A ROBUST FISTA-LIKE ALGORITHM

Mihai I. Florea and Sergiy A. Vorobyov

Department of Signal Processing and Acoustics, Aalto University, Espoo, Finland

ABSTRACT

The Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) is regarded as the state-of-the-art among a number of proximal gradient-based methods used for addressing large-scale optimization problems with simple but non-differentiable objective functions. However, the efficiency of FISTA in a wide range of applications is hampered by a simple drawback in the line search scheme. The local estimate of the Lipschitz constant, the inverse of which gives the step size, can only increase while the algorithm is running. As a result, FISTA can slow down significantly if the initial estimate of the Lipschitz constant is excessively large or if the local Lipschitz constant decreases in the vicinity of the optimal point. We propose a new FISTA-like method endowed with a robust step size search procedure and demonstrate its effectiveness by means of a rigorous theoretical convergence analysis and simulations.

Index Terms— FISTA, backtracking, line search, convergence

1. INTRODUCTION

Simple continuous convex optimization problems are used to model many inverse problems and several simple classification tasks, particularly in imaging applications. Often, as in the case of sparse inverse problems, the objective is not differentiable in certain parts of the search space [1]. Accelerated algorithms that rely on gradient information (e.g. [2]) cannot be used directly to solve such problems. However, if the problem objective has a composite structure, certain algorithms are effective when supplied with *proximal gradient* information, instead of gradient information [3]. Although the number of variables can be large, usually of the order of millions [1], with recent advances in graphics processors it is possible to compute the proximal gradient on a single machine without the need for expensive communication between processing nodes. Consequently, proximal gradient methods are increasingly employed for addressing composite problems, and have become the subject of very active research [4–8]. Among proximal gradient methods, the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [9] is currently regarded as the state-of-the-art [1, 10].

FISTA is designed to solve the following problem:

$$\min_{\boldsymbol{x} \in Q} F(\boldsymbol{x}) = f(\boldsymbol{x}) + \Psi(\boldsymbol{x}), \tag{1}$$

where $Q \subseteq \mathbb{R}^n$ is a closed convex set, $\boldsymbol{x}$ is a vector of optimization variables, and F is the objective function, which has a composite structure, i.e., f is a convex differentiable function with Lipschitz continuous gradient (Lipschitz constant L_f) and Ψ is a convex function that may not be differentiable nor defined outside Q. However, Ψ is simple in the sense that the proximal operator

$$\text{prox}_{\tau\Psi}(\boldsymbol{v}) = \arg\min_{\boldsymbol{x} \in Q} \left(\Psi(\boldsymbol{x}) + \frac{1}{2\tau} \|\boldsymbol{x} - \boldsymbol{v}\|_2^2 \right) \tag{2}$$

can be computed for any v in the search space and any non-negative step size τ in $\mathcal{O}(n)$ time. The popularity of FISTA owes to its simplicity and speed, but mostly to its generality [10]. The algorithm is unaware of the nature of the objective function or the problem constraints. It progresses using a sequence of calls to a proximal gradient routine, each involving one call to the proximal operator.

When the Lipschitz constant L_f is not known in advance, FISTA employs a simple backtracking "line search" procedure. However, the effectiveness of FISTA is hampered by a simple drawback in the search scheme, namely the estimate of the Lipschitz constant *can only increase* while the algorithm is running. As the step size is set to be the inverse of the Lipschitz constant estimate, the line search may slow the progress of the algorithm. In two situations, the drawbacks of the search are evident: (i) L_0, the initial estimate of L_f, far exceeds the actual value and (ii) the local curvature of f is large in the vicinity of the initial iterates but decreases around the optimal point.

A number of approaches addressing both aforementioned issues have been proposed in the literature. A method that predates FISTA, which we choose to call Nesterov's Accelerated Multistep Gradient Scheme (Nesterov's AMGS) [11], does feature a step size increase schedule. While having comparable theoretical convergence guarantees to FISTA, it is slower in the most common applications [12]. A variation of Nesterov's AMGS more similar to FISTA, mentioned in [3], is nominally equipped with a step size increase option when "conditions permit". However, the study does not quantitatively describe these conditions nor does it provide any theoretical convergence guarantees. In a recent study [13], FISTA has been equipped with an "exact" line search procedure. This version comes with a rigorous convergence analysis but assumes that the objective function is known to the algorithm, detracting from the generality of FISTA's black-box philosophy.

In this work, we propose an algorithm that alleviates the drawbacks of FISTA in the above mentioned situations, rendering it *robust* in the sense that it can be applied without parameter adjustment to the full spectrum of problems it addresses. Furthermore, our method does not restrict the generality of FISTA and does not alter the theoretical convergence guarantees while surpassing FISTA in practice. We support our findings with simulation results.

2. ROBUST LINE SEARCH FISTA

Our goal is to create an algorithm that can dynamically adjust the Lipschitz constant estimate at every iteration. The simplest and most straightforward way of achieving this is by decreasing the Lipschitz constant estimate slightly at the beginning of every iteration, relying on backtracking to correct excessive reduction. This search strategy is not applicable to FISTA (i.e., convergence cannot be theoretically guaranteed) because this method collects insufficient information while it is running. In FISTA, the first iteration $k = 0$ is a proximal point step [10]. At every subsequent iteration $k \geq 1$,

978-1-5090-4117-6/17/$31.00 ©2017 IEEE

the new iterate $\boldsymbol{x}_{k+1}$ (estimate of optimal point $\boldsymbol{x}^*$) is obtained by querying the proximal gradient not at the previous iterate $\boldsymbol{x}_k$, but at a point $\boldsymbol{y}_{k+1}$ obtained from the two preceding iterates $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k-1}$ through extrapolation, the extent of which depends on terms t_k and t_{k+1} of a recursively defined weight sequence $\{t_i\}_{i\geq 1}$. Although not maintained explicitly while running, FISTA's convergence analysis relies on an auxiliary sequence (which we denote by $\{\boldsymbol{z}_i\}_{i\geq 0}$) that is updated in parallel with $\{\boldsymbol{x}_i\}_{i\geq 0}$. Note that $\boldsymbol{y}_{k+1}$ can be obtained as a convex combination of $\boldsymbol{x}_k$ and $\boldsymbol{z}_k$, with the weighting determined by t_k and t_{k+1}.

FISTA benefits from the same simplification employed by the Fast Gradient Method (FGM) [2]. When the step size is non-increasing, the sequence $\{t_i\}_{i\geq 1}$ can be determined a priori, irrespective of Lipschitz constant estimate values. Maintaining at every iteration k the accumulated weight property

$$T_{k+1} = \sum_{i=1}^{k+1} t_i = t_{k+1}^2, \quad \forall k \geq 0, \tag{3}$$

guarantees an $\mathcal{O}(\frac{1}{k^2})$ rate of convergence, optimal for a class of first-order algorithms introduced in [2]. By relaxing the non-increasing step size assumption, the weight sequence $\{t_i\}_{i\geq 1}$ can be updated to take into account the current and past Lipschitz constant estimates, yielding a more robust algorithm that retains the $\mathcal{O}(\frac{1}{k^2})$ rate of convergence. Specifically, we can replace the accumulated weight property (3) with

$$T_{k+1} = \sum_{i=1}^{k+1} \frac{t_i}{L_i} \geq L_{k+1} t_{k+1}^2, \quad \forall k \geq 0, \tag{4}$$

where L_{k+1} represents the new Lipschitz constant estimate obtained at iteration k. Equality in (4) ensures the fastest theoretical convergence rate but our framework accommodates also methods that violate equality to trade off speed for other desirable properties, such as weak convergence of iterates [7] or ease of interpretation [14].

Using equality in (4), we propose the method outlined in Algorithm 1. Our notation differs slightly from the one used by FISTA [9]. We define the quadratic function $U_{L,y}(\boldsymbol{x})$ as

$$U_{L,y}(\boldsymbol{x}) = f(\boldsymbol{y}) + \langle \bigtriangledown f(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2. \tag{5}$$

Then, by fixing the step size in (2) to be $\tau = 1/L$, the proximal gradient expression becomes

$$p_L(\boldsymbol{y}) = \arg\min_{\boldsymbol{x}\in Q} \left(U_{L,y}(\boldsymbol{x}) + \Psi(\boldsymbol{x})\right) \tag{6}$$

$$= \operatorname{prox}_{\tau\Psi}\left(\boldsymbol{y} - \tau \bigtriangledown f(\boldsymbol{y})\right). \tag{7}$$

While the formulation of our algorithm appears greatly dissimilar to FISTA, in fact, we can obtain an algorithm that is *completely equivalent* to FISTA (which we designate as z-FISTA) by simply removing line 4 and by modifying lines 6 and 19. For z-FISTA, line 6 reads instead as

$$\hat{t} := \frac{1 + \sqrt{1 + 4T_k}}{2} \tag{8}$$

and line 19 is replaced with $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \hat{t}(\hat{\boldsymbol{x}} - \hat{\boldsymbol{y}})$.

The relation of the remainder of parameters in Algorithm 1 to those present in z-FISTA is listed in Table 1. In the proposed method, the accumulated weight T_{k+1} constitutes a valid convergence rate at each iteration k (see Section 3 for proof). The quantity T_{k+1} does not hold the same meaning in z-FISTA, where a valid convergence rate is instead given by $\bar{T}_{k+1} = T_{k+1}/L_{k+1}$. This is poorer than in our method. In FISTA, $\{\boldsymbol{z}_i\}_{i\geq 0}$ and $\{T_i\}_{i\geq 0}$ are abstracted away. Similarly, in the proposed method, there is no need to maintain $\{\boldsymbol{y}_i\}_{i\geq 1}$ nor $\{t_i\}_{i\geq 1}$ across iterations. Instead, we update only current estimates of these sequences.

Algorithm 1 A robust FISTA-like algorithm

1: $\boldsymbol{z}_0 = \boldsymbol{x}_0$
2: $T_0 = 0$
3: **for** k = 0,...,K-1 **do**
4:     $\hat{L} := \gamma_d L_k$
5:     **loop**
6:         $\hat{t} := \frac{1+\sqrt{1+4\hat{L}T_k}}{2\hat{L}}$
7:         $\hat{T} := T_k + \hat{t}$
8:         $\hat{\boldsymbol{y}} := \frac{1}{\hat{T}}(T_k \boldsymbol{x}_k + \hat{t}\boldsymbol{z}_k)$
9:         $\hat{\boldsymbol{x}} := p_{\hat{L}}(\hat{\boldsymbol{y}})$
10:         **if** $f(\hat{\boldsymbol{x}}) \leq U_{\hat{L},\hat{\boldsymbol{y}}}(\hat{\boldsymbol{x}})$ **then**
11:             Break from loop
12:         **else**
13:             $\hat{L} := \gamma_u \hat{L}$
14:         **end if**
15:     **end loop**
16:     $L_{k+1} = \hat{L}$
17:     $\boldsymbol{x}_{k+1} = \hat{\boldsymbol{x}}$
18:     $T_{k+1} = \hat{T}$
19:     $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \hat{t}\hat{L}(\hat{\boldsymbol{x}} - \hat{\boldsymbol{y}})$
20: **end for**

Table 1. Parameters and variables used by our method

Type	Symbol	Domain	Description	z-FISTA equivalent
Input	$\boldsymbol{x}_0$	$\mathbb{R}^n$	initial estimate of $\boldsymbol{x}^*$	same
Input	L_0	$(0,\infty)$	initial estimate of L_f	same
Input	γ_u	$(1,\infty)$	increase rate of $\hat{L}$	same
Input	γ_d	$(0,1)$	decrease rate of $\hat{L}$	none
Internal	$\hat{L}$	$(0,\infty)$	estimate of L_f	same
Internal	$\hat{\boldsymbol{x}}$	Q	estimate of $\boldsymbol{x}_{k+1}$	same
Internal	$\hat{\boldsymbol{y}}$	$\mathbb{R}^n$	estimate of $\boldsymbol{y}_{k+1}$	$\boldsymbol{y}_{k+1}$
Internal	$\hat{t}$	$(0,\infty)$	weight of $\boldsymbol{z}_{k+1}$	$\frac{t_{k+1}}{L_{k+1}}$
Internal	$\hat{T}$	$(0,\infty)$	estimate of T_{k+1}	$\frac{T_{k+1}}{L_{k+1}}$
Output	$\boldsymbol{x}_K$	Q	final estimate of $\boldsymbol{x}^*$	same

Our method cannot be benchmarked directly against FISTA and Nesterov's AMGS in terms of theoretical computational complexity. Each method calls a dynamic mix of functions which, depending on the problem specification, may have vastly varying relative complexities. Table 2 provides a detailed description of the type and number of function calls in several stages of an iteration. Our method requires more computation than FISTA for a backtracking operation while showing no increase in complexity when no backtracks occur. Nesterov's AMGS cannot be compared in any algorithmic state as it was designed to work without the need to implement function value calls to f. An iteration of Nesterov's AMGS does, however, require at least two projection calls (proximal operator plus gradient computation) which gives it a clear disadvantage when these operations are more complex than calls to f. Overall, our method strikes a balance in a variety of situations, further contributing to its robustness.

Table 2. Per iteration complexity, measured in terms of operator calls, of FISTA, Nesterov's AMGS, and our method

	FISTA			Nesterov's AMGS			Our method		
	f	∇f	$\text{prox}_{\tau\Psi}$	f	∇f	$\text{prox}_{\tau\Psi}$	f	∇f	$\text{prox}_{\tau\Psi}$
Step size validation (lines 4 to 11)	2	1	1	0	2	1	2	1	1
Backtrack (lines 4 to 15)	1	0	1	0	2	1	2	1	1
State update (lines 16 to 19)	0	0	0	0	0	1	0	0	0
Iteration without backtrack (lines 4 to 19)	2	1	1	0	2	2	2	1	1

3. CONVERGENCE ANALYSIS

While our method does not keep track of the sequences $\{\boldsymbol{y}_k\}_{k\geq 1}$ and $\{t_k\}_{k\geq 1}$, they can be easily recovered. Lines 6, 7, 8 and 18 in Algorithm 1 imply that

$$t_{k+1} = T_{k+1} - T_k, \quad \forall k \geq 0, \tag{9}$$
$$T_{k+1}\boldsymbol{y}_{k+1} = T_k\boldsymbol{x}_k + t_{k+1}\boldsymbol{z}_k, \quad \forall k \geq 0. \tag{10}$$

Given that for every $k \geq 1$, $\boldsymbol{x}_k$ satisfies relations

$$f(\boldsymbol{x}_k) \leq U_{L_k,\boldsymbol{y}_k}(\boldsymbol{x}_k), \tag{11}$$
$$\boldsymbol{x}_k = p_{L_k}(\boldsymbol{y}_k), \tag{12}$$

enforced by lines 9, 10, 16 and 17 in Algorithm 1, it follows that [9, Lemma 2.3] holds for all $k \geq 1$ and $\boldsymbol{x} \in \mathbb{R}^n$, that is

$$F(\boldsymbol{x}) - F(\boldsymbol{x}_k) \geq \frac{L_k}{2}\|\boldsymbol{x}_k - \boldsymbol{y}_k\|_2^2 + L_k\langle \boldsymbol{y}_k - \boldsymbol{x}, \boldsymbol{x}_k - \boldsymbol{y}_k\rangle. \tag{13}$$

Let us consider the sequence $\{\Delta_k\}_{k\geq 0}$, defined as

$$\Delta_k = T_k(F(\boldsymbol{x}_k) - F^*) + \frac{1}{2}\|\boldsymbol{z}_k - \boldsymbol{x}^*\|_2^2, \quad \forall k \geq 0. \tag{14}$$

We aim to prove that this sequence is non-increasing. Indeed, applying (13) at iteration $k+1$ (for all $k \geq 0$) using $\boldsymbol{x}_k$ and $\boldsymbol{x}^*$ as values of $\boldsymbol{x}$ we obtain

$$F(\boldsymbol{x}_k) - F(\boldsymbol{x}_{k+1}) \geq \frac{L_{k+1}}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{y}_{k+1}\|_2^2 + L_{k+1}\langle \boldsymbol{y}_{k+1} - \boldsymbol{x}_k, \boldsymbol{x}_{k+1} - \boldsymbol{y}_{k+1}\rangle, \tag{15}$$

$$F(\boldsymbol{x}^*) - F(\boldsymbol{x}_{k+1}) \geq \frac{L_{k+1}}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{y}_{k+1}\|_2^2 + L_{k+1}\langle \boldsymbol{y}_{k+1} - \boldsymbol{x}^*, \boldsymbol{x}_{k+1} - \boldsymbol{y}_{k+1}\rangle. \tag{16}$$

Lines 6, 7, 16 and 18 ensure that (4) holds with equality. Using (9), (10), and (4) in $T_k \cdot (15) + t_{k+1} \cdot (16)$ we obtain

$$T_k(F(\boldsymbol{x}_k) - F^*) - T_{k+1}(F(\boldsymbol{x}_{k+1}) - F^*) \geq L_{k+1}^2 t_{k+1}^2\|\boldsymbol{x}_{k+1} - \boldsymbol{y}_{k+1}\|_2^2 + t_{k+1}L_{k+1}\langle \boldsymbol{z}_k - \boldsymbol{x}^*, \boldsymbol{x}_{k+1} - \boldsymbol{y}_{k+1}\rangle. \tag{17}$$

Lines 16 and 19 translate into the following recursion rule:

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + t_{k+1}L_{k+1}(\boldsymbol{x}_{k+1} - \boldsymbol{y}_{k+1}), \quad \forall k \geq 0. \tag{18}$$

Then, using (18) in (17) and rearranging terms, we obtain the desired result,

$$\Delta_{k+1} \leq \Delta_k, \quad \forall k \geq 0. \tag{19}$$

This last inequality implies that every term Δ_k, $k \geq 1$, is upper bounded by Δ_0. Given that $\Delta_0 = \frac{1}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2^2$ and that the quantity $\frac{1}{2}\|\boldsymbol{z}_k - \boldsymbol{x}^*\|_2^2$ is always non-negative, we can write down a convergence rate explicitly as

$$F(\boldsymbol{x}_k) - F^* \leq \frac{1}{2T_k}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2^2, \quad \forall k \geq 1. \tag{20}$$

Clearly, T_k constitutes a valid convergence rate. To obtain a simple closed form convergence rate, it suffices to find a simple lower bound for T_k. When $L_k \geq L_f$, due to the Lipschitz continuous property of ∇f, inequality (11) holds regardless of the values of $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$. Hence, the backtracking search will never produce a value of L_k beyond $\gamma_u L_f$. Therefore, combining (4) with $L_k \leq \gamma_u L_f$ we obtain

$$T_{k+1} \geq T_k + \frac{1}{2\gamma_u L_f} + \sqrt{\frac{1}{4(\gamma_u L_f)^2} + \frac{T_k}{\gamma_u L_f}}, \quad \forall k \geq 0. \tag{21}$$

Using (21) and $T_0 = 0$, it follows through induction that

$$T_k \geq \frac{(k+1)^2}{4\gamma_u L_f}, \quad \forall k \geq 1. \tag{22}$$

Finally, substituting the lower bound on T_k (22) in (20) we obtain the same quadratic convergence rate as the original FISTA [9, Theorem 4.4], namely

$$F(\boldsymbol{x}_k) - F^* \leq \frac{2\gamma_u L_f}{(k+1)^2}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|_2^2, \quad \forall k \geq 1. \tag{23}$$

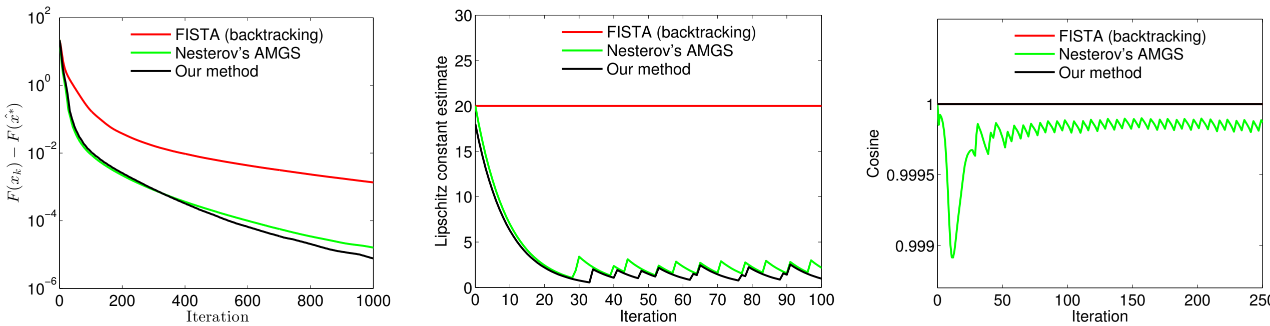
Note that our convergence analysis is more general than the one provided in [9]. Inequality (19) applies to FISTA as well (by replacing T_k with $\bar{T}_k = t_k^2/L_k$ for $k \geq 1$ and setting $T_0 = 0$) and can be used to obtain the same convergence rate for variations on the weight update rule (8).

4. NUMERICAL ANALYSIS

The performance of our method (Algorithm 1) was tested and compared to that of FISTA with backtracking line search and Nesterov's AMGS on the l_1 regularized deblurring of a simple test image. For ease of benchmarking, we used the experimental setup from [9, Section 5.1]. The composite objective function is given by

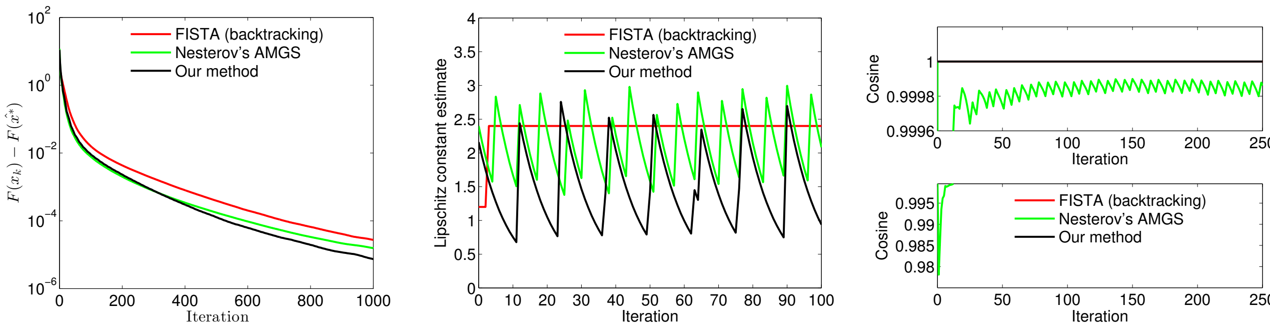
$$f(\boldsymbol{x}) = \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2, \qquad \Psi(\boldsymbol{x}) = \lambda\|\boldsymbol{x}\|_1, \tag{24}$$

where $\boldsymbol{A} = \boldsymbol{R}\boldsymbol{W}$; $\boldsymbol{R}$ is a matrix representing Gaussian blur (9×9 pixel kernel, standard deviation 4.0, reflexive boundary conditions [15]); $\boldsymbol{W}$ is the inverse three-stage Haar wavelet transform; $\boldsymbol{b}$ is obtained by applying $\boldsymbol{R}$ to the 256×256 cameraman test image (pixel values scaled to the $[0, 1]$ range), followed by the addition of Gaussian noise (zero-mean, standard deviation 10^{-3}). Here, ∇f has a Lipschitz constant value $L_f = 2.0$, computed as the maximum eigenvalue of a symmetric Toeplitz-plus-Hankel matrix, according to [15], and $\lambda = 2 \cdot 10^{-5}$ is a regularization parameter. In addition, we chose $\gamma_u = 2.0$ and $\gamma_d = 0.9$ for each method tested, as these values were suggested in [3] to "provide good performance in many applications". Two scenarios are considered: a pathologically overestimated initial guess $L_0 = 10L_f$ (Fig. 1) and a normally underestimated $L_0 = 0.3L_f$ (Fig. 2).



(a) Convergence rate (b) Lipschitz constant estimate (c) Inertial degree

Fig. 1. Comparison of FISTA, Nesterov's AMGS, and our method for an overestimated initial Lipschitz constant: $L_0 = 10L_f$



(a) Convergence rate (b) Lipschitz constant estimate (c) Inertial degree

Fig. 2. Comparison of FISTA, Nesterov's AMGS, and our method for an underestimated initial Lipschitz constant: $L_0 = 0.3L_f$

Convergence is measured in terms of the difference between objective function values and an optimal value estimate, during the first 1000 iterations (Figs. 1(a) and 2(a)). This estimate was computed as $F(\hat{\boldsymbol{x}^*})$, where $\hat{\boldsymbol{x}^*}$ is the iterate obtained after running fixed step size FISTA with the correct Lipschitz constant parameter for 10000 iterations. Key algorithm state parameters, such as Lipschitz constant estimates (Figs. 1(b) and 2(b)) and inertial degrees (Figs. 1(c) and 2(c)) are shown only during the first 100 iterations, as subsequent iterations did not reveal more information. Inertial degrees are defined at every iteration $k \geq 0$ as the cosine of the angle between $\boldsymbol{y}_{k+1}$ and $\boldsymbol{x}_k$ at $\boldsymbol{x}_{k-1}$. When two of these points match, the inertial degree is set to 1.

In both scenarios, after the first 400 iterations, our method clearly surpasses the others in terms of function value (Figs. 1(a) and 2(a)). FISTA converges slowly, especially in the pathological case where, as expected, FISTA is unable to reduce its Lipschitz constant estimate (Fig. 1(b)), whereas the other methods are able to decrease their estimates at comparable rates during the first 30 iterations. Under normal conditions (Fig. 2(b)), FISTA quickly increases its estimate in the first iterations after which the value reaches a saturation level. The other methods are constantly adjusting their estimates. In both situations, our method produces on average a lower L_k than the other two methods. FISTA's inability to reduce L_k accounts for its high estimates whereas Nesterov's AMGS has a stricter backtracking condition than our method or FISTA, leading to more backtracks.

In our method, just as in FISTA, $\boldsymbol{y}_{k+1}$, $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k-1}$ are collinear (Figs. 1(c) and 2(c)). However, in Nesterov's AMGS they are not, contradicting the notion found in several monographs in the field (e.g. [10,16]) that all accelerated first order methods rely on extrapolation. We provide in [12] a rigorous proof of collinearity in the proposed method and corroborate the superiority of our algorithm with a more detailed performance analysis.

In summary, our method can be regarded as a hybrid of FISTA with its collinear iterates (Figs. 1(c) and 2(c)) and Nesterov's AMGS with its dynamic step search procedure (Figs. 1(b) and 2(b)), benefiting from the strengths of these methods while alleviating the drawbacks. Namely, our method produces more accurate estimates of the local curvature of f (unlike the artificially high estimates of FISTA) and is able to utilize both gradient and subgradient information (whereas Nesterov's AMGS updates a weighted average of gradients without taking into consideration the subgradient of Ψ), resulting in larger steps and, consequently, faster convergence.

5. CONCLUSION

By updating the weight sequence to take into account the current and past Lipschitz constant estimates, we have devised a FISTA-like algorithm with a robust step size search strategy. We have shown that the same theoretical convergence rate of $\mathcal{O}(\frac{1}{k^2})$ applies to our method, with a provably smaller constant. Simulation results on the very problem FISTA was introduced to solve show that our method surpasses both FISTA and the more complex Nesterov's AMGS, without the need to adjust any parameters.

The properties of the proposed method follow naturally from the augmented estimate sequence framework [12]. In fact, our method is a particular case of the Accelerated Composite Gradient Method, a general-purpose optimization scheme [12]. Thus, the concepts presented in this work are of importance to the entire field of accelerated optimization algorithms.

6. REFERENCES

[1] A. Chambolle and T. Pock, "An introduction to continuous optimization for imaging," *Acta Numer.*, vol. 25, pp. 161–319, May 2016.

[2] Y. Nesterov, *Introductory lectures on convex optimization. Applied optimization, vol. 87*. Kluwer Academic Publishers, Boston, 2004.

[3] S. R. Becker, E. J. Candès, and M. C. Grant, "Templates for convex cone problems with applications to sparse signal recovery," *Math. Program. Comput.*, vol. 3, no. 3, pp. 165–218, Sep. 2011.

[4] A. Auslender and M. Teboulle, "Interior gradient and proximal methods for convex and conic optimization," *SIAM J. Optim.*, vol. 16, no. 3, pp. 697–725, 2006.

[5] G. Lan, Z. Lu, and R. D. Monteiro, "Primal-dual first-order methods with $\mathcal{O}(1/\epsilon)$ iteration-complexity for cone programming," *Math. Program.*, vol. 126, no. 1, pp. 1–29, 2011.

[6] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," May 2008, submitted to SIAM J. Optim. Available: http://www.mit.edu/~dimitrib/PTseng/papers/apgm.pdf

[7] A. Chambolle and C. Dossal, "On the convergence of the iterates of the fast iterative shrinkage/thresholding algorithm," *J. Optim. Theory Appl.*, vol. 166, no. 3, pp. 968–982, May 2015.

[8] B. O'Donoghue and E. Candès, "Adaptive restart for accelerated gradient schemes," *Found. Comput. Math.*, vol. 15, no. 3, pp. 715–732, 2015.

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

[10] N. Parikh, S. P. Boyd, et al., "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.

[11] Y. Nesterov, "Gradient methods for minimizing composite objective function," CORE, Université Catholique de Louvain, Tech. Rep. 76, Sep. 2007.

[12] M. I. Florea and S. A. Vorobyov, "An accelerated composite gradient method for large-scale composite objective problems," *arXiv preprint* arXiv:1612.02352 [math.OC], Dec. 2016.

[13] Z. Zhang and V. Saligrama, "RAPID: Rapidly accelerated proximal gradient algorithms for convex minimization," *Proc. IEEE Intern. Conf. Acoust., Speech and Signal Process. (ICASSP)*, Brisbane, Australia, Apr. 2015, pp. 3796–3800.

[14] W. Su, S. Boyd, and E. Candes, "A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights," *Advances in Neural Inform. Process. Syst. 27*, Montréal, Canada, Dec. 2014, pp. 2510–2518.

[15] P. Hansen, J. Nagy, and D. O'Leary, *Deblurring images*. SIAM, Philadelphia, 2006.

[16] D. P. Bertsekas, *Convex optimization algorithms*. Athena Scientific, Belmont, 2015.