Adaptive Accelerated Gradient Converging Method under Hölderian Error Bound Condition

Mingrui Liu, Tianbao Yang

Department of Computer Science, The University of Iowa

Problem of Interest

Smooth Composite Optimization Problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) \triangleq f(\mathbf{x}) + g(\mathbf{x}) \quad (1)$$

- $f(\mathbf{x})$: continuously differentiable convex with L-Lipschitz continuous gradient
- $g(\mathbf{x})$: proper lower semi-continuous convex
- proximal mapping: $P_g(\mathbf{u}) = \mathrm{argmin}_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|_2^2 + g(\mathbf{x})/L$.
- proximal gradient $G(\mathbf{x}) = L(\mathbf{x} - P_g(\mathbf{x} - \nabla f(\mathbf{x})/L))$.

Accelerated Proximal Gradient Methods:

- Nesterov's APG:

$$\mathbf{x}_{\tau+1} = P_g(\mathbf{y}_\tau - \nabla f(\mathbf{y}_\tau)/L),\ \mathbf{y}_{\tau+1} = \mathbf{x}_{\tau+1} + \beta_\tau(\mathbf{x}_{\tau+1} - \mathbf{x}_\tau).$$

- $\beta_\tau = \frac{\tau}{\tau+3}$: iteration complexity (IC): $O(1/\sqrt{\epsilon})$
- if $f(\mathbf{x})$ is α-strongly convex: $\beta_\tau = \frac{\sqrt{L}-\sqrt{\tau}}{\sqrt{L}-\sqrt{\tau}}$, IC: $O(\sqrt{L/\alpha}\log(1/\epsilon))$
- if $g(\mathbf{x})$ is α-strongly convex: Nesterov's ADG, same IC as above

Algorithm 1 ADG

$\mathbf{x}_0 \in \Omega$, $A_0 = 0$, $\mathbf{v}_0 = \mathbf{x}_0$
for $t = 0, \ldots, T$ **do**
  Find a_{t+1} from quadratic equation $\frac{a^2}{A_t+a} = 2\frac{1+\alpha A_t}{L}$
  Set $A_{t+1} = A_t + a_{t+1}$
  Set $\mathbf{y}_t = \frac{A_t}{A_{t+1}}\mathbf{x}_t + \frac{a_{t+1}}{A_{t+1}}\mathbf{v}_t$
  Compute $\mathbf{x}_{t+1} = P_g(\mathbf{y}_t - \nabla f(\mathbf{y}_t)/L)$
  Compute $\mathbf{v}_{t+1} = \arg\min_{\mathbf{x}} \sum_{\tau=1}^{t+1} a_\tau \nabla f(\mathbf{x}_\tau)^\top \mathbf{x} + A_{t+1} g(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|_2^2$
end for

Recent Advances

Linear Convergence under weaker conditions, e.g., quadratic error bound condition (QEB, or quadratic growth condition).

$$\mathrm{dist}(\mathbf{x}, \Omega_*) \le c(F(\mathbf{x}) - F_*)^{1/2},$$

Examples: LASSO

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{n}\|A\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_1$$

- PG [4]: IC = $O(Lc^2\log(1/\epsilon))$
- a restarting version of APG [3]: IC = $O(\sqrt{Lc^2}\log(1/\epsilon))$ provided that the value of c is known.
- **Issue:** the value of c is usually unknown

Questions: how to develop algorithms with improved IC when c is unknown? What if considering a weaker condition than QEB?

Hölderian Error Bound (HEB)

Definition 1: A function $F(\mathbf{x})$ is said to satisfy a Hölderian error bound condition on ξ-sublevel set if there exist $\theta \in (0, 1]$ and $0 < c < \infty$ such that for any $\mathbf{x} \in \mathcal{S}_\xi$,

$$\mathrm{dist}(\mathbf{x}, \Omega_*) \le c(F(\mathbf{x}) - F_*)^\theta,$$

where Ω_* denotes the set of optimal solution.

- closely related to the Kurdyka - Łojasiewicz (KL) inequality in real algebraic geometry.
- when functions are semi-algebraic and continuous, the above inequality is known to hold on any compact set
- $\theta = 0$ can be considered as a special case

PG and restarting APG under HEB

Algorithm 2 PG under HEB

Input: $\mathbf{x}_0 \in \Omega$ such that $F(\mathbf{x}_0) - F(\mathbf{x}_*) \le \epsilon_0$
for $\tau = 1, \ldots, t$ **do**
  $\mathbf{x}_{\tau+1} = P_g(\mathbf{x}_\tau - \nabla f(\mathbf{x}_\tau)/L)$
end for
Option I: return $\mathbf{x}_{t+1}$
Option II: return $\mathbf{x}_k$ s.t. $G(\mathbf{x}_k) = \min_\tau \|G(\mathbf{x}_\tau)\|_2$

- Option I for achieving $F(\mathbf{x}_t) - F_* \le \epsilon$,
- IC is $O(\max\{\frac{Lc^2}{\epsilon^{1-2\theta}}, Lc^2\log(\frac{\epsilon_0}{\epsilon})\})$ if $\theta \le 1/2$, and $O(Lc^2\epsilon_0^{2\theta-1})$ ow.
- Option II is for achieving $G(\mathbf{x}_k) \le \epsilon$
- IC is $O\left(Lc^{\frac{1}{1-\theta}}\max\left\{\frac{1}{\epsilon^{\frac{1-2\theta}{1-\theta}}}, \log(\frac{\epsilon_0}{\epsilon})\right\}\right)$ if $\theta \le 1/2$, and is $O(c^2L\epsilon_0^{2\theta-1})$ ow.

Algorithm 3 restarting APG (rAPG) under HEB

Input: the number of stages K and $\mathbf{x}_0 \in \Omega$
for $k = 1, \ldots, K$ **do**
  Apply t_k steps of APG starting from $\mathbf{x}_{k-1}$
  Let $\mathbf{x}_k = \mathbf{x}^k_{t_k+1}$
end for
Output: $\mathbf{x}_K$

- Set $t_k = \lceil 2\sqrt{L}c(\frac{\epsilon_0}{2^{k-1}})^{\theta-1/2}\rceil$, for achieving $F(\mathbf{x}_K) - F_* \le \epsilon$
- IC is $O(\max\{\frac{c\sqrt{L}}{\epsilon^{1/2-\theta}}, c\sqrt{L}\log(\frac{\epsilon_0}{\epsilon})\})$ if $\theta \le 1/2$.
- IC is $O(\sqrt{L}c\epsilon_0^{1/2-\theta})$ if $\theta > 1/2$,
- rAPG requires the knowledge of c.

Adaptive Accelerated Gradient Converging Methods (adaAGC)

Algorithm 4 adaAGC

Input: $\mathbf{x}_0 \in \Omega$ and c_0 and $\gamma > 1$
Let $c_e = c_0$ and $\epsilon_0 = \|G(\mathbf{x}_0)\|_2$,
for $k = 1, \ldots, K$ **do**
  Let δ_k be given in (2) and $g_{\delta_k}(\mathbf{x}) = g(\mathbf{x}) + \frac{\delta_k}{2}\|\mathbf{x} - \mathbf{x}_{k-1}\|_2^2$
  $\mathbf{x}^k_1 = \mathbf{x}_{k-1}$ and $\mathbf{y}_1 = \mathbf{x}_{k-1}$
  for $s = 1, \ldots$ **do**
   for $\tau = 1, \ldots$ **do**
    Apply one step of ADG to $f(\mathbf{x}) + g_{\delta_k}(\mathbf{x})$ to generate $\mathbf{x}^k_{\tau+1}$
    if $\|G(\mathbf{x}^k_{\tau+1})\|_2 \le \epsilon_{k-1}/2$ **then**
     let $\mathbf{x}_k = \mathbf{x}^k_{\tau+1}$ and $\epsilon_k = \epsilon_{k-1}/2$.
     break the two enclosing for loops
    else if $\tau = \left\lceil 2\sqrt{\frac{L+\delta_k}{\delta_k}}\log\frac{\sqrt{L(L+\delta_k)}}{\delta_k}\right\rceil$ **then**
     let $c_e = \gamma c_e$ and break the enclosing for loop
    end if
   end for
  end for
end for
Output: $\mathbf{x}_K$

Why does the Algorithm Work?

- Adaptive Regularization:

$$\delta_k = \begin{cases} \min\left(\frac{L}{32}, \frac{\varepsilon_{k-1}^{\frac{1-2\theta}{1-\theta}}}{16c_e^{1/(1-\theta)}2^{\frac{\theta}{1-\theta}}}\right) & \text{if } \theta \in (0, 1/2] \\ \min\left(\frac{L}{32}, \frac{1}{32c_e^2\epsilon_0^{2\theta-1}}\right) & \text{if } \theta \in (1/2, 1] \end{cases} \quad (2)$$

- Conditional Restarting:
- Condition I: $\|G(\mathbf{x}^k_{\tau+1})\|_2 \le \epsilon_{k-1}/2$
- Condition II: $\tau = \left\lceil 2\sqrt{\frac{L+\delta_k}{\delta_k}}\log\frac{\sqrt{L(L+\delta_k)}}{\delta_k}\right\rceil$
- If Condition II becomes true before Condition I, then $c_e < c$
- Reasoning: ADG guarantees that

$$\|G(\mathbf{x}^k_{\tau+1})\|_2 \le \left(\sqrt{L(L+\delta_k)}\left(1 + \sqrt{\frac{\delta_k}{2L}}\right)^{-\tau} + 2\sqrt{2}\delta_k\right)\|\mathbf{x}_{k-1} - \mathbf{x}_*\|_2.$$

Our Key Lemma: Assume HEB for any $\mathbf{x} \in \mathcal{S}_\xi$ with $\theta \in (0, 1]$.

$$\|\mathbf{x} - \mathbf{x}_*\|_2 \le \frac{2}{L}\|G(\mathbf{x})\|_2 + c^{\frac{1}{1-\theta}}2^{\frac{\theta}{1-\theta}}\|G(\mathbf{x})\|_2^{\frac{\theta}{1-\theta}},\ \theta \in (0, 1/2]$$

$$\|\mathbf{x} - \mathbf{x}_*\|_2 \le \left(\frac{2}{L} + 2c^2\xi^{2\theta-1}\right)\|G(\mathbf{x})\|_2.,\ \theta \in (1/2, 1)$$

Main Theorem: Suppose $F(\mathbf{x}_0) - F_* \le \epsilon_0$, $F(\mathbf{x})$ satisfies HEB on $\mathcal{S}_{\epsilon_0}$ and $c_0 \le c$. Let $\varepsilon_0 = \|G(\mathbf{x}_0)\|_2$, $K = \lceil\log_2(\frac{\varepsilon_0}{\epsilon})\rceil$, $p = (1-2\theta)/(1-\theta)$ for $\theta \in (0, 1/2]$. The IC of adaAGC for having $\|G(\mathbf{x}_K)\|_2 \le \epsilon$ is (where $\widetilde{O}(\cdot)$ suppresses a log term depending on c, c_0, L, γ)

$$\mathrm{IC} = \begin{cases} \widetilde{O}\left(\sqrt{L}c^{\frac{1}{2(1-\theta)}}\max(\frac{1}{\epsilon^{p/2}}, \log(\varepsilon_0/\epsilon)\right) & \text{if } \theta \in (0, 1/2) \\ \widetilde{O}\left(\sqrt{L}c\log(\varepsilon_0/\epsilon)\right) & \text{if } \theta = 1/2 \\ \widetilde{O}(\sqrt{L}c\epsilon_0^{\theta-1/2}) & \theta \in (1/2, 1] \end{cases}$$

Table 1: Summary of iteration complexities in this work under the HEB condition with $\theta \in (0, 1/2]$, where $G(\mathbf{x})$ denotes the proximal gradient, $\mathcal{C}(1/\epsilon^\alpha) = \max(1/\epsilon^\alpha, \log(1/\epsilon))$ and $\widetilde{O}(\cdot)$ suppresses a logarithmic term. If $\theta > 1/2$, all algorithms can converge with finite steps of proximal mapping. rAPG stands for restarting APG. $*$ mark results available for certain subclasses of problems.

algo.	PG	rAPG	adaAGC
$F(\mathbf{x}) - F_* \le \epsilon$	$O\left(c^2L\mathcal{C}\left(\frac{1}{\epsilon^{1-2\theta}}\right)\right)$	$O\left(c\sqrt{L}\mathcal{C}\left(\frac{1}{\epsilon^{1/2-\theta}}\right)\right)$	$O\left(c\sqrt{L}\mathcal{C}\left(\frac{1}{\epsilon^{1/2-\theta}}\right)\right)^*$
$\|G(\mathbf{x})\|_2 \le \epsilon$	$O\left(c^{\frac{1}{1-\theta}}L\mathcal{C}\left(\frac{1}{\epsilon^{\frac{1-2\theta}{1-\theta}}}\right)\right)$	–	$\widetilde{O}\left(c^{\frac{1}{2(1-\theta)}}\sqrt{L}\mathcal{C}\left(\frac{1}{\epsilon^{\frac{1-2\theta}{2(1-\theta)}}}\right)\right)$
requires θ	No	Yes	Yes
requires c	No	Yes	No

Applications and Experimental Results

Applications: Consider the regularized problems with a smooth loss:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n \ell(\mathbf{x}^\top\mathbf{a}_i, b_i) + \lambda R(\mathbf{x}), \quad (3)$$

where $(\mathbf{a}_i, b_i), i = 1, \ldots, n$ denote a set of training examples, $R(\mathbf{x})$ is the regularizer.

Examples of $\theta = 1/2$: piecewise quadratic convex function [1]

- square loss, squared hinge loss, huber loss
- ℓ_1, ℓ_∞ norm, Huber norm and $\ell_{1,\infty}$ norm.

Examples of $\theta = 1/2$: structured smooth composite functions

- $f(\mathbf{x}) = h(A\mathbf{x})$: h is smooth and strongly convex on any compact set
- $g(\mathbf{x})$ is a polyhedral function (e.g., ℓ_1 norm)

Examples of $\theta < 1/2$: ℓ_1 constrained ℓ_p regression: $\theta = 1/p$

$$\min_{\|\mathbf{x}\|_1 \le s} F(\mathbf{x}) \triangleq \frac{1}{n}\sum_{i=1}^n (\mathbf{a}_i^\top\mathbf{x} - b_i)^p, \quad p \in 2\mathbb{N}. \quad (4)$$

Experimental Results:

Table 2: squared hinge loss with ℓ_1 norm (left) and ℓ_∞ norm (right) regularization on splice data

Algorithm	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-7}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-7}$
PG	2040	2040	2040	2040	3514	3724	3724	3724
FISTA	**1289**	**1289**	**1289**	**1289**	5526	5526	5526	5526
urFISTA	1666	2371	2601	3480	**1674**	**2379**	2605	3488
adaAGC	1410	1410	1410	1410	2382	2382	**2382**	**2382**
FISTA > adaAGC > PG > urFISTA					adaAGC > urFISTA > PG > FISTA			

Table 3: square loss with ℓ_1 norm (left) and ℓ_∞ norm (right) regularization on cpusmall data

Algorithm	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-7}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-7}$
PG	109298	159908	170915	170915	139505	204120	210874	210874
FISTA	**6781**	16387	23779	23779	**6610**	16418	20082	20082
urFISTA	18278	26706	35173	43603	18276	26704	35169	43601
adaAGC	9571	**12623**	**13575**	**13575**	9881	**13033**	**13632**	**13632**
adaAGC > FISTA > urFISTA > PG					adaAGC > FISTA > urFISTA > PG			

Table 4: ℓ_1 regularized huber loss (left) and ℓ_1 constrained square loss (right) on bodyfat data

Algorithm	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-7}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-7}$
PG	258723	423181	602043	681488	1006880	1768482	2530085	2632578
FISTA	**6630**	25020	74416	124261	**15805**	66319	180977	181176
urFISTA	6855	**12662**	**17994**	**23933**	138359	235081	331203	426341
adaAGC	16976	16980	23844	25697	23054	**33818**	**44582**	**48127**
urFISTA > adaAGC > FISTA > PG					adaAGC> FISTA > urFISTA > PG			

Table 5: ℓ_1 constrained ℓ_p norm regression on bodyfat data ($\epsilon = 10^{-3}$)

Algorithm	$p = 2$	$p = 4$	$p = 6$	$p = 8$
PG	250869 (1)	979401 (3.90)	1559753 (6.22)	4015665 (16.00)
adaAGC	**8710 (1)**	**17494 (2.0)**	**22481 (2.58)**	**33081 (3.80)**

Remark:

- For comparison of urFISTA [2] and adaAGC, we use the same initial estimate of the unknown Hölderian constant c.
- In Table 4, the numbers in parenthesis indicate the increasing factor in the number of proximal mappings compared to the base case $p = 2$, which is consistent with theory.

References:

[1] G. Li. Global error bounds for piecewise convex polynomials. Math. Program., 137(1-2):37-64. 2013.

[2] O. Fercoq and Z. Qu. Restarting accelerated gradient methods with a rough strong convexity estimate.arXiv preprint arXiv:1609.07358, 2016.

[3] I. Necoara, Y. Nesterov, and F. Glineur. Linear convergence of first order methods for nonstrongly convex optimization. CoRR, abs/1504.06298, 2015.

[4]. H. Karimi, J. Nutini, and M. W. Schmidt. Linear convergence of gradient and proximal gradient methods under the Polyak - Łojasiewicz condition. In ECML-PKDD, 2016.