Fast sparse optimization via adaptive shrinkage

Fast sparse optimization via adaptive shrinkage

Vito Cerone * **Sophie M. Fosson** * **Diego Regruto** *

* *Department of Control and Computer Engineering, Politecnico di Torino, Italy (e-mail: sophie.fosson@polito.it).*

Abstract. The need for fast sparse optimization is emerging, e.g., to deal with large-dimensional data-driven problems and to track time-varying systems. In the framework of linear sparse optimization, the iterative shrinkage-thresholding algorithm is a valuable method to solve Lasso, which is particularly appreciated for its ease of implementation. Nevertheless, it converges slowly. In this paper, we develop a proximal method, based on logarithmic regularization, which turns out to be an iterative shrinkage-thresholding algorithm with adaptive shrinkage hyperparameter. This adaptivity substantially enhances the trajectory of the algorithm, in a way that yields faster convergence, while keeping the simplicity of the original method. Our contribution is twofold: on the one hand, we derive and analyze the proposed algorithm; on the other hand, we validate its fast convergence via numerical experiments and we discuss the performance with respect to state-of-the-art algorithms.

Keywords: Sparse learning, optimization, estimation, iterative/recursive algorithms, proximal algorithms, accelerated algorithms.

1. INTRODUCTION

Sparse optimization consists in learning sparse models through the solution of suitable optimization problems. We call "sparse" those models that depend on a reduced number of parameters, which is a desirable condition for several motivations, ranging from the decrease of numerical complexity and memory footprint to the circumvention of overfitting; see, e.g., Hastie et al. (2015); Brunton and Kutz (2019) for an overview. Recently, the attention on sparsity is increasing in system identification and in machine learning, on the one hand to select models that are physically interpretable, on the other hand, to train models that can be embedded in devices with limited resources, such as mobile applications; see, e.g., Brunton et al. (2016); Zhao et al. (2020); Louizos et al. (2018) for different perspectives on this topic.

Nowadays, the optimization techniques to learn sparse models are quite mature, in particular in the context of linear systems. The Lasso problem, proposed in Tibshirani (1996), is a popular, effective approach, that combines least squares and ℓ_1 minimization to search sparse solutions via convex optimization. However, the development of fast algorithms for sparse optimization is still an open problem. In fact, in case of large-dimensional datasets, the convexity is not sufficient to guarantee solutions in a reasonable time. This aspect becomes critical, e.g., to identify time-varying or hybrid systems, where the parameters either evolve in time or switch among different unknown modes; in this framework, a prompt identification is necessary to track the dynamics; see, e.g., Lauer and Bloch (2019); Fosson (2021).

In the specific case of Lasso, several methods are proposed in the literature to minimize it efficiently. Among the iterative algorithms, the iterative shrinkage-thresholding algorithm (ISTA) proposed by Daubechies et al. (2004) is a simple proximal method, straightforward to implement even in a distributed context. Similarly, the alternating direction method of multipliers (ADMM) is low-complex and prone to distributed and parallelized computation; see Boyd et al. (2010) for a complete overview. ISTA and ADMM share an R-linear convergence rate, as proven in Bredies and Lorenz (2008); Hong and Luo (2017), while in practice ADMM is usually faster. Methods to accelerate ISTA are proposed in the literature, such as FISTA, see Beck and Teboulle (2009), which improves the nonasymptotic global rate of convergence from $\mathcal{O}\left(t^{-1}\right)$ to $\mathcal{O}\left(t^{-2}\right)$.

In this paper, we analyze a different strategy for fast sparse optimization. We start from the development of a proximal gradient method for a Lasso-kind problem with logarithmic regularization, here denoted as Log-Lasso, and we obtain a variant of ISTA whose shrinkage hyperparameter is adaptive, i.e., it is updated at each iteration, based on the current estimate. This feature makes the algorithm significantly faster than ISTA. Throughout the paper, we name AD-ISTA the proposed adaptive ISTA.

More precisely, by starting from the analysis in Dahlke et al. (2012), we show that ISTA, FISTA and ADMM share a common behavior in the minimization of Lasso: in a first phase, they decrease the least squares term, while the ℓ_1-norm substantially increases; in a second phase, they reduce the ℓ_1-norm while keeping the least squares

⋆ This paper is part of the project NODES which has received funding from the MUR – M4C2 1.5 of PNRR with grant agreement no. ECS00000036.

error almost constant. This trajectory yields a transient overestimation of the sparsity, which is time-consuming. In contrast, the proposed approach is not affected by this drawback. In this work, we limit the investigation to linear systems and Lasso, while the proposed methodology might be extended to different optimization problems.

In summary, our contributions are as follows. Firstly, we illustrate how to apply a proximal method to Log-Lasso, which results into AD-ISTA, and we propose a straightforward analysis of its convergence; contextually, we show that AD-ISTA is equivalent to an ISTA with adaptive shrinkage parameter. Secondly, we illustrate via numerical experiments that AD-ISTA improves the trajectory with respect to classical algorithms, which results in a faster convergence. Moreover, by applying the approach of the fast ISTA (FISTA) proposed by Beck and Teboulle (2009), we propose an accelerated version of AD-ISTA, called AD-FISTA. Finally, we also compare AD-ISTA and AD-FISTA to ℓ_1 reweighting methods.

We organize the paper as follows. In Sec. 2, we state the problem. In Sec. 3, we derive and illustrate the proposed approach, while Sec. 4 discusses the expected faster behavior. This is validated via numerical experiments in Sec. 6, where we add to the comparison the related algorithms illustrated in Sec. 5; finally, we draw some conclusions.

2. PROBLEM STATEMENT

Let us consider sparse optimization problems of the kind

$$\min_{x\in\mathbb{R}^n} \mathcal{G}(x) + \mathcal{R}_\alpha(x) \tag{1}$$

where $\mathcal{G} : \mathbb{R}^n \mapsto \mathbb{R}^+$ is a convex, smooth cost functional and

$$\mathcal{R}_\alpha(x) := \sum_{i=1}^{n} \alpha_i r(x_i) \tag{2}$$

is a sparsity promoting regularization with weights $\alpha = (\alpha_1, \dots, \alpha_n)$, $\alpha_i \geq 0$ for each $i = 1, \dots, n$. Usually, $\mathcal{R}_\alpha(x)$ is not differentiable at $x = 0$, and it may be non-convex.

To solve this class of composite non-smooth problems, where gradient descent is not feasible, we can exploit the proximal gradient method (PGM). PGM consists in iterating a Landweber step on $\mathcal{G}$, i.e., a gradient descent step with constant stepsize $\tau > 0$, and a proximal mapping. More precisely, by defining $\lambda := \tau\alpha \in \mathbb{R}^n$ and

$$\text{prox}_{\mathcal{R}_\lambda}(z) := \underset{x\in\mathbb{R}^n}{\text{argmin}} \left(\mathcal{R}_\lambda(x) + \frac{1}{2}\|x - z\|_2^2 \right), \tag{3}$$

PGM is as follows: for $t = 0, 1, 2, \dots$

$$x_{t+1} = \text{prox}_{\mathcal{R}_\lambda}\left(x_t - \tau\nabla\mathcal{G}(x_t)\right). \tag{4}$$

The convergence of $\mathcal{G}(x_t) + \mathcal{R}_\alpha(x_t)$, where x_t is the sequence of iterates of PGM, is proven when $\mathcal{R}_\lambda$ is convex, $\mathcal{G}$ has Lipschitz continuous gradient, with constant L, and $\tau = \frac{2}{L}$. Specifically, if $x^\star \in \mathbb{R}^n$ is a minimizer, then $\mathcal{G}(x_t)+\mathcal{R}_\alpha(x_t)$ converges to the minimum $\mathcal{G}(x^\star)+\mathcal{R}_\alpha(x^\star)$. In addition, if $\mathcal{G}$ is strongly convex, the convergence of x_t to the (unique) minimizer is also proven; see, e.g., Combettes and Pesquet (2011); Calafiore and El Ghaoui (2014).

Lasso is an instance of problem (1): given $A \in \mathbb{R}^{m,n}$ and a vector of measurements $y \in \mathbb{R}^m$, Lasso corresponds to $\mathcal{G}(x) = \frac{1}{2}\|Ax-y\|_2^2$ and $\mathcal{R}_\alpha(x) = \alpha\|x\|_1$, with scalar $\alpha > 0$. On the other hand, PGM applied to Lasso corresponds to ISTA. The convergence of ISTA can be proven in a peculiar way, developed by Daubechies et al. (2004), that exploits the definition of a surrogate functional. With this approach, the sequence x_t generated by ISTA is proven to converge to a minimizer, even though $\mathcal{G}(x)$ is not strongly convex. We notice that $\frac{1}{2}\|Ax - y\|_2^2$ is not strongly convex when $m < n$, which allows to efficiently use ISTA in compressed sensing; see, e.g., Fornasier (2010).

The development of our approach starts by considering a Lasso with possible non-convex regularization $\mathcal{R}_\alpha$, i.e.,

$$\min_{x\in\mathbb{R}^n} \mathcal{F}(x) := \frac{1}{2}\|Ax - y\|_2^2 + \mathcal{R}_\alpha(x). \tag{5}$$

In principle, we can apply PGM to solve (5). However, we have to cope with two critical points. On the one hand, the computation of $\text{prox}_{\mathcal{R}_\lambda}$ may be not straightforward when $\mathcal{R}_\lambda$ (or equivalently $\mathcal{R}_\alpha$) is non-convex. On the other hand, to prove the convergence of the algorithm is challenging.

In the literature, some attention is devoted to the application of PGM to problem (5). In particular, Bredies et al. (2015) provide conditions under which $\mathcal{F}(x_t)$ converges in infinite-dimensional Hilbert spaces, by leveraging the results in Attouch et al. (2011), and they analyze the specific case of ℓ_p regularization. Moreover, Bayram (2016) analyze the convergence of x_t for (5) when $\mathcal{R}_\lambda(x)$ is weakly convex and $A^T A$ is positive definite, which is not the case, e.g., of compressed sensing.

In the following, we focus on the case

$$r(x_i) = \log(|x_i| + \epsilon) \tag{6}$$

where $\epsilon > 0$ is a design hyperparameter that tunes the concavity of the function. We call Log-Lasso the problem (5)-(2)-(6). We consider Log-Lasso because log-regularization is known to be efficient, see, e.g., Candès et al. (2008); however, the proposed approach might be extended also to other classes of non-convex regularizers.

3. PROPOSED APPROACH

In this section, we develop the proposed approach by starting from the application of PGM to (5)-(6). Firstly, we analyze the convergence of PGM applied to (5); then, we deal with the computation of the proximal operator of the log-regularizer.

Given $\mathcal{F}(x)$ defined in (5), we introduce the surrogate functional

$$\mathcal{S}(x, \zeta) = \mathcal{F}(x) + \frac{1}{2\tau}\|x - \zeta\|_2^2 - \frac{1}{2}\|Ax - A\zeta\|_2^2 \tag{7}$$

where $\zeta \in \mathbb{R}^n$ is an auxiliary variable. We assume $\tau < \|A\|_2^{-2}$, so that

$$\frac{1}{\tau}\|x-\zeta\|_2^2 - \|Ax-A\zeta\|_2^2 \geq \left(\frac{1}{\tau} - \|A\|_2^2\right)\|x-\zeta\|_2^2 \geq 0 \tag{8}$$

where the equality holds only for $x = \zeta$. Then, we proceed by alternating minimization of (7) with respect to x and to ζ. According to (8),

$$x = \underset{\zeta\in\mathbb{R}^n}{\text{argmin}}\, \mathcal{S}(x, \zeta). \tag{9}$$

On the other hand, we have

$$\underset{x\in\mathbb{R}^n}{\text{argmin}}\, \mathcal{S}(x, \zeta) = \text{prox}_{\mathcal{R}_\lambda}\left(\zeta + \tau A^T(y - A\zeta)\right). \tag{10}$$

Then,

$$x_{t+1} = \text{prox}_{\mathcal{R}_\lambda}\left(x_t + \tau A^T(y - Ax_t)\right). \tag{11}$$

We notice that $\mathcal{F}(x_t) = \mathcal{S}(x_t, x_t) \geq \mathcal{S}(x_{t+1}, x_t) \geq \mathcal{S}(x_{t+1}, x_{t+1}) = \mathcal{F}(x_{t+1}) \geq 0$, i.e., $\mathcal{F}(x_t)$ is a non-increasing, bounded sequence. This yields the following result.

Lemma 1. Given the sequence x_t generated by PGM applied to (5), $\mathcal{F}(x_t)$ converges. Moreover, PGM applied to (5) defines an asymptotically regular map, that is, $\lim_{t\to\infty} \|x_{t+1} - x_t\|_2 = 0$.

In the following lemma, we specify how to compute $\text{prox}_{\mathcal{R}_\lambda}(z) = \underset{x\in\mathbb{R}^n}{\text{argmin}}\, \mathcal{R}_\lambda(x) + \frac{1}{2}\|x - z\|_2^2$ in case of log-regularizer (6), for any $z \in \mathbb{R}^n$. Since the problem is separable, we explicitly evaluate the minimum for each component $i = 1, \ldots, n$. From now onwards, we tune the hyperparameters $\lambda = \tau\alpha$ and ϵ such that

Assumption 1. For each $i = 1, \ldots, n$, $\lambda_i < \epsilon^2$.

Lemma 2. Under Assumption 1, given any $z_i \in \mathbb{R}$,

$$\underset{x_i\in\mathbb{R}}{\text{argmin}}\, \mu(x_i) := \lambda_i \log(|x_i| + \epsilon) + \frac{1}{2}(x_i - z_i)^2 = \\ = \begin{cases} z_i - \gamma_i(z_i) & \text{if } z_i > \dfrac{\lambda_i}{\epsilon} \\ z_i + \gamma_i(z_i) & \text{if } z_i < -\dfrac{\lambda_i}{\epsilon} \\ 0 & \text{if } z_i \in \left[-\dfrac{\lambda_i}{\epsilon}, \dfrac{\lambda_i}{\epsilon}\right] \end{cases} \tag{12}$$

where

$$\gamma_i(z_i) := \frac{|z_i| + \epsilon - \sqrt{(|z_i| + \epsilon)^2 - 4\lambda_i}}{2}. \tag{13}$$

Proof. Let us consider the case $x_i \in [0, +\infty)$.

For $x_i \in (0, +\infty)$, we have $\mu'(x_i) = \frac{\lambda_i}{x_i+\epsilon} + x_i - z_i$, and $\mu''(x_i) = 1 - \frac{\lambda_i}{(x_i+\epsilon)^2} > 1 - \frac{\lambda_i}{\epsilon^2}$. Thus, $\mu''(x_i) > 0$ under Assumption 1, i.e., $\mu(x_i)$ is strongly convex in $[0, +\infty)$. Therefore, if there exists a stationary point, it corresponds to the unique minimum; otherwise, the minimum is at $x_i = 0$, since μ is strongly convex in $[0, +\infty)$ and $\lim_{x_i\to+\infty} \mu(x_i) = +\infty$.

Let us compute the possible stationary points. We notice that $x_i + \epsilon \neq 0$ because $x_i \in [0, +\infty)$. Therefore,

$$\begin{aligned} \mu'(x_i) = 0 \quad &\Leftrightarrow \lambda_i + (x_i - z_i)(x_i + \epsilon) = 0 \\ &\Leftrightarrow x_i = \frac{z_i - \epsilon \pm \sqrt{(z_i + \epsilon)^2 - 4\lambda_i}}{2}. \end{aligned} \tag{14}$$

Then, we have to check whether these two solutions are consistent with the condition $x_i \geq 0$.

First of all, it is straightforward to evaluate that $(z_i+\epsilon)^2 - 4\lambda_i \geq 0$ whenever $z_i \in \Omega := (-\infty, -2\sqrt{\lambda_i} - \epsilon) \cup (2\sqrt{\lambda_i} - \epsilon, +\infty)$. Afterwards,

$$z_i - \epsilon + \sqrt{(z_i + \epsilon)^2 - 4\lambda_i} > 0 \Leftrightarrow z_i \in \left(\frac{\lambda_i}{\epsilon}, +\infty\right) \subset \Omega;$$

$$z_i - \epsilon - \sqrt{(z_i + \epsilon)^2 - 4\lambda_i} < 0 \text{ for any } z_i \in \Omega.$$

We remark that $\left(\frac{\lambda_i}{\epsilon}, +\infty\right) \subset \Omega$ because $2\sqrt{\lambda_i} - \epsilon < \frac{\lambda_i}{\epsilon}$ under Assumption 1.

In conclusion, if $x_i \in [0, +\infty)$, the minimizer is $x_i = \frac{z_i - \epsilon + \sqrt{(z_i+\epsilon)^2 - 4\lambda_i}}{2}$ if $z_i > \frac{\lambda_i}{\epsilon}$; otherwise, if $z_i \leq \frac{\lambda_i}{\epsilon}$, the minimizer is $x_i = 0$.

We omit the computations for the case $x_i \in (-\infty, 0)$, which are based on the same ideas and yield symmetric conclusions. To sum up,

$$\underset{x_i\in\mathbb{R}}{\text{argmin}}\, \lambda_i \log(|x_i| + \epsilon) + \frac{1}{2}(x_i - z_i)^2 = \\ = \begin{cases} \dfrac{z_i - \epsilon + \sqrt{(z_i + \epsilon)^2 - 4\lambda_i}}{2} & \text{if } z_i > \dfrac{\lambda_i}{\epsilon} \\ \dfrac{z_i + \epsilon - \sqrt{(z_i - \epsilon)^2 - 4\lambda_i}}{2} & \text{if } z_i < -\dfrac{\lambda_i}{\epsilon} \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

From (15), we easily derive the thesis. □

Remark 1. The evaluation of the proximal operator is more complex for ℓ_p. As one can see in Zuo et al. (2013); Bredies et al. (2015), no closed form is provided. This makes the choice of log regularization more affordable. On the other hand, the results in (Bredies et al., 2015, Lemma 3.3) that define an exact formula for the proximal operation do not envisage the logarithmic case, as Assumption 3.2 in Bredies et al. (2015) requires that $r'(x_i) \to \infty$ for $x_i \to 0$, which is not our case.

Lemma 2 provides an interesting interpretation of PGM applied to Log-Lasso. In fact, according to Lemma 2, we can formulate the PGM iteration as

$$\begin{aligned} z_t &= x_t + \tau A^T(y - Ax_t), \\ x_{t+1} &= \mathrm{S}_{\frac{\lambda}{\epsilon}, \gamma(z_t)}(z_t) \end{aligned} \tag{16}$$

where $\gamma(z) = (\gamma_1(z_1), \ldots, \gamma_n(z_n))$ is defined in (13) and $\mathrm{S}_{\frac{\lambda}{\epsilon}, \gamma(z)}(z)$ is a shrinkage-thresholding operator, defined componentwise as

$$\mathrm{S}_{\frac{\lambda_i}{\epsilon}, \gamma_i(z_i)}(z_i) := \begin{cases} z_i - \gamma_i(z_i) & \text{if } z_i > \dfrac{\lambda_i}{\epsilon} \\ z_i + \gamma_i(z_i) & \text{if } z_i < -\dfrac{\lambda_i}{\epsilon} \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

In (17), $\gamma_i(z_i)$ is the shrinkage value, while $\frac{\lambda_i}{\epsilon}$ is the threshold below which variables are set to zero. We notice that, by using the notation of (17), we can write ISTA as

$$x_{t+1} = \mathrm{S}_{\lambda,\lambda}\left(x_t + \tau A^T(y - Ax_t)\right) \tag{18}$$

because in ISTA the shrinkage and thresholding hyperparameters are the same. In particular, in (18), λ is time-invariant and constant for each component i. In contrast, (16) represents a generalization of ISTA, where shrinkage and thresholding hyperparameters, namely $\gamma(z)$ and $\frac{\lambda}{\epsilon}$, are different. In particular, $\gamma(z)$ is time-varying, as it adapts to the current value of $x_t + \tau A^T(y - Ax_t)$ and it penalizes more the values closer to zero. This penalization is "less democratic" than the one of ISTA, and it causes less bias in the non-zero components of the solutions. The proposed algorithm (16) is an adaptive ISTA (AD-ISTA), because the shrinkage is adaptive.

4. WHY AD-ISTA IS FASTER THAN ISTA?

In this section, we discuss why AD-ISTA is expected to be faster than ISTA. The aim of this analysis is to illustrate the role played by the shrinkage hyperparameter

in determining the trajectory of the algorithm. More rigorous proofs are left for extended work.

As to ISTA, the role of λ in the speed of convergence is fundamental. In general, by increasing λ we obtain a faster algorithm; on the other hand, a too large λ would cause a substantial bias in the solution. Prior information on the solution can be used to set λ. As an extreme example, if the solution has support $\mathcal{S}$, $|\mathcal{S}| = k \ll n$, we assume $\lambda \in \mathbb{R}^n$ where $\lambda_i \notin \mathcal{S}$ are very large, then the correct support is identified in one step, and the problem is immediately reduced to dimension $k \ll n$, which substantially reduces the number of iterations.

In Daubechies et al. (2008), the Authors analyze the dynamics of ISTA for Lasso: ISTA first reduces the residual $\|Ax - y\|_2^2$ and contextually overestimates the ℓ_1-norm; then, it corrects back the ℓ_1-norm. This causes a "long detour" which yields slow convergence, see (Daubechies et al., 2008, Fig.1). For this motivation, the Authors propose to constrain the ℓ_1-norm within a given ball; however, this does not result in a clear acceleration of the method.

The idea is further developed in (Dahlke et al., 2009, Section 1.2), where the Authors propose to project the Landweber iteration ℓ_1-balls with slowly increasing radius. In practice, they implement the idea by proposing an ISTA with decreasing shrinkage-thresholding hyperparameter. This algorithm, called D-ISTA is further analyzed in Dahlke et al. (2012). In particular, it is proven to converge with R-linear rate under some conditions, e.g, by assuming that the hyperparameter decreases geometrically. However, choosing a suitable decreasing hyperparameter is challenging.

In AD-ISTA we solve this issue, because the hyperparameter adapts to the magnitude of the gradient step over the previous estimate. This provides a larger shrinkage for small values in magnitude. In particular, if we start from the natural initial condition $x_0 = 0$, and if τ is small, in general the hyperparameter is larger in a first phase, which keeps the ℓ_1-norm small, while as far as the components move away from zero, the shrinkage is smaller. Therefore, to some extent, we have a decreasing behavior, but only for those components that move far from zero. This adaptation is the key motivation that makes AD-ISTA more effective than ISTA.

We remark that in the literature the study of optimal hyperparameters for ISTA currently is an active topic. For example, a learned ISTA (LISTA) is developed in Gregor and LeCun (2010), and subsequently enhanced in, e.g., Liu et al. (2019); Chen et al. (2021). Since ISTA iterates a linear step and a non-linearity, the main structure is similar to a neural network, and techniques to learn the hyperparameters are studied in the mentioned papers. The main drawback of this approach is the time required for the training.

5. RELATED ALGORITHMS

In this section, we present an accelerated version of AD-ISTA and we compare AD-ISTA to an ℓ_1-reweighting method.

5.1 AD-FISTA: a fast version of AD-ISTA

Since AD-ISTA shares the same structure of ISTA, the fast version of ISTA proposed in Beck and Teboulle (2009) and known as FISTA can be applied to it. Basically, FISTA exploits two previous iterates to compute the current estimate and it shares the improved convergence rate $\mathcal{O}\left(\frac{1}{t^2}\right)$, while keeping the low complexity of ISTA per iteration.

The application of FISTA approach to AD-ISTA, that we denote as AD-FISTA, is as follows: given $v_0 = x_0 \in \mathbb{R}^n, u_0 = 1 \in \mathbb{R}$, for any $t = 0, 1, 2, \ldots,$

$$\begin{aligned} z_t &= v_t + \tau A^T(y - Av_t) \\ x_{t+1} &= \mathrm{S}_{\frac{\lambda}{\epsilon},\gamma(z_t)}\left(z_t\right) \\ u_{t+1} &= \frac{1 + \sqrt{1 + 4u_t^2}}{2} \\ v_{t+1} &= x_{t+1} + \frac{u_t - 1}{u_{t+1}}(x_{t+1} - x_t). \end{aligned} \tag{19}$$

For the same motivations illustrated in Beck and Teboulle (2009), we expect that AD-FISTA converges in less iterations than AD-ISTA.

5.2 Comparison to ℓ_1-reweighting ISTA

In the context of sparse optimization, ℓ_1-reweighting techniques are popular to improve the accuracy of ℓ_1 minimization methods. The key idea, proposed in Candès et al. (2008) is to iterate the solution of a Basis Pursuit by updating the weight of the ℓ_1-norm, with the final aim of penalizing less the larger components in magnitude, which is in line with the approach proposed in this work. The algorithm in Candès et al. (2008) leverages the local minimization of a log-concave penalty through its linearization. This yields to weight the ℓ_1-norm with the derivatives of the log-concave penalty. An ISTA-based variant of ℓ_1 reweighting is proposed in (Fosson, 2018, Sec. III), and it is observed to be fast with respect to classic ℓ_1-reweighting. In case of logarithmic penalty, this algorithm, here denoted as RW-ISTA, is as follows:

$$\begin{aligned} (w_t)_i &= \frac{1}{|(x_t)_i| + \epsilon}, \quad i = 1, \ldots, n \\ x_{t+1} &= \mathrm{S}_{\lambda w_t, \lambda w_t}\left(x_t + \tau A^T(y - Ax_t)\right). \end{aligned} \tag{20}$$

Even though RW-ISTA originates from the ℓ_1-reweighting framework, while AD-ISTA is obtained via proximal methods, the final structure of RW-ISTA has an adaptive shrinkage-thresholding parameter as in AD-ISTA. This may explain the increased velocity of RW-ISTA observed in Fosson (2018). On the other hand, in RW-ISTA, shrinkage and thresholding parameters are equal, while, as discussed above, in AD-ISTA they are different.

6. NUMERICAL RESULTS

In this section, we present some numerical results to validate the proposed method.

For our experiments, we consider a matrix $A \in \mathbb{R}^{m,n}$ with $m = 500$ and $n = 1000$, whose components are independently generated with Gaussian distribution $\mathcal{N}(0, \frac{1}{m})$. Given $y = A\widetilde{x} + \eta$, where $\eta \in \mathbb{R}^m$ is an unknown random noise $\sim \mathcal{N}(0, 10^{-2})$, we aim at estimating $\widetilde{x}$, which

has sparsity $k = 10$, and non-zero components randomly generated with uniform distribution, with magnitude in $(1, 2)$. To estimate $\widetilde{x}$, we implement the proposed AD-ISTA and AD-FISTA, and we compare them to ISTA, FISTA, ADMM, and RW-ISTA. We set $\lambda = 10^{-3}$ for Lasso, and $\lambda = 4 \times 10^{-4}$ and $\epsilon = 10^{-2}$ for Log-Lasso. Finally, $\tau = \|A\|_2^{-2}$. The considered setting guarantees that Lasso and Log-Lasso are successful, that is, by solving them we recover the correct support; see, e.g., Fuchs (2005) for theoretical guarantees. In particular, all the algorithms converge almost to the same solution. Instead, our goal is to analyze the convergence rate, in terms of number of iterations. We specify that each iteration has comparable computational complexity for all the algorithms; therefore the number of iterations well represents the velocity of the algorithm.

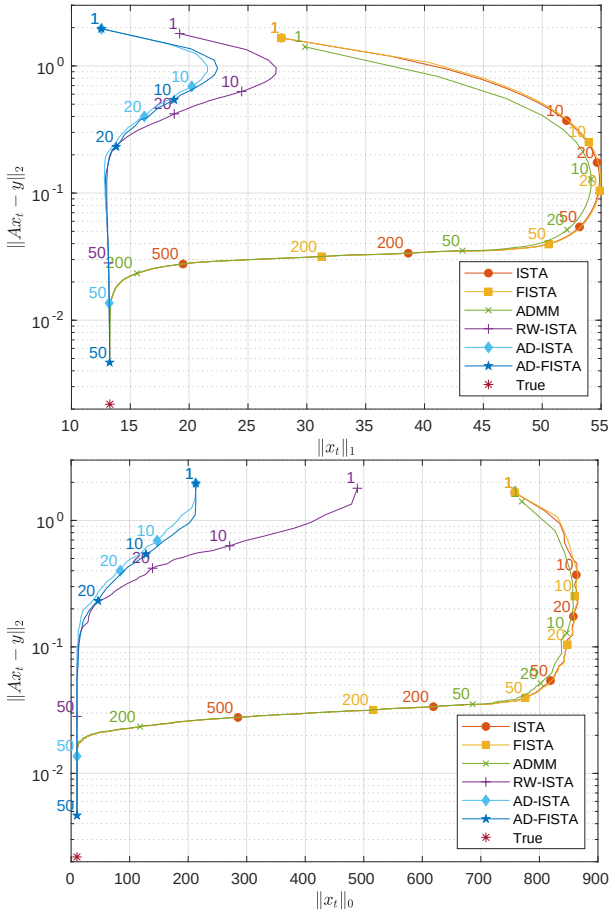

Figure 1. Residual norm $\|Ax_t - y\|_2$ with respect to $\|x_t\|_1$ and $\|x_t\|_0$, respectively. The curve are parametrized with time. We label the iterations 1,10,20,50,200,500 to ease the comparison of the algorithms. "True" refers to the value of $\widetilde{x}$, which is estimated by Lasso with a small bias on the non-zero components.

In Fig. 1, we can see the evolution of $\|Ax_t - y\|_2$ with respect to $\|x_t\|_1$ and $\|x_t\|_0$ in a single experiment. As discussed in previous sections, ISTA, FISTA, and ADMM are characterized by a transient overestimation of $\|x_t\|_1$. As a consequence, a similar behavior is observed for the sparsity $\|x_t\|_0$. In contrast, the proposed AD-ISTA and AD-FISTA keep $\|x_t\|_1$ smaller. As expected AD-FISTA is faster. Also RW-ISTA maintains a low $\|x_t\|_1$, but less effectively than the proposed algorithms.

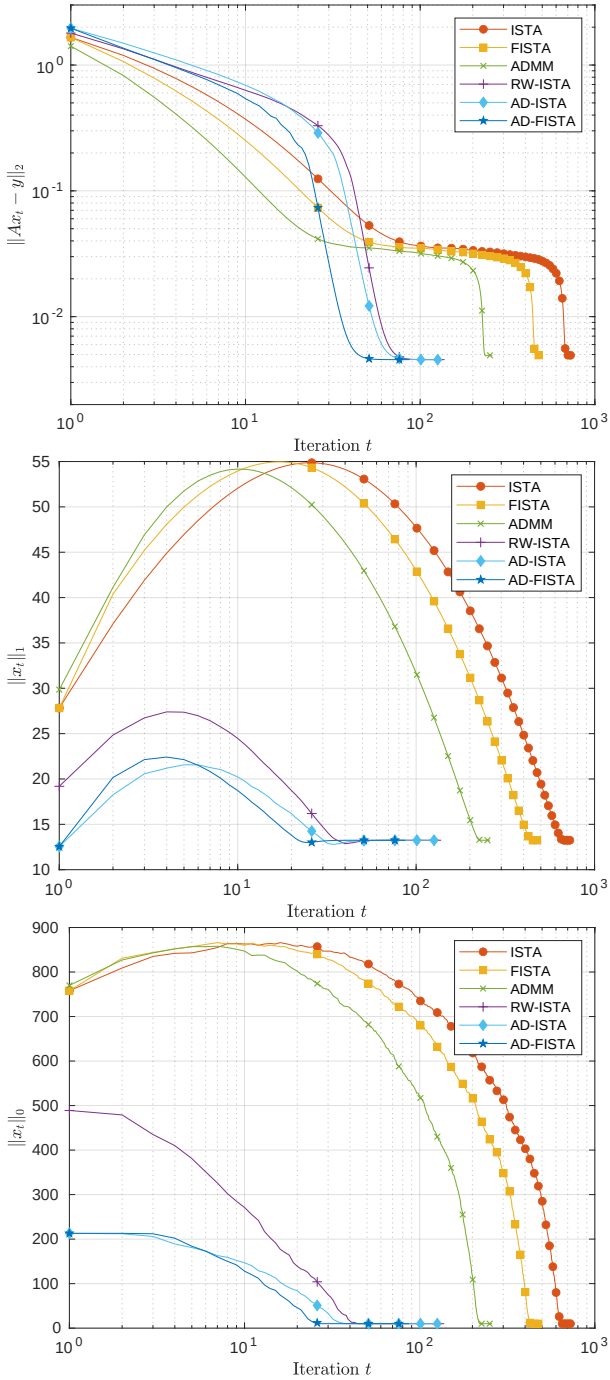

Figure 2. The time evolution of $\|Ax_t - y\|_2$, $\|x_t\|_1$ and $\|x_t\|_0$.

In Fig. 2, we depict the time evolution of $\|Ax_t - y\|_2$, $\|x_t\|_1$ and $\|x_t\|_0$, for the same experiment. We can see that the number of iterations required by AD-ISTA and AD-FISTA is substantially lower than the one of ISTA, FISTA and ADMM.

In Table 1, we collect some statistics on the number of iterations over 100 random runs. We notice that the maximum of iterations required by AD-ISTA and AD-FISTA is smaller than the minimum of iterations required by ISTA, FISTA and ADMM.

	Number	of	iterations
Algorithm	Mean	Min	Max
ISTA	895.44	703	1085
FISTA	595.94	467	722
ADMM	318.49	255	378
RW-ISTA	147.47	126	173
AD-ISTA	138.34	119	162
AD-FISTA	90.64	78	107

Table 1. Number of iterations to converge over 100 random runs.

7. CONCLUSIONS

In this work, we propose and analyze AD-ISTA, a variant of ISTA developed by applying the proximal gradient method to Log-Lasso. AD-ISTA converges in less iterations with respect to ISTA, FISTA and ADMM, thanks to an adaptive shrinkage hyperparameter, that limits the increase of the ℓ_1-norm during the first phase. Moreover, by applying the principles of FISTA, we also propose the accelerated version AD-FISTA. Through numerical experiments, we verify that AD-ISTA is faster than the state-of-the-art algorithms for Lasso and that we obtain a further acceleration with AD-FISTA. Possible extensions of this work include the rigorous proof of the convergence rate and the generalization to sparse optimization problems different from Lasso.

REFERENCES

Attouch, H., Bolte, J., and Svaiter, B.F. (2011). Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. *Mathematical Programming, Series A*, 137(1), 91–124.

Bayram, I. (2016). On the convergence of the iterative shrinkage/thresholding algorithm with a weakly convex penalty. *IEEE Trans. Signal Process.*, 64(6), 1597–1608.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imag. Sci.*, 2(1), 183–202.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2010). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1), 1 – 122.

Bredies, K. and Lorenz, D. (2008). Linear convergence of iterative soft-thresholding. *J. Fourier Anal. Appl.*, 14(5-6), 813–837.

Bredies, K., Lorenz, D.A., and Reiterer, S. (2015). Minimization of non-smooth, non-convex functionals by iterative thresholding. *J. Optim. Theory Appl.*, 165(1), 78–112.

Brunton, S.L., Proctor, J.L., and Kutz, J.N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *PNAS*, 113(15), 3932–3937.

Brunton, S.L. and Kutz, J.N. (2019). *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control.* Cambridge University Press.

Calafiore, G.C. and El Ghaoui, L. (2014). *Optimization Models.* Cambridge University Press.

Candès, E.J., Wakin, M.B., and Boyd, S. (2008). Enhancing sparsity by reweighted ℓ_1 minimization. *J. Fourier Anal. Appl.*, 14(5-6), 877–905.

Chen, X., Liu, J., Wang, Z., and Yin, W. (2021). Hyperparameter tuning is all you need for LISTA. In A. Beygelzimer, Y. Dauphin, P. Liang, and J.W. Vaughan (eds.), *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS).*

Combettes, P.L. and Pesquet, J.C. (2011). *Proximal Splitting Methods in Signal Processing*, 185–212. Springer New York, New York, NY.

Dahlke, S., Fornasier, M., and Raasch, T. (2012). Multilevel preconditioning and adaptive sparse solution of inverse problems. *Math. Comput.*, 81(277), 419–446.

Dahlke, S., Fornasier, M., and Raasch, T. (2009). Multilevel preconditioning for adaptive sparse optimization. *extended preprint.*

Daubechies, I., Defrise, M., and De Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.*, 57(11), 1413 – 1457. doi:10.1002/cpa.20042.

Daubechies, I., Fornasier, M., and Loris, I. (2008). Accelerated projected gradient method for linear inverse problems with sparsity constraints. *J. Fourier Anal. Appl.*, 14, 764–792.

Fornasier, M. (2010). Numerical methods for sparse recovery. In M. Fornasier (ed.), *Theoretical Foundations and Numerical Methods for Sparse Recovery*, 93–200. Radon Series Comp. Appl. Math., de Gruyter.

Fosson, S.M. (2018). A biconvex analysis for lasso ℓ_1 reweighting. *IEEE Signal Process. Lett.*, 25(12), 1795–1799.

Fosson, S.M. (2021). Centralized and distributed online learning for sparse time-varying optimization. *IEEE Trans. Autom. Control*, 66(6), 2542–2557.

Fuchs, J.J. (2005). Recovery of exact sparse representations in the presence of bounded noise. *IEEE Trans. Inf. Theory*, 51(10), 3601–3608.

Gregor, K. and LeCun, Y. (2010). Learning Fast Approximations of Sparse Coding. In *Proc. Int. Conf. Mach. Learn. (ICML)*, 399–406. Omnipress.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations.* CRC press, 2nd edition.

Hong, M. and Luo, Z. (2017). On the linear convergence of the alternating direction method of multipliers. *Math. Progamm.*, 162, 165–199.

Lauer, F. and Bloch, G. (2019). *Hybrid system identification.* Springer.

Liu, J., Chen, X., Wang, Z., and Yin, W. (2019). ALISTA: Analytic weights are as good as learned weights in LISTA. In *Proc. Int. Conf. Learn. Represent. (ICLR).*

Louizos, C., Welling, M., and Kingma, D.P. (2018). Learning sparse neural networks through ℓ_0 regularization. In *Proc. Int. Conf. Learn. Represent. (ICLR).*

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *J. Roy. Stat. Soc. Series B*, 58, 267–288.

Zhao, W., Yin, G., and Bai, E.W. (2020). Sparse system identification for stochastic systems with general observation sequences. *Automatica*, 121, 109162.

Zuo, W., Meng, D., Zhang, L., Feng, X., and Zhang, D. (2013). A generalized iterated shrinkage algorithm for non-convex sparse coding. In *IEEE Intern. Conf. Comput. Vis.*, 217–224.