OPT2020: 12th Annual Workshop on Optimization for Machine Learning

A FISTA-type average curvature accelerated composite gradient method for nonconvex optimization problems

Jiaming Liang JIAMING.LIANG@GATECH.EDU
Renato D.C. Monteiro RENATO.MONTEIRO@ISYE.GATECH.EDU
School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0205

Abstract

This paper presents an accelerated composite gradient (ACG) variant of the FISTA type, referred to as AC-FISTA, for solving nonconvex smooth composite minimization problems. As opposed to well-known ACG variants that are either based on a known Lipschitz gradient constant or a line search procedure, AC-FISTA uses the average of all observed functional curvatures and never backtracks. This paper also provides the convergence rate result of AC-FISTA in terms of the aforementioned average curvatures. Finally, computational results are presented to illustrate the efficiency of AC-FISTA on real-world problem instances.

Keywords: smooth nonconvex optimization, average curvature, line search free method.

1. Introduction

In this paper, we study an ACG-type algorithm for solving a nonconvex smooth composite optimization (N-SCO) problem

$$\phi_* := \min \{\phi(z) := f(z) + h(z) : z \in \mathbb{R}^n\} \tag{1}$$

where $h : \mathbb{R}^n \to (-\infty, \infty]$ is a proper lower semicontinuous convex function and f is a real-valued differentiable (possibly nonconvex) function with an M-Lipschitz continuous gradient on a compact convex set containing $\operatorname{dom} h$.

A large class of methods for solving (1) computes the next iterate y_{k+1} by solving a linearized prox subproblem of the form

$$y_{k+1} = y(\tilde{x}_k; M_k) := \operatorname{argmin} \left\{ l_f(x; \tilde{x}_k) + h(x) + \frac{M_k}{2}\|x - \tilde{x}_k\|^2 : x \in \mathbb{R}^n \right\} \tag{2}$$

where $\tilde{x}_k$ is chosen as either the current iterate y_k (as in unaccelerated algorithms) or a convex combination of y_k and another auxiliary iterate x_k (as in accelerated algorithms), and M_k is an upper estimation of the "local function curvature" of f at $\tilde{x}_k$. More specifically, letting

$$\mathcal{C}(y; \tilde{x}) := \frac{2\,[f(y) - \ell_f(y; \tilde{x})]}{\|y - \tilde{x}\|^2}, \tag{3}$$

M_k is chosen so as to satisfy

$$\mathcal{C}_k := \mathcal{C}(y_{k+1}; \tilde{x}_k) \le M_k. \tag{4}$$

It is well-known that the smaller the sequence $\{M_k\}$ is, the faster the convergence rate of the method becomes. Hence, it is desirable to choose M_k to be the smallest value satisfying (4). Different schemes for choosing M_k have been employed in pure ACG variants for solving (1). The

AG method proposed in [2] is a direct extension of the ACG variant based on the constant choice of M_k (i.e., $M_k = M$ where M is the Lipschitz constant of ∇f) to the N-SCO context. AG performs two resolvent evaluations of ∂h per iteration. NC-FISTA of [6] is an extension of the well-known ACG variant FISTA [1] to the N-SCO context. It requires as input a pair (M, m) and sets $M_k = M + \kappa_0 m/(Ma_k) > M$ where κ_0 is a positive universal constant and $\{a_k\}$ is a sequence constructed in the method. In contrast to an iteration of the AG method, every iteration of NC-FISTA performs exactly one resolvent evaluation. One drawback of NC-FISTA is that it requires as input (M, m), which is usually hard to obtain or is often poorly estimated. On the other hand, ADAP-NC-FISTA of [6] remedies this drawback in that it only requires as input an arbitrary initial pair (M_0, m_0), and dynamically updates (M_k, m_k) by means of backtracking procedures.

Paper [5] proposes a new ACG variant, namely the AC-ACG method, for solving the N-SCO problem where M_k is computed as a positive multiple of the average of all observed curvatures up to the previous iteration, and presents numerical results of AC-ACG demonstrating its outperforming practical performance. As opposed to ACG variants based on the schemes outlined in the previous paragraph as well as other ACG variants, AC-ACG always computes a new step regardless of whether M_k overestimates or underestimates $\mathcal{C}_k$. The main result of [5] shows that AC-ACG obtains a pair (y, v) satisfying $v \in \nabla f(y) + \partial h(y)$ and $\|v\| = \mathcal{O}(\sqrt{M_k}/\sqrt{k})$. Since M_k is usually much smaller than $\bar{M}$, which is the smallest Lipschitz constant of ∇f on $\operatorname{dom} h$, this convergence rate bound explains the good empirical performance of AC-ACG.

This paper proposes a variant of AC-ACG, namely AC-FISTA, for solving (1). Like AC-ACG, AC-FISTA is an ACG variant based on the average of all observed curvatures. More specifically, while AC-ACG computes the average C_k^{avg} of the observed curvatures $\tilde{C}_k$ defined as

$$\tilde{C}_k = \max\left\{\mathcal{C}(y(\tilde{x}_k; M_k); \tilde{x}_k), \frac{\|\nabla f(y(\tilde{x}_k; M_k)) - \nabla f(\tilde{x}_k)\|}{\|y(\tilde{x}_k; M_k) - \tilde{x}_k\|}\right\} \tag{5}$$

where $y(\tilde{x}_k; M_k)$ is as in (2), AC-FISTA computes the average C_k^{avg} of the observed curvature $C_k = \mathcal{C}(y(\tilde{x}_k; M_k); \tilde{x}_k)$. Since C_k is smaller than $\tilde{C}_k$, the resulting M_k in AC-FISTA is smaller than that of AC-ACG, and hence AC-FISTA has a faster convergence in practice. In contrast to AC-ACG, which performs two resolvent evaluations of ∂h per iteration, AC-FISTA performs only one resolvent evaluation in those iterations for which (4) holds, and two resolvent evaluations in the other ones. Moreover, following the approach of [5] which analyzes the convergence rate of AC-ACG, this paper also establishes the convergence rate of AC-FISTA. Although the use of C_k^{avg} has already been observed in [5] to yield a quite efficient variant of AC-ACG, the convergence rate analysis of the latter was left as an open problem there. In this regards, this paper presents a FISTA-type ACG variant, related to but different from the one considered in [5], based on C_k^{avg} with a provable convergence rate bound.

Organization of the paper. Section 2 describes the N-SCO problem and the assumptions made on it. It also presents AC-FISTA for solving the N-SCO problem and describes the main result of the paper, which establishes a convergence rate bound for AC-FISTA in terms of two averages of observed curvatures. Section 3 presents computational results illustrating the efficiency of AC-FISTA. Section 4 presents some concluding remarks. Finally, Appendix A provides the detailed experimental setup.

Basic definitions and notation. The set of real numbers is denoted by $\mathbb{R}$. Let $\mathbb{R}^n$ denote the standard n-dimensional Euclidean space with inner product and norm denoted by $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$, respectively. The Frobenius norm in $\mathbb{R}^{m \times n}$ is denoted by $\| \cdot \|_F$. The indicator function I_S of a

set $S \subset \mathbb{R}^n$ is defined as $I_S(z) = 0$ for every $z \in S$, and $I_S(z) = \infty$, otherwise. Let $\psi : \mathbb{R}^n \to (-\infty, +\infty]$ be given. The effective domain of ψ is denoted by $\operatorname{dom}\psi := \{x \in \mathbb{R}^n : \psi(x) < \infty\}$ and ψ is proper if $\operatorname{dom}\psi \neq \emptyset$. If ψ is differentiable at $\bar{z} \in \mathbb{R}^n$, then its affine approximation $\ell_\psi(\cdot; \bar{z})$ at $\bar{z}$ is defined as $\ell_\psi(z; \bar{z}) := \psi(\bar{z}) + \langle \nabla\psi(\bar{z}), z - \bar{z}\rangle$ for every $z \in \mathbb{R}^n$. The subdifferential of ψ at $z \in \mathbb{R}^n$ is denoted by $\partial\psi(z)$. The set of all proper lower semi-continuous convex functions $\psi : \mathbb{R}^n \to (-\infty, +\infty]$ is denoted by $\overline{\text{Conv}}\,(\mathbb{R}^n)$.

2. AC-FISTA and the main result

This section presents the main algorithm studied in this paper, namely, AC-FISTA, describes the N-SCO problem and the assumptions made on it, and states the main result of the paper, i.e. the convergence rate of AC-FISTA.

The problem of interest in this paper is the N-SCO problem (1), where the following conditions are assumed to hold:

(A1) $h \in \overline{\text{Conv}}\,(\mathbb{R}^n)$;

(A2) f is a nonconvex differentiable function on Δ ($\supset \operatorname{dom} h$) and there exist scalars $m \geq 0$, $M \geq 0$ such that for every $u, u' \in \Delta$,

$$-\frac{m}{2}\|u - u'\|^2 \leq f(u) - \ell_f(u; u'), \qquad \|\nabla f(u) - \nabla f(u')\| \leq M\|u - u'\|; \tag{6}$$

(A3) the diameters $D_h := \sup\{\|u - u'\| : u, u' \in \operatorname{dom} h\}$ and $D_\Delta := \sup\{\|u - u'\| : u, u' \in \Delta\}$ are finite.

Throughout the paper, we let $\bar{m}$ (resp., $\bar{M}$) denote the smallest scalar $m \geq 0$ (resp., $M \geq 0$) satisfying the first (resp., second) inequality in (6).

A necessary condition for $\hat{y}$ to be a local minimum of (1) is that $0 \in \nabla f(\hat{y}) + \partial h(\hat{y})$, i.e, $\hat{y}$ be a stationary point of (1). Hence, we have the following definition of a $\hat{\rho}$-approximate stationary point.

Definition 1 *Given a tolerance $\hat{\rho} > 0$, a pair $(\hat{y}, \hat{v}) \in \mathbb{R}^n \times \mathbb{R}^n$ is called a $\hat{\rho}$-approximate stationary point of (1), if $\hat{v} \in \nabla f(\hat{y}) + \partial h(\hat{y})$ and $\|\hat{v}\| \leq \hat{\rho}$.*

We are now ready to state AC-FISTA.

AC-FISTA

---

0. Let a parameter $\gamma \in (0, 1)$, a scalar $M \geq \bar{M}$, a tolerance $\hat{\rho} > 0$ and an initial point $y_0 \in \operatorname{dom} h$ be given and set $A_0 = 0$, $x_0 = y_0$, $M_0 = \gamma M$, $k = 0$ and $\alpha = \frac{0.9}{8}\left(1 + \frac{1}{0.9\gamma}\right)^{-1}$;

1. compute $a_k = (1 + \sqrt{1 + 4M_k A_k})/2M_k$, $\quad A_{k+1} = A_k + a_k$, $\quad \tilde{x}_k = (A_k y_k + a_k x_k)/A_{k+1}$;

2. set $y^g_{k+1} = y(\tilde{x}_k; M_k)$ where $y(\cdot;\cdot)$ is as in (2) and compute

$$\begin{aligned} x_{k+1} &= P_\Delta\left(a_k M_k y^g_{k+1} - \frac{A_k}{a_k} y_k\right), \\ v_{k+1} &= M_k(\tilde{x}_k - y^g_{k+1}) + \nabla f(y^g_{k+1}) - \nabla f(\tilde{x}_k); \end{aligned} \tag{7}$$

3. **if** $\|v_{k+1}\| \le \hat{\rho}$ **then** output $(\hat{y}, \hat{v}) = (y^g_{k+1}, v_{k+1})$ and **stop**; **else**, compute

$$C_k = \mathcal{C}(y^g_{k+1}; \tilde{x}_k), \quad C^{avg}_k = \frac{1}{k+1}\sum_{j=0}^{k} C_j, \quad M_{k+1} = \max\left\{\frac{1}{\alpha}C^{avg}_k, \gamma M\right\} \tag{8}$$

where $\mathcal{C}(\cdot;\cdot)$ is as in (3);

4. set

$$y_{k+1} = \begin{cases} y^b_{k+1} := \frac{A_k y_k + a_k x^b_{k+1}}{A_{k+1}}, & \text{if } C_k > 0.9M_k; \\ y^g_{k+1}, & \text{otherwise} \end{cases} \tag{9}$$

where

$$x^b_{k+1} = \operatorname*{argmin}_{u\in\mathbb{R}^n}\left\{a_k[\ell_f(u;\tilde{x}_k) + h(u)] + \frac{1}{2}\|u - x_k\|^2\right\}, \tag{10}$$

and $k \leftarrow k+1$, and go to step 1.

We add a few observations about AC-FISTA. First, steps 0-2 are in the format of FISTA, which performs only one resolvent evaluation of ∂h for solving y_{k+1}. Second, in the iterations for which $C_k > 0.9M_k$ (called the bad iterations, see (9)), an extra resolvent evaluation of ∂h is required to compute x^b_{k+1} in (10). Third, if Δ is properly chosen, then the projection onto Δ in (7) is usually considerably cheaper than a resolvent evaluation of ∂h. Fourth, in view of step 3, AC-FISTA terminates when a $\hat{\rho}$-approximate stationary point of (1) as in Definition 1 is obtained.

We briefly compare AC-FISTA with AC-ACG presented in [5]. First, while AC-ACG computes C^{avg}_k as the average of $\{\tilde{C}_i : i = 0, \ldots, k\}$ where $\tilde{C}_i$ is as in (5), AC-FISTA computes this quantity as being the average of $\{C_i : i = 0, \ldots, k\}$ where C_i is as in (8). Since C_k is substantially smaller than $\tilde{C}_k$, the resulting C^{avg}_k, and hence M_{k+1}, computed by AC-FISTA is also substantially smaller than those computed by AC-ACG. Second, while AC-ACG performs two resolvent evaluations per iteration, AC-FISTA performs one resolvent evaluation in a good iteration (i.e., an iterations for which $C_k \le 0.9M_k$) and two resolvent evaluations in a bad iteration. Since one of the key results behind the analysis of AC-FISTA is that the number of bad iterations up to iteration k is bounded by $k/3$, it can be shown that the average iteration cost (in terms of the number of resolvent evaluations of ∂h) of AC-FISTA is no more than $2/3$ times the average iteration cost of AC-ACG (although in practice it is frequently observed to be close to $1/2$).

We now state the main result of the paper which describes how fast one of the iterates $y^g_1, \ldots, y^g_k$ approaches the stationary condition $0 \in \nabla f(y) + \partial h(y)$.

Theorem 2 *The following statements hold:*

(a) for every $k \ge 1$, we have $v_k \in \nabla f(y^g_k) + \partial h(y^g_k)$;

(b) for every $k \ge 12$, we have

$$\min_{1\le i\le k}\|v_i\| = \mathcal{O}\left(\left(M_k + \bar{C}^{avg}_k\right)\left(\frac{d_0}{k^{3/2}} + \left(\frac{\sqrt{1-\gamma}}{\sqrt{\gamma}} + \frac{\sqrt{\bar{m}\theta_k}}{\sqrt{M_k}}\right)\frac{D_\Delta}{k} + \frac{\sqrt{\bar{m}\theta_k}D_h}{\sqrt{M_k k}}\right)\right)$$

where $\theta_k := \max\{M_k/M_i : 0 \le i \le k\}$ and

$$\bar{C}^{avg}_k := \frac{1}{k}\sum_{i=0}^{k-1}\bar{C}_i, \quad \bar{C}_k := \frac{\|\nabla f(y^g_{k+1}) - \nabla f(\tilde{x}_k)\|}{\|y^g_{k+1} - \tilde{x}_k\|}.$$

3. Numerical experiments

This section presents computational results to illustrate the performance of AC-FISTA on a constrained version of the nonconvex low-rank matrix completion (NLRMC) problem.

We compare AC-FISTA with six other nonconvex optimization methods, namely: (i) AG proposed in [2]; (ii) NM-APG of [4]; (iii) UPFAG proposed in [3]; (iv) NC-FISTA of [6]; (v) ADAP-NC-FISTA also described in [6]; and (vi) AC-ACG introduced in [5].

The constrained version of the NLRMC problem considered in this section is

$$\min_{Z\in\mathbb{R}^{l\times n}} \left\{ \frac{1}{2}\|\Pi_\Omega(Z-O)\|_F^2 + \mu\sum_{i=1}^{r} p(\sigma_i(Z)) : Z\in\mathcal{B}_R \right\}. \tag{11}$$

The detailed description of the problem and the setup of numerical experiments can be found in Appendix A.

M	Function Value / Iteration Count					Running Time (s)				
	AG	NM	UP	NC/AD	AC/ACF	AG	NM	UP	NC/AD	AC/ACF
4.4	2257 3856	**1809** 1036	2605 521	2628/2625 4780/1674	2288/1816 765/381	4568	1033	1545	3925/1946	923/**440**
8.9	3886 9158	3359 1617	4261 576	4246/4203 9751/1794	3884/**3346** 968/462	10251	1605	1621	7901/1930	1173/**539**
20	4282 22902	3635 2875	4637 676	4641/4582 22259/2209	4267/**3513** 1079/835	29274	2836	1914	15912/2364	1236/**1091**
30	5967 37032	5237 3717	6753 606	6380/6293 32223/1963	5975/**5140** 1085/798	41673	4182	1628	22265/2104	1263/**1068**

Table 1: Numerical results for AG, NM, UP, NC, AD, AC and ACF

Numerical results of the seven methods in four test cases with different M values are given in Table 1, where NM, UP. NC, AD, AC and ACF are short names for NM-APG, UPFAG, NC-FIST, ADAP-NC-FISTA, AC-ACG and AC-FISTA, respectively. In summary, computational results demonstrate that: i) AC-FISTA has the best performance in terms of running time in all four test cases; and ii) AC-FISTA finds the smallest objective function values in the last three test cases, and also finds the second smallest objective function value in the first test case, which is close enough to the best one obtained by NM.

4. Concluding remarks

This paper proposes a FISTA-type ACG variant, namely AC-FISTA, for solving the N-SCO problem (1) which uses the average C_{k-1}^{avg} of all observed curvatures $C_0,\ldots,C_{k-1}$ to compute the next iterate y_{k+1}. Its convergence rate is established in terms of two average curvatures. Numerical results are also presented to illustrate the efficiency of AC-FISTA.

References

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[2] S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Math. Programming*, 156:59–99, 2016. ISSN 1436-4646.

[3] S. Ghadimi, G. Lan, and H. Zhang. Generalized uniformly optimal methods for nonlinear programming. *Journal of Scientific Computing*, 79(3):1854–1881, 2019.

[4] H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. *Adv. Neural Inf. Process. Syst.*, 28:379–387, 2015.

[5] J. Liang and R. D. C. Monteiro. An average curvature accelerated composite gradient method for nonconvex smooth composite optimization problems. *Available on arXiv:1909.04248*, 2019.

[6] J. Liang, R. D. C. Monteiro, and C.-K. Sim. A FISTA-type accelerated gradient algorithm for solving smooth nonconvex composite optimization problems. *Available on arXiv:1905.07010*, 2019.

Appendix A. Details of numerical experiments

In this section, we provide the details of numerical experiments presented in Section 3.

Before stating the constrained version of the NLRMC problem, we first give a few definitions. Let Ω be a subset of $\{1,\ldots,l\}\times\{1,\ldots,n\}$ and let Π_Ω denote the linear operator that maps a matrix A to the matrix whose entries in Ω have the same values of the corresponding ones in A and whose entries outside of Ω are all zero. Also, for given parameters $\beta > 0$ and $\theta > 0$, let $p : \mathbb{R} \to \mathbb{R}_+$ denote the log-sum penalty defined as

$$p(t) = p_{\beta,\theta}(t) := \beta \log\left(1 + \frac{|t|}{\theta}\right).$$

The constrained version of the NLRMC problem considered in numerical experiments is as in (11), where R is a positive scalar, $\mathcal{B}_R := \{Z \in \mathbb{R}^{l\times n} : \|Z\|_F \le R\}$, $O \in \mathbb{R}^\Omega$ is an incomplete observed matrix, $\mu > 0$ is a parameter, $r := \min\{l, n\}$ and $\sigma_i(Z)$ is the i-th singular value of Z.

The NLRMC problem in (11) is equivalent to

$$\min_{Z\in\mathbb{R}^{l\times n}} f(Z) + h(Z)$$

where

$$f(Z) = \frac{1}{2}\|\Pi_\Omega(Z - O)\|_F^2 + \mu\sum_{i=1}^{r}[p(\sigma_i(Z)) - p_0\sigma_i(Z)],$$
$$h(Z) = \mu p_0\|Z\|_* + I_{\mathcal{B}_R}(Z), \quad p_0 = p'(0) = \frac{\beta}{\theta},$$

and $\|\cdot\|_*$ denotes the nuclear norm defined as $\|\cdot\|_* := \sum_{i=1}^r \sigma_i(\cdot)$. Recall that assumption (A2) requires $\Delta \supset \operatorname{dom} h$, and hence we choose $\Delta = \mathcal{B}_R$.

We use the *MovieLens* dataset[1] to obtain the observed index set Ω and the incomplete observed matrix O. The dataset includes a sparse matrix with 100,000 ratings of {1,2,3,4,5} from 943 users

1. http://grouplens.org/datasets/movielens/

on 1682 movies, namely $l = 943$ and $n = 1682$. The radius R is chosen as the Frobenius norm of the matrix of size 943×1682 containing the same entries as O in Ω and 5 in the entries outside of Ω.

We start all seven methods from the same initial point Z_0 that is sampled from the standard Gaussian distribution and is within $\mathcal{B}_R$. The parameter pair (α, γ) is set to $(0.5, 0.01)$ in both AC and ACF. All seven methods terminate with a pair (Z, V) satisfying

$$V \in \nabla f(Z) + \partial h(Z), \qquad \frac{\|V\|}{\|\nabla f(Z_0)\| + 1} \leq \hat{\rho}$$

where $\hat{\rho} = 5 \times 10^{-4}$. All the computational results were obtained using MATLAB R2017b on a MacBook Pro with a quad-core Intel Core i7 processor and 16 GB of memory.