONCLUSIONS

We have proposed to solve the LASSO regression problem by minimizing a smooth approximation to the actual non-smooth cost function. The resulting optimization is solved by applying an accelerated gradient descent technique on the smooth approximation. We have proved that the resulting smoothed LASSO algorithm possesses a linear rate of convergence. We have demonstrated, through experiments on synthesized signals, that smoothed LASSO yields improved performance over ISTA in terms of MSE and performs on par with FISTA. In case of image deblurring experiment, we have shown that the smoothed LASSO technique results in reconstruction performance slightly superior to that obtained using the FISTA algorithm, for a given run-time constraint.

5. REFERENCES

[1] E. J. Candés and M. Wakin, “An introduction to compressive sampling,” *IEEE Signal Process. Mag.*, vol. 25, pp. 21 –30, Mar. 2008.

[2] R. Baraniuk, “Compressive sensing," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul 2007.

[3] E. J. Candés, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Info. Theory,* vol. 52, no. 2, pp. 489–509, Feb. 2006.

[4] M. Lustig, D. Donoho, and J. M. Pauly, “Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, Dec. 2007.

[5] M. Sheikh, S. Sarvotham, O. Milenkovic, and R. Baraniuk, “DNA array decoding from nonlinear measurements by belief propagation," in *Proc. IEEE Workshop on Statistical Signal Process.* (SSP), Aug. 2007.

[6] R. Baraniuk and P. Steeghs, “Compressive radar imaging," in *Proc. IEEE Radar Conf.*, Apr. 2007.

[7] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Info. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[8] D. Needell and J. A. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *arXiv: 0803.2392v2*, Apr. 2008.

[9] D. L. Donoho, “For most large underdetermined systems of linear equations, the minimal ℓ_1-norm near-solution approximates the sparsest near-solution," *Comm. on Pure and Applied Math.*, vol. 59, no. 7, pp. 907–934, Jul. 2006.

[10] E. J. Candés, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements," *arXiv: 0503066v2*, Dec. 2005.

[11] R. Tibshirani, “Regression shrinkage and selection via the LASSO," *J. Royal Stat. Society, Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[12] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[13] M. J. Wainwright, “Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso)", *IEEE Trans. Info. Theory*, vol. 55, no. 5, pp. 2183–2202, May 2009.

[14] K. Knight and W. J. Fu, “Asymptotics for lasso-type estimators," *Ann. Statist.*, vol. 28, pp. 1356–1378, 2000.

[15] M. Schmidt, G. Fung, and R. Rosales, “Fast optimization methods for ℓ_1 regularization: A comparative study and two new approaches," in *Proc. European Conf. on Machine Learning*, pp. 286–297, 2007.

[16] Y. E. Nesterov, “A method for solving the convex programming problem with convergence rate $\mathcal{O}\left(\frac{1}{k^2}\right)$," *Dokl. Akad. Nauk SSSR*, vol. 269, pp. 543–547, 1983.

[17] Y. E. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publishers, London, 2004.

[18] I. Daubechies, M. Defrise, and C. D. Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math.*, vol. 57, pp. 1413–1457, 2004.

[19] A. Beck and M. Teboulle, “A fast iterative shrinkage thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sciences*, vol. 2, no. 1, pp. 183-202, 2009.

[20] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, “Iteratively re-weighted least squares minimization for sparse recovery," *Communications on Pure and Applied Math.*, vol. LXIII, pp. 1–38, 2010.

[21] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming," version 2.0 beta, Sep. 2013.

[22] A. Beck and M. Teboulle, *The wavelet-FISTA toolbox*, available at *http://iew3.technion.ac.il/Home/Users/becka.html*.