Adaptive Accelerated Gradient Converging Methods under Hölderian Error Bound Condition

Mingrui Liu MINGRUI-LIU@UIOWA.EDU
Tianbao Yang TIANBAO-YANG@UIOWA.EDU
Department of Computer Science
The University of Iowa, Iowa City, IA 52242

First version: November 22, 2016

Abstract

Recent studies have shown that proximal gradient (PG) method and accelerated gradient method (APG) with restarting can enjoy a linear convergence under a weaker condition than strong convexity, namely a quadratic growth condition (QGC). However, the faster convergence of restarting APG method relies on the potentially unknown constant in QGC to appropriately restart APG, which restricts its applicability. We address this issue by developing a novel adaptive gradient converging methods, i.e., leveraging the magnitude of proximal gradient as a criterion for restart and termination. Our analysis extends to a much more general condition beyond the QGC, namely the Hölderian error bound (HEB) condition. *The key technique* for our development is a novel synthesis of *adaptive regularization and a conditional restarting scheme*, which extends previous work focusing on strongly convex problems to a much broader family of problems. Furthermore, we demonstrate that our results have important implication and applications in machine learning: (i) if the objective function is coercive and semi-algebraic, PG's convergence speed is essentially $o(\frac{1}{t})$, where t is the total number of iterations; (ii) if the objective function consists of an ℓ_1, ℓ_∞, $\ell_{1,\infty}$, or huber norm regularization and a convex smooth piecewise quadratic loss (e.g., squares loss, squared hinge loss and huber loss), the proposed algorithm is parameter-free and enjoys a *faster linear convergence* than PG without any other assumptions (e.g., restricted eigen-value condition). It is notable that our linear convergence results for the aforementioned problems are global instead of local. To the best of our knowledge, these improved results are the first shown in this work.

1. Introduction

We consider the following smooth optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}), \tag{1}$$

where $f(\mathbf{x})$ is a continuously differential convex function, whose gradient is L-Lipschitz continuous. More generally, we also tackle the following composite optimization:

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) \triangleq f(\mathbf{x}) + g(\mathbf{x}), \tag{2}$$

where $g(\mathbf{x})$ is a proper lower semi-continuous convex function and $f(\mathbf{x})$ is a continuously differentiable convex function, whose gradient is L-Lipschitz continuous. The above problem has been studied extensively in literature and many algorithms have been developed with

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.

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

convergence guarantee. In particular, by employing the proximal mapping associated with $g(\mathbf{x})$, i.e.,

$$P_{\eta g}(\mathbf{u}) = \arg\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{x} - \mathbf{u}\|_2^2 + \eta g(\mathbf{x}), \tag{3}$$

proximal gradient (PG) and accelerated proximal gradient (APG) methods have been developed for solving (2) with $O(1/\epsilon)$ and $O(1/\sqrt{\epsilon})$ [1] iteration complexities for finding an ϵ-optimal solution. When either $f(\mathbf{x})$ or $g(\mathbf{x})$ is strongly convex, both PG and APG can enjoy a linear convergence, i.e., the iteration complexity is improved to be $O(\log(1/\epsilon))$.

Recently, a wave of study is to generalize the linear convergence to problems without strong convexity but under certain structured condition of the objective function or more generally a quadratic growth condition (Hou et al., 2013; Zhou et al., 2015; So, 2013; Wang and Lin, 2014a; Gong and Ye, 2014; Zhou and So, 2015; Bolte et al., 2015; Necoara et al., 2015; Karimi et al., 2016; Zhang, 2016a; Drusvyatskiy and Lewis, 2016). Earlier work along the line dates back to (Luo and Tseng, 1992a,b, 1993). An example of the structured condition is such that $f(\mathbf{x}) = h(A\mathbf{x})$ where $h(\cdot)$ is strongly convex function and $\nabla h(\mathbf{x})$ is Lipschitz continuous on any compact set, and $g(\mathbf{x})$ is a polyhedral function. Under such a structured condition, a local error bound condition can be established (Luo and Tseng, 1992a,b, 1993), which renders an asymptotic (local) linear convergence for the proximal gradient method. A quadratic growth condition (QGC) prescribes that the objective function satisfies for any $\mathbf{x} \in \mathbb{R}^d$ [2]: $\frac{\alpha}{2}\|\mathbf{x} - \mathbf{x}_*\|_2^2 \leq F(\mathbf{x}) - F(\mathbf{x}_*)$, where $\mathbf{x}_*$ denotes a closest point to $\mathbf{x}$ in the optimal set. Under such a quadratic growth condition, several recent studies have established the linear convergence of PG, APG and many other algorithms (e.g., coordinate descent methods) (Bolte et al., 2015; Necoara et al., 2015; Drusvyatskiy and Lewis, 2016; Karimi et al., 2016; Zhang, 2016a). A notable result is that PG enjoys an iteration complexity of $O(\frac{L}{\alpha}\log(1/\epsilon))$ without knowing the value of α, while a restarting version of APG studied in Necoara et al. (2015) enjoys an improved iteration complexity of $O(\sqrt{\frac{L}{\alpha}}\log(1/\epsilon))$ hinging

1. For the moment, we neglect the constant factor.
2. It can be relaxed to a fixed domain as done in this work.

on the value of α to appropriately restart APG periodically. Other equivalent conditions or more restricted conditions are also considered in several studies to show the linear convergence of (proximal) gradient method and other methods (Karimi et al., 2016; Necoara et al., 2015; Zhang, 2016a,b).

In this paper, we extend this line of work to a more general error bound condition, i.e., the Hölderian error bound (HEB) condition on a compact sublevel set $\mathcal{S}_\xi = \{\mathbf{x} \in \mathbb{R}^d : F(\mathbf{x}) - F(\mathbf{x}_*) \leq \xi\}$: there exists $\theta \in (0, 1]$ and $0 < c < \infty$ such that

$$\|\mathbf{x} - \mathbf{x}_*\|_2 \leq c(F(\mathbf{x}) - F(\mathbf{x}_*))^\theta, \ \forall \mathbf{x} \in \mathcal{S}_\xi. \tag{4}$$

Note that when $\theta = 1/2$ and $c = \sqrt{1/\alpha}$, the HEB reduces to the QGC. In the sequel, we will refer to $C = Lc^2$ as condition number of the problem. It is worth mentioning that Bolte et al. (2015) considered the same condition or an equivalent Kurdyka - Łojasiewicz inequality but they only focused on descent methods that bear a sufficient decrease condition for each update consequentially excluding APG. In addition, they do not provide explicit iteration complexity under the general HEB condition.

As a warm-up and motivation, we will first present a straightforward analysis to show that PG is automatically adaptive and APG can be made adaptive to the HEB by restarting. In particular if $F(\mathbf{x})$ satisfies a HEB condition on the initial sublevel set, PG has an iteration complexity of $O(\max(\frac{C}{\epsilon^{1-2\theta}}, C\log(\frac{1}{\epsilon})))$ [3], and restarting APG enjoys an iteration complexity of $O(\max(\frac{\sqrt{C}}{\epsilon^{1/2-\theta}}, \sqrt{C}\log(\frac{1}{\epsilon})))$ for the convergence of objective value, where $C = Lc^2$ is the condition number. These two results resemble but generalize recent works that establish linear convergence of PG and restarting APG under the QGC - a special case of HEB. Although enjoying faster convergence, restarting APG has some caveats: (i) it requires the knowledge of constant c in HEB to restart APG, which is usually difficult to compute or estimate; (ii) there lacks an appropriate machinery to terminate the algorithm. In this paper, we make nontrivial contributions to obtain faster convergence of the proximal gradient's norm under the HEB condition by developing an adaptive accelerated gradient converging method.

The main results of this paper are summarized in Table 1. In summary the contributions of this paper are:

- We extend the analysis of PG and restarting APG under the quadratic growth condition to more general HEB condition, and establish the adaptive iteration complexities of both algorithms.
- To enjoy faster convergence of restarting APG and to eliminate the algorithmic dependence on the unknown parameter c, we propose and analyze an adaptive accelerated gradient converging (adaAGC) method.

The developed algorithms and theory have important implication and applications in machine learning. Firstly, if the considered objective function is also coercive and semi-algebraic (e.g., a norm regularized problem in machine learning with a semi-algebraic loss function), then PG's convergence speed is essentially $o(1/t)$ instead of $O(1/t)$, where t is the total number of iterations. Secondly, for solving ℓ_1, ℓ_∞ or $\ell_{1,\infty}$ regularized smooth loss

3. When $\theta > 1/2$, all algorithms can converge in finite steps.

minimization problems including least-squares loss, squared hinge loss and huber loss, the proposed adaAGC method enjoys a linear convergence and a square root dependence on the "condition" number. In contrast to previous work, the proposed algorithm is parameter free and does not rely on any restricted conditions (e.g., the restricted eigen-value conditions).

2. Related Work

At first, we review some related work for solving the problem (1) and (2). In Nesterov's seminal work (Nesterov, 1983, 2007), the accelerated (proximal) gradient (APG) method were proposed for (composite) smooth optimization problems, enjoying $O(1/\sqrt{\epsilon})$ iteration complexity for achieving a ϵ-optimal solution. When the objective is also strongly convex, APG can converge to the optimal solution linearly with an appropiate step size depending on the strong convexity modulus, which enjoys $O(\log(1/\epsilon))$ iteration complexity.

To address the issue of unknown strong convexity modulus for some problems, several restarting schemes were developed. Nesterov (2007) proposed a restarting scheme for the APG method to approximate the unknown strongly convexity parameter and achieved a linear convergence rate. Lin and Xiao (2014) proposed an adaptive APG method which employs the restart and line search technique to automatically estimate the strong convexity parameter. Odonoghue and Candes (2015) proposed an heuristic approach to adaptively restart accelerated gradient schemes and showed good experimental results. Nevertheless, they provide no theoretical guarantee of their proposed heuristic approach. In contrast to these work, we do not assume any strong convexity or restricted strong convexity for sparse learning. It was brought to our attention that a recent work (Fercoq and Qu, 2016) considered QGC and proposed restarted accelerated gradient and coordinate descent methods, including APG, FISTA and the accelerated proximal coordinate descent method (APPROX). The difference from their restarting scheme for APG and the restarting schemes in (Nesterov, 2007; Lin and Xiao, 2014; Odonoghue and Candes, 2015) and the present work is that their restart doest not involve evaluation of the gradient or the objective value but rather depends on a restarting frequency parameter and a convex combination parameter for computing the restarting solution, which can be set based on a rough estimate of the strong convexity parameter. As a result, their linear convergence (established for distance of solutions to the optimal set) heavily depends on the rough estimate of the strong convexity parameter.

Leveraging error bound conditions dates back to (Luo and Tseng, 1992a,b, 1993), which employed the error bound condition to establish the asymptotic (local) linear convergence for feasible descent methods. Luo & Tseng' bounds the distance of a local solution to the optmal set by the norm of proximal gradient. Several recent work (Hou et al., 2013; Zhou et al., 2015; So, 2013) have considered Luo & Pseng's error bound condition for more problems in machine learning and established local linear convergence for proximal gradient methods. Wang and Lin (2014b) established a global error bound version of Luo & Pseng's condition for a family of problems in machine learning (e.g., the dual formulation of SVM), and provided the global linear convergence for a series of algorithms, including cyclic coordinate descent methods for solving dual support vector machine. Note that the Hölderian error bound (Bolte et al., 2015) used in our analysis is different from Luo & Pseng's condition, and is actually more general. Bolte et al. (2015) established the equivalence of HEB and Kurdyka-Łojasiewicz (KL) inequality and showed how to derive lower computational complexity via employing KL inequality. As a special case of Hölderian error bound condition, quadratic growth condition (QGC) has been considered in several recent work for deriving linear convergence. Gong and Ye (2014) established linear convergence of proximal variance-reduced gradient (Prox-SVRG) algorithm under QGC. Necoara et al. (2015) showed that QGC is one of the relaxations of strong convexity conditions, which can still guarantee the linear convergence for several first order methods, including projected gradient, fast gradient and feasible descent methods. Drusvyatskiy and Lewis (2016) also showed that proximal gradient algorithm achieved the linear convergence under QGC. There also exist other conditions (stronger than or equivalent to QGC) that can help achieve linear convergence rate. For example, Karimi et al. (2016) showed that the Polyak-Łojasiewicz (PL) inequality suffices to guarantee a global linear convergence for (proximal) gradient descent methods. Zhang (2016a) summarized different sufficient conditions which are capable of deriving linear convergence, and discussed their relationships.

3. Notations and Preliminaries

In this section, we present some notations and preliminaries. In the sequel, we let $\|\cdot\|_p$ $(p \geq 1)$ denote the p-norm of a vector. A function $g(\mathbf{x}) : \mathbb{R}^d \rightarrow]-\infty, \infty]$ is a proper function if $g(\mathbf{x}) < +\infty$ for at least one $\mathbf{x}$ and $g(\mathbf{x}) > -\infty$ for all $\mathbf{x}$. $g(\mathbf{x})$ is lower semi-continuous at a point $\mathbf{x}_0$ if $\liminf_{\mathbf{x}\rightarrow\mathbf{x}_0} g(\mathbf{x}) = g(\mathbf{x}_0)$. A function $F(\mathbf{x})$ is coercive if and only if $F(\mathbf{x}) \rightarrow \infty$ as $\|\mathbf{x}\|_2 \rightarrow \infty$.

A subset $S \subset \mathbb{R}^d$ is a real semi-algebraic set if there exists a finite number of real polynomial functions $g_{ij}, h_{ij} : \mathbb{R}^d \rightarrow \mathbb{R}$ such that

$$S = \cup_{j=1}^{p} \cap_{i=1}^{q} \{\mathbf{u} \in \mathbb{R}^d; g_{ij}(\mathbf{u}) = 0 \text{ and } h_{ij}(\mathbf{u}) \leq 0\}.$$

A function $F(\mathbf{x})$ is semi-algebraic if its graph $\{(\mathbf{u}, s) \in \mathbb{R}^{d+1} : F(\mathbf{u}) = s\}$ is a semi-algebraic set.

Denote by $\mathbb{N}$ the set of all positive integers. A function $h(\mathbf{x})$ is a real polynomial if there exists $r \in \mathbb{N}$ such that $h(\mathbf{x}) = \sum_{0\leq|\alpha|\leq r} \lambda_\alpha \mathbf{x}^\alpha$, where $\lambda_\alpha \in \mathbb{R}$ and $\mathbf{x}^\alpha = x_1^{\alpha_1} \ldots x_d^{\alpha_d}$, $\alpha_j \in \mathbb{N}\cup\{0\}$, $|\alpha| = \sum_{j=1}^{d} \alpha_j$ and r is referred to as the degree of $h(\mathbf{x})$. A continuous function $f(\mathbf{x})$ is said to be a piecewise convex polynomial if there exist finitely many polyhedra $P_1, \ldots, P_k$ with $\cup_{j=1}^{k} P_j = \mathbb{R}^n$ such that the restriction of f on each P_j is a convex polynomial. Let f_j be the restriction of f on P_j. The degree of a piecewise convex polynomial function f denoted by $deg(f)$ is the maximum of the degree of each f_j. If $deg(f) = 2$, the function is referred to as a piecewise convex quadratic function. Note that a piecewise convex polynomial function is not necessarily a convex function (Li, 2013).

A function $f(\mathbf{x})$ is L-smooth w.r.t $\|\cdot\|_2$ if it is differentiable and has a Lipschitz continuous gradient with the Lipschitz constant L, i.e., $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq L\|\mathbf{x} - \mathbf{y}\|_2, \forall \mathbf{x}, \mathbf{y}$. Let $\partial g(\mathbf{x})$ denote the subdifferential of g at $\mathbf{x}$, i.e.,

$$\partial g(\mathbf{x}) = \{\mathbf{u} \in \mathbb{R}^d : g(\mathbf{y}) \geq g(\mathbf{x}) + \mathbf{u}^\top(\mathbf{y} - \mathbf{x}), \forall \mathbf{y}\}.$$

Denote by $\|\partial g(\mathbf{x})\|_2 = \min_{\mathbf{u}\in\partial g(\mathbf{x})} \|\mathbf{u}\|_2$. A function $g(\mathbf{x})$ is α-strongly convex w.r.t $\|\cdot\|_2$ if it satisfies for any $\mathbf{u} \in \partial g(\mathbf{y})$ such that $g(\mathbf{x}) \geq g(\mathbf{y}) + \mathbf{u}^\top(\mathbf{x} - \mathbf{y}) + \frac{\alpha}{2}\|\mathbf{x} - \mathbf{y}\|_2^2, \forall \mathbf{x}, \mathbf{y}$.

Denote by $\eta > 0$ a positive scalar, and let $P_{\eta g}$ be the proximal mapping associated with $\eta g(\cdot)$ defined in (3). Given an objective function $F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$, where $f(\mathbf{x})$ is L-smooth and $g(\mathbf{x})$ is a simple non-smooth function, define a **proximal gradient** $G_\eta(\mathbf{x})$ as:

$$G_\eta(\mathbf{x}) = \frac{1}{\eta}(\mathbf{x} - \mathbf{x}_\eta^+), \text{ where } \mathbf{x}_\eta^+ = P_{\eta g}(\mathbf{x} - \eta \nabla f(\mathbf{x}))$$

When $g(\mathbf{x}) = 0$, we have $G_\eta(\mathbf{x}) = \nabla f(\mathbf{x})$, i.e., the proximal gradient is the gradient. It is known that $\mathbf{x}$ is an optimal solution iff $G_\eta(\mathbf{x}) = 0$. If $\eta = 1/L$, for simplicity we denote by $G(\mathbf{x}) = G_{1/L}(\mathbf{x})$ and $\mathbf{x}^+ = P_{g/L}(\mathbf{x} - \nabla f(\mathbf{x})/L)$. Below, we give several technical propositions related to $G_\eta(\mathbf{x})$ and the proximal gradient update.

Proposition 1 *(Nesterov, 2007) Given* $\mathbf{x}$, *$\|G_\eta(\mathbf{x})\|_2$ is a monotonically decreasing function of η.*

Proposition 2 *(Beck and Teboulle, 2009) Let $F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$. Assume $f(\mathbf{x})$ is L-smooth. For any $\mathbf{x}, \mathbf{y}$ and $\eta \le 1/L$, we have*

$$F(\mathbf{y}_\eta^+) \le F(\mathbf{x}) + G_\eta(\mathbf{y})^\top (\mathbf{y} - \mathbf{x}) - \frac{\eta}{2}\|G_\eta(\mathbf{y})\|_2^2. \tag{5}$$

The following corollary is useful for our analysis.

Corollary 1 *Let $F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$. Assume $f(\mathbf{x})$ is L-smooth. For any $\mathbf{x}, \mathbf{y}$ and $0 < \eta \le 1/L$, we have*

$$\frac{\eta}{2}\|G_\eta(\mathbf{y})\|_2^2 \le F(\mathbf{y}) - F(\mathbf{y}_\eta^+) \le F(\mathbf{y}) - \min_{\mathbf{x}} F(\mathbf{x}). \tag{6}$$

Remark: The proof of Corollary 1 is immediate by employing the convexity of F and Proposition 2.

Let F_* denote the optimal objective value to $\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x})$ and Ω_* denote the optimal set. Denote by $\mathcal{S}_\xi = \{\mathbf{x} : F(\mathbf{x}) - F_* \le \xi\}$ the ξ-sublevel set of $F(\mathbf{x})$. Let $D(\mathbf{x}, \Omega) = \min_{\mathbf{y}\in\Omega} \|\mathbf{x} - \mathbf{y}\|_2$.

The proximal gradient (PG) method solves the problem (2) by the update

$$\mathbf{x}_{t+1} = P_{\eta g}(\mathbf{x}_t - \eta \nabla f(\mathbf{x}_t)), \tag{7}$$

with $\eta \le 1/L$ starting from some initial solution $\mathbf{x}_1 \in \mathbb{R}^d$. It can be shown that PG has an iteration complexity of $O(\frac{LD(\mathbf{x}_1,\Omega_*)^2}{\epsilon})$. The convergence guarantee of PG is presented in the following proposition.

Proposition 3 *(Nesterov, 2004) Let (7) run for $t = 1, \ldots, T$ with $\eta \le 1/L$, we have*

$$F(\mathbf{x}_{T+1}) - F_* \le \frac{D(\mathbf{x}_1, \Omega_*)^2}{2\eta T}.$$

Based on the above proposition, one can deduce that PG has an iteration complexity of $O(\frac{LD(\mathbf{x}_1,\Omega_*)^2}{\epsilon})$. Nevertheless, accelerated proximal gradient (APG) converges faster than PG. There are many variants of APG in literature (Tseng, 2008). The simplest variant adopts the following update

$$\begin{cases} \mathbf{y}_t = \mathbf{x}_t + \beta_t(\mathbf{x}_t - \mathbf{x}_{t-1}), \\ \mathbf{x}_{t+1} = P_{\eta g}(\mathbf{y}_t - \eta \nabla f(\mathbf{y}_t)), \end{cases} \tag{8}$$

where $\eta \leq 1/L$ and $\beta_t = \frac{t-1}{t+2}$. APG enjoys an iteration complexity of $O(\frac{\sqrt{L}D(\mathbf{x}_1,\Omega_*)}{\sqrt{\epsilon}})$ (Tseng, 2008). The convergence guarantee of APG is presented in the following proposition.

Proposition 4 *(Tseng, 2008) Let (8) run for $t = 1, \ldots, T$ with $\eta \leq 1/L$ and $\mathbf{x}_0 = \mathbf{x}_1$, we have*

$$F(\mathbf{x}_{T+1}) - F_* \leq \frac{2D(\mathbf{x}_1, \Omega_*)^2}{\eta(T+1)^2}.$$

Based on the above proposition, one can deduce that APG has an iteration complexity of $O(\frac{\sqrt{L}D(\mathbf{x}_1,\Omega_*)}{\sqrt{\epsilon}})$.

Furthermore, if $f(\mathbf{x})$ is both L-smooth and α-strongly convex, one can set $\beta_t = \frac{\sqrt{L}-\sqrt{\alpha}}{\sqrt{L}+\sqrt{\alpha}}$ and deduce a linear convergence (Lin and Xiao, 2014) with a better dependence on the condition number than that of PG.

Proposition 5 *(Lin and Xiao, 2014) Assume $f(\mathbf{x})$ is L-smooth and α-strongly convex. Let (8) run for $t = 1, \ldots, T$ with $\eta = 1/L$, $\beta_t = \frac{\sqrt{L}-\sqrt{\alpha}}{\sqrt{L}+\sqrt{\alpha}}$ and $\mathbf{x}_0 = \mathbf{x}_1$, we have for any $\mathbf{x}$*

$$F(\mathbf{x}_{T+1}) - F(\mathbf{x}) \leq \left(1 - \sqrt{\frac{\alpha}{L}}\right)^T \left[F(\mathbf{x}_0) - F(\mathbf{x}) + \frac{\alpha}{2}\|\mathbf{x}_0 - \mathbf{x}\|_2^2\right].$$

If $\phi(\mathbf{x})$ is α-strongly convex and $f(\mathbf{x})$ is L-smooth, Nesterov (2007) proposed a different variant based on dual averaging, which is referred to accelerated dual gradient (ADG) method and will be useful for our develeopment. The key steps are presented in Algorithm 1. The convergence guarantee of ADG is given the following proposition.

Proposition 6 *(Nesterov, 2007) Assume $f(\mathbf{x})$ is L-smooth and $g(\mathbf{x})$ is α-strongly convex. Let Algorithm 1 run for $t = 0, \ldots, T$. Then for any $\mathbf{x}$ we have*

$$F(\mathbf{x}_{T+1}) - F(\mathbf{x}) \leq \frac{L}{2}\|\mathbf{x}_0 - \mathbf{x}\|_2^2 \left(\frac{1}{1 + \sqrt{\alpha/2L}}\right)^{2T}.$$

A. Hölderian error bound (HEB) condition

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

$$dist(\mathbf{x}, \Omega_*) \leq c(F(\mathbf{x}) - F_*)^\theta, \tag{9}$$

where Ω_* denotes the optimal set of $\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x})$.

Algorithm 1 ADG

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

The HEB condition is closely related to the Łojasiewicz inequality or more generally Kurdyka- Łojasiewicz (KL) inequality in real algebraic geometry. It has been shown that when functions are semi-algebraic and continuous, the above inequality is known to hold on any compact set (Bolte et al., 2015). We refer the readers to (Bolte et al., 2015) for more discussions on HEB and KL inequalities.

In the remainder of this section, we will review some previous results to demonstrate that HEB is a generic condition that holds for a broad family of problems of interest. The following proposition states that any proper, coercive, convex, lower-semicontinuous and semi-algebraic functions satisfy the HEB condition.

Proposition 7 *(Bolte et al., 2015) Let $F(\mathbf{x})$ be a proper, coercive, convex, lower semicontinuous and semi-algebraic function. Then there exists $\theta \in (0, 1]$ and $0 < c < \infty$ such that $F(\mathbf{x})$ satisfies the HEB on any ξ-sublevel set.*

Example: Most optimization problems in machine learning with an objective that consists of an empirical loss that is semi-algebraic (e.g., hinge loss, squared hinge loss, absolute loss, square loss) and a norm regularization $\|\cdot\|_p$ ($p \geq 1$ is a rational) or a norm constraint are proper, coercive, lower semicontinuous and semi-algebraic functions.

Next two propositions exhibit the value θ for piecewise convex quadratic functions and piecewise convex polynomial functions.

Proposition 8 *(Li, 2013) Let $F(\mathbf{x})$ be a piecewise convex quadratic function on $\mathbb{R}^d$. Suppose $F(\mathbf{x})$ is convex. Then for any $\xi > 0$, there exists $0 < c < \infty$ such that*

$$D(\mathbf{x}, \Omega_*) \leq c(F(\mathbf{x}) - F_*)^{1/2}, \forall \mathbf{x} \in \mathcal{S}_\xi.$$

Many problems in machine learning are piecewise convex quadratic functions, which will be discussed more in Section 7.

Proposition 9 *(Li, 2013) Let $F(\mathbf{x})$ be a piecewise convex polynomial function on $\mathbb{R}^d$. Suppose $F(\mathbf{x})$ is convex. Then for any $\xi > 0$, there exists $c > 0$ such that*

$$D(\mathbf{x}, \Omega_*) \leq c(F(\mathbf{x}) - F_*)^{\frac{1}{(deg(F)-1)^d + 1}}, \forall \mathbf{x} \in \mathcal{S}_\xi.$$

Indeed, for a polyhedral constrained convex polynomial, we can have a tighter result, as show below.

Proposition 10 *(Yang, 2009) Let $F(\mathbf{x})$ be a convex polynomial function on $\mathbb{R}^d$ with degree m. If $P \subset \mathbb{R}^d$ is a polyhedral set, then the problem $\min_{\mathbf{x}\in P} F(\mathbf{x})$ admits a global error bound: $\forall \mathbf{x} \in P$ there exists $0 < c < \infty$ such that*

$$D(\mathbf{x}, \Omega_*) \leq c\left[(F(\mathbf{x}) - F_*) + (F(\mathbf{x}) - F_*)^{\frac{1}{m}}\right], \tag{10}$$

From the global error bound (10), one can easily derive the Hölderian error bound condition (4). For an example, we can consider an ℓ_1 constrained ℓ_p norm regression (Nyquist, 1983):

$$\min_{\|\mathbf{x}\|_1 \leq s} F(\mathbf{x}) \triangleq \frac{1}{n}\sum_{i=1}^{n}(\mathbf{a}_i^\top \mathbf{x} - b_i)^p, \quad p \in 2\mathbb{N} \tag{11}$$

which satisfies the HEB condition (4) with $\theta = \frac{1}{p}$.

Many previous papers have considered a family of structured smooth composite problems:

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) = h(A\mathbf{x}) + g(\mathbf{x}) \tag{12}$$

where $g(\mathbf{x})$ is a polyhedral function and $h(\cdot)$ is a smooth and strongly convex function on any compact set. Suppose the optimal set of the above problem is non-empty and compact (e.g., the function is coercive) so is the sublevel set $\mathcal{S}_\xi$, it can been shown that such a function satisfies HEB with $\theta = 1/2$ on any sublevel set $\mathcal{S}_\xi$. Examples of $h(\mathbf{u})$ include logistic loss $h(\mathbf{u}) = \sum_i \log(1 + \exp(-u_i))$.

Proposition 11 *(Necoara et al., 2015, Theorem 4.3) Suppose the optimal set of (12) is non-empty and compact, $g(\mathbf{x})$ is a polyhedral function and $h(\cdot)$ is a smooth and strongly convex function on any compact set. Then $F(\mathbf{x})$ satisfies the HEB on any sublevel set $\mathcal{S}_\xi$ with $\theta = 1/2$ for $\xi > 0$.*

Finally, we note that there exist problems that admit HEB with $\theta > 1/2$. A trivial example is given by $F(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2 + \|\mathbf{x}\|_p^p$ with $p \in [1, 2)$, which satisfies HEB with $\theta = 1/p \in (1/2, 1]$. An interesting non-trivial family of problems is that $f(\mathbf{x}) = 0$ and $g(\mathbf{x})$ is a piece-wise linear functions according to Proposition 9. PG or APG applied to such family of problems is closely related to proximal point algorithm (Rockafellar, 1976). Explorations of such algorithmic connection is not the focus of this paper.

4. PG and restarting APG under HEB

As a warm-up and motivation of the major contribution presented in next section, we present a convergence result of PG and a restarting APG under the HEB condition. We first present a result of PG as shown in Algorithm 2.

Theorem 1 *Suppose $F(\mathbf{x}_0) - F_* \leq \epsilon_0$ and $F(\mathbf{x})$ satisfies HEB on $\mathcal{S}_{\epsilon_0}$. The iteration complexity of PG (with option I) for achieving $F(\mathbf{x}_t) - F_* \leq \epsilon$ is $O(c^2 L \epsilon_0^{2\theta-1})$ if $\theta > 1/2$, and is $O(\max\{\frac{c^2 L}{\epsilon^{1-2\theta}}, c^2 L \log(\frac{\epsilon_0}{\epsilon})\})$ if $\theta \leq 1/2$.*

Algorithm 2 PG

1: **Input**:$\mathbf{x}_0 \in \Omega$
2: **for** $\tau = 1, \ldots, t$ **do**
3: $\quad \mathbf{x}_{\tau+1} = P_{g/L}(\mathbf{x}_\tau - \nabla f(\mathbf{x}_\tau)/L)$
4: **end for**
5: Option I: return $\mathbf{x}_{t+1}$
6: Option II: return $\mathbf{x}_k$ s.t. $G(\mathbf{x}_k) = \min_\tau \|G(\mathbf{x}_\tau)\|_2$

Proof Divide the whole FOR loop of the Algorithm 2 into K stages, denote t_k by the number of iterations in the k-th stage, and denote $\mathbf{x}_k$ by the updated $\mathbf{x}$ at the end of the k-th stage, where $k = 1, \ldots K$. Define $\epsilon_k := \frac{\epsilon_0}{2^k}$.

Choose $t_k = \lceil c^2 L \epsilon_{k-1}^{2\theta-1} \rceil$, and we will prove $F(\mathbf{x}_k) - F_* \leq \epsilon_k$ by induction. Suppose $F(\mathbf{x}_{k-1}) - F_* \leq \epsilon_{k-1}$, we have $\mathbf{x}_{k-1} \in \mathcal{S}_{\epsilon_0}$. According to Proposition 3, at the k-th stage, we have

$$F(\mathbf{x}_k) - F_* \leq \frac{L\|\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^*\|_2^2}{2t_k},$$

where $\mathbf{x}_{k-1}^* \in \Omega_*$, the closest point to $\mathbf{x}_{k-1}$ in the optimal set. By the HEB condition, we have

$$F(\mathbf{x}_k) - F_* \leq \frac{c^2 L \epsilon_{k-1}^{2\theta}}{2t_k}.$$

Since $t_k \geq c^2 L \epsilon_{k-1}^{2\theta-1}$, we have $F(\mathbf{x}_k) - F_* \leq \epsilon_k$. The total number of iterations is

$$\sum_{k=1}^{K} t_k \leq O(c^2 L \sum_{k=1}^{K} \epsilon_{k-1}^{2\theta-1}).$$

From the above analysis, we see that after each stage, the optimality gap decreases by half, so taking $K = \lceil \log_2 \frac{\epsilon_0}{\epsilon} \rceil$ guarantees $F(\mathbf{x}_k) - F_* \leq \epsilon$.

If $\theta > 1/2$, the iteration complexity is $O(c^2 L \epsilon_0^{2\theta-1})$. If $\theta = 1/2$, the iteration complexity is $O(c^2 L \log \frac{\epsilon_0}{\epsilon})$. If $\theta < 1/2$, the iteration complexity is

$$\sum_{k=1}^{K} t_k \leq O(c^2 L \sum_{k=1}^{K} (\frac{\epsilon_0}{2^{k-1}})^{2\theta-1}) = O(c^2 L/\epsilon^{1-2\theta}).$$

■

Next, we show that APG can be made adaptive to HEB by periodically restarting given c and θ. This is similar to (Necoara et al., 2015) under the QGC. The steps of restarting APG (rAPG) are presented in Algorithm 3, where we employ the simplest variant of APG.

Theorem 2 *Suppose $F(\mathbf{x}_0) - F_* \leq \epsilon_0$ and $F(\mathbf{x})$ satisfies HEB on $\mathcal{S}_{\epsilon_0}$. By running Algorithm 2 with $K = \lceil \log_2 \frac{\epsilon_0}{\epsilon} \rceil$ and $t_k = \lceil 2c\sqrt{L}\epsilon_{k-1}^{\theta-1/2} \rceil$, we have $F(\mathbf{x}_K) - F_* \leq \epsilon$. The iteration complexity of rAPG is $O(c\sqrt{L}\epsilon_0^{1/2-\theta})$ if $\theta > 1/2$, and if $\theta \leq 1/2$ it is $O(\max\{\frac{c\sqrt{L}}{\epsilon^{1/2-\theta}}, c\sqrt{L}\log(\frac{\epsilon_0}{\epsilon})\})$.*

Algorithm 3 restarting APG (rAPG)

Input: the number of stages K and $\mathbf{x}_0 \in \Omega$
for $k = 1, \ldots, K$ **do**
  Set $\mathbf{y}_1^k = \mathbf{x}_{k-1}$ and $\mathbf{x}_1^k = \mathbf{x}_{k-1}$
  for $\tau = 1, \ldots, t_k$ **do**
    Update $\mathbf{x}_{\tau+1}^k = P_{g/L}(\mathbf{y}_\tau^k - \nabla f(\mathbf{y}_\tau^k)/L)$
    Update $\mathbf{y}_{\tau+1}^k = \mathbf{x}_{\tau+1}^k + \frac{\tau}{\tau+3}(\mathbf{x}_{\tau+1}^k - \mathbf{x}_\tau^k)$
  end for
  Let $\mathbf{x}_k = \mathbf{x}_{t_k+1}^k$
  Update t_k
end for
Output: $\mathbf{x}_K$

Proof Similar to the proof of Theorem 1, we will prove by induction that $F(\mathbf{x}_k) - F_* \leq \epsilon_k \triangleq \frac{\epsilon_0}{2^k}$. Assume that $F(\mathbf{x}_{k-1}) - F_* \leq \epsilon_{k-1}$. Hence, $\mathbf{x}_{k-1} \in \mathcal{S}_{\epsilon_0}$. Then according to Proposition 4 and the HEB condition, we have

$$F(\mathbf{x}_k) - F_* \leq \frac{2c^2 L \epsilon_{k-1}^{2\theta}}{(t_k+1)^2}.$$

Since $t_k \geq 2c\sqrt{L}\epsilon_{k-1}^{\theta-1/2}$, we have

$$F(\mathbf{x}_k) - F_* \leq \frac{\epsilon_{k-1}}{2} = \epsilon_k.$$

After K stages, we have $F(\mathbf{x}_K) - F_* \leq \epsilon$. The total number of iterations is

$$T_K = \sum_{k=1}^{K} t_k \leq O(c\sqrt{L}\epsilon_{k-1}^{\theta-1/2}).$$

When $\theta > 1/2$, we have $T_K \leq O(c\sqrt{L}\epsilon_0^{\theta-1/2})$. When $\theta \leq 1/2$, we have

$$T_K \leq O\left(\max\{c\sqrt{L}\log(\epsilon_0/\epsilon), c\sqrt{L}/\epsilon^{1/2-\theta}\}\right).$$

■

From Algorithm 3, we can see that rAPG requires the knowledge of c besides θ to restart APG. However, for many problems of interest, the value of c is unknown, which makes rAPG impractical. To address this issue, we propose to use the magnitude of the proximal gradient as a measure for restart and termination. Previous work (Nesterov, 2004) have considered the strongly convex optimization problems where the strong convexity parameter is unknown, where they also use the magnitude of the proximal gradient as a measure for restart and termination. However, in order to achieve faster convergence under the HEB condition without the strong convexity, we have to introduce a novel technique of adaptive regularization that adapts to the HEB. With a novel synthesis of the adaptive regularization and a conditional restarting that searches for the c, we are able to develop practical adaptive accelerated gradient methods.

Before diving into the details of the proposed algorithm, we will first present a variant of PG as a baseline for comparison motivated by (Nesterov, 2012) for smooth problems, which enjoys a faster convergence than the vanilla PG in terms of the proximal gradient's norm. The idea is to return a solution that achieves the minimum magnitude of the proximal gradient (the option II in Algorithm 2). The convergence of $\min_{1\leq\tau\leq t}\|G(\mathbf{x}_\tau)\|_2$ under HEB is presented in the following theorem.

Theorem 3 *Suppose $F(\mathbf{x}_0) - F_* \leq \epsilon_0$ and $F(\mathbf{x})$ satisfies HEB on $\mathcal{S}_{\epsilon_0}$. The iteration complexity of PG (with option II) for achiving $\min_{1\leq\tau\leq t}\|G(\mathbf{x}_\tau)\|_2 \leq \epsilon$, is $O(c^{\frac{1}{1-\theta}} L \max\{1/\epsilon^{\frac{1-2\theta}{1-\theta}}, \log(\frac{\epsilon_0}{\epsilon})\})$ if $\theta \leq 1/2$, and is $O(c^2 L\epsilon_0^{2\theta-1})$ if $\theta > 1/2$.*

Proof By the update of Algorithm 2 with option II and Corollary 1, we have

$$F(\mathbf{x}_\tau) - F(\mathbf{x}_{\tau+1}) \geq \frac{1}{2L}\|G(\mathbf{x}_\tau)\|_2^2.$$

Let $t = 2j$. Summing over $\tau = j, \ldots, t$ gives

$$F(\mathbf{x}_j) - F(\mathbf{x}_{t+1}) \geq \frac{1}{2L}\sum_{\tau=j}^{t}\|G(\mathbf{x}_\tau)\|_2^2.$$

Since $\|G(\mathbf{x}_\tau)\|_2 \geq \min_{1\leq\tau\leq t}\|G(\mathbf{x}_\tau)\|_2$ and $F(\mathbf{x}_{t+1}) \geq F_*$, we have

$$\frac{j}{2L}\min_{1\leq\tau\leq t}\|G(\mathbf{x}_\tau)\|_2^2 \leq F(\mathbf{x}_j) - F_*.$$

Hence,

$$\min_{1\leq\tau\leq t}\|G(\mathbf{x}_\tau)\|_2^2 \leq \frac{2L}{j}(F(\mathbf{x}_j) - F_*). \tag{13}$$

We consider three scenarios of θ.

(I). If $\theta > 1/2$, according to Theorem 1, we know that $F(\mathbf{x}_j) - F_*$ converges to 0 in $j = O(c^2 L\epsilon_0^{2\theta-1})$ steps, so $\min_{1\leq\tau\leq t}\|G(\mathbf{x}_\tau)\|_2^2$ converges to 0 in $t = O(c^2 L\epsilon_0^{2\theta-1})$ steps.

(II). If $\theta = 1/2$, let $j = \max(k, 2L)$ and $t = 2j$, where $k = ac^2 L\log(\frac{\epsilon_0}{\epsilon^2})$, and a is a constant hided in the big O notation. According to Theorem 1, we have

$$F(\mathbf{x}_k) - F_* \leq \epsilon^2, \tag{14}$$

then the inequality (13), (14) and the choice of j, k yields

$$\min_{1\leq\tau\leq t}\|G(\mathbf{x}_\tau)\|_2^2 \leq \frac{2L}{j}(F(\mathbf{x}_j) - F_*) \leq \epsilon^2,$$

so we know that $t = O(c^2 L\log(\frac{\epsilon_0}{\epsilon}))$.

(III). If $\theta < 1/2$, let j be an index such that $F(\mathbf{x}_j) - F_* \leq \epsilon'$. We can set $j = 2ac^2L/{\epsilon'}^{1-2\theta}$ and hence $t = 4ac^2L/{\epsilon'}^{1-2\theta}$, and have

$$\min_{1\leq\tau\leq t} \|G(\mathbf{x}_\tau)\|_2^2 \leq \frac{2L}{j}(F(\mathbf{x}_j) - F_*) \leq \frac{\epsilon'{\epsilon'}^{1-2\theta}}{ac^2} = \frac{{\epsilon'}^{2-2\theta}}{ac^2}.$$

Let $\epsilon' = c^{\frac{1}{1-\theta}}\epsilon^{\frac{1}{(1-\theta)}}$, we have $\min_{1\leq\tau\leq t}\|G(\mathbf{x}_\tau)\|_2^2 \leq \epsilon^2/a$. We can conclude $t = O(c^{\frac{1}{1-\theta}}L/\epsilon^{\frac{1-2\theta}{1-\theta}})$.

By combining the three scenarios, we can complete the proof. ■

The final theorem in this section summarizes an $o(1/t)$ convergence result of PG for minimizing a proper, coercive, convex, lower semicontinuous and semi-algebraic function, which could be interesting of its own.

Theorem 4 *Let $F(\mathbf{x})$ be a proper, coercive, convex, lower semicontinuous and semi-algebraic functions. Then PG (with option I and option II) converges at a speed of $o(1/t)$ for $F(\mathbf{x}) - F_*$ and $G(\mathbf{x})$, respectively, where t is the total number of iterations.*

Remark: This can be easily proved by combining Proposition 7 and Theorems 1, 3.

5. Adaptive Accelerated Gradient Converging Methods for Smooth Optimization

In the following two sections, we will present adaptive accelerated gradient converging methods that are faster than minPG for the convergence of (proximal) gradient's norm. Due to its simplicity, we first consider the following unconstrained optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$$

where $f(\mathbf{x})$ is a L-smooth function. We abuse Ω_* to denote the optimal set of above problem. The lemma below that bounds the distance of a point to the optimal set by a function of the gradient's norm.

Lemma 1 *If $f(\mathbf{x})$ satisfies the HEB on $\mathbf{x} \in \mathcal{S}_\xi$ with $\theta \in (0, 1]$, i.e., there exists $c > 0$ such that for any $\mathbf{x} \in \mathcal{S}_\xi$ we have*

$$D(\mathbf{x}, \Omega_*) \leq c(f(\mathbf{x}) - f_*)^\theta.$$

If $\theta \in (0, 1)$, then for any $\mathbf{x} \in \mathcal{S}_\xi$

$$D(\mathbf{x}, \Omega_*) \leq c^{\frac{1}{1-\theta}}\|\partial f(\mathbf{x})\|_2^{\frac{\theta}{1-\theta}}.$$

If $\theta = 1$, then for any $\mathbf{x} \in \mathcal{S}_\xi$

$$D(\mathbf{x}, \Omega_*) \leq c^2\xi\|\partial f(\mathbf{x})\|_2.$$

The proof of this lemma is included in the Appendix.

Note that for a smooth function $f(\mathbf{x})$, we can restrict our discussion on HEB condition to $\theta \leq 1/2$. Since $f(\mathbf{x}) - f(\mathbf{x}_*) \leq \frac{L}{2}\|\mathbf{x} - \mathbf{x}_*\|_2^2$ where $\mathbf{x}_* \in \Omega_*$, plugging this equality into the HEB we can see θ has to be less than 1/2 if c remains a constant. In order to derive faster convergence than minPG, we employ the technique of regularization, i.e., adding a strongly convex regularizer into the objective. To this end, we define the following problem:

$$f_\delta(\mathbf{x}) = f(\mathbf{x}) + \frac{\delta}{2}\|\mathbf{x} - \mathbf{x}_0\|_2^2,$$

where $\mathbf{x}_0$ is the initial solution. It is clear that $f_\delta(\mathbf{x})$ is a $(L+\delta)$-smooth and δ-strongly convex function. The proposed adaAGC algorithm will run in multiple stages. At the k-th stage, we construct a problem like above using a value of δ_k and an initial solution $\mathbf{x}_{k-1}$, and employ APG for smooth and strongly convex minimization to solve the constructed problem until the gradient's norm is decreased by a factor of 2. The initial solution for each stage is the output solution of the previous stage and the value of δ will be adaptively decreasing based on θ in the HEB condition. Specifically, the choice of δ_k can be set in the following way:

$$\delta_k = \frac{\epsilon_{k-1}^{\frac{1-2\theta}{1-\theta}}}{6c_e^{1/(1-\theta)}} \tag{15}$$

We also embed a search procedure for the value of c into the algorithm in order to leverage the HEB condition. The detailed steps of adaAGC for solving $\min_{\mathbf{x}} f(\mathbf{x})$ are presented in Algorithm 4 assuming $f(\mathbf{x})$ satisfies a HEB condition.

Below, we first present the analysis for each stage to pave the path of proof for our main theorem.

Theorem 5 *Suppose $f(\mathbf{x})$ is L-smooth. By running the update in (8) for solving $f_\delta(\mathbf{x}) = f(\mathbf{x}) + \frac{\delta}{2}\|\mathbf{x} - \mathbf{x}_0\|_2^2$ with $\beta = \frac{\sqrt{L+\delta}-\sqrt{\delta}}{\sqrt{L+\delta}+\sqrt{\delta}}$ and an initial solution $\mathbf{x}_0$, we have for any $\mathbf{x} \in \mathbb{R}^d$*

$$f_\delta(\mathbf{x}_{t+1}) - f_\delta(\mathbf{x}) \leq \left(1 - \sqrt{\frac{\delta}{L+\delta}}\right)^t [f(\mathbf{x}_0) - f(\mathbf{x})],$$

and $f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_0)$. If $t \geq \sqrt{\frac{L+\delta}{\delta}}\log\left(\frac{L}{\delta}\right)$, we have

$$\|\mathbf{x}_{t+1} - \mathbf{x}_0\|_2 \leq \sqrt{2}\|\mathbf{x}_0 - \mathbf{x}_*\|_2.$$

Proof By Proposition 5, we have

$$f_\delta(\mathbf{x}_{t+1}) - f_\delta(\mathbf{x}) \leq \left(1 - \sqrt{\frac{\delta}{L+\delta}}\right)^t \left[f_\delta(\mathbf{x}_0) - f_\delta(\mathbf{x}) + \frac{\delta}{2}\|\mathbf{x} - \mathbf{x}_0\|_2^2\right].$$

By noting the definition of $f_\delta(\mathbf{x})$ we can prove the first inequality. To prove the second inequality, we let $\mathbf{x} = \mathbf{x}_0$ in the first inequality, we have

$$f(\mathbf{x}_{t+1}) + \frac{\delta}{2}\|\mathbf{x}_{t+1} - \mathbf{x}_0\|_2^2 - f(\mathbf{x}_0) \leq 0.$$

Algorithm 4 adaAGC for solving (1)

Input: $\mathbf{x}_0 \in \Omega$ and c_0 and $\gamma > 1$
Let $c_e = c_0$ and $\epsilon_0 = \|\nabla f(\mathbf{x}_0)\|_2$,
for $k = 1, \ldots, K$ **do**
  Let δ_k be given in (15) and $f_{\delta_k}(\mathbf{x}) = f(\mathbf{x}) + \frac{\delta_k}{2}\|\mathbf{x} - \mathbf{x}_{k-1}\|_2^2$
  $\mathbf{x}_1^k = \mathbf{x}_{k-1}$ and $\mathbf{y}_1 = \mathbf{x}_{k-1}$
  for $s = 1, \ldots$ **do**
    for $\tau = 1, \ldots$ **do**
      $\mathbf{x}_{\tau+1}^k = \mathbf{y}_\tau - \frac{1}{L+\delta_k}\nabla f_{\delta_k}(\mathbf{y}_\tau)$
      $\mathbf{y}_{\tau+1} = \mathbf{x}_{\tau+1}^k + \frac{\sqrt{L+\delta_k} - \sqrt{\delta_k}}{\sqrt{L+\delta_k} + \sqrt{\delta_k}}(\mathbf{x}_{\tau+1}^k - \mathbf{x}_\tau^k)$
      if $\|\nabla f(\mathbf{x}_{\tau+1}^k)\|_2 \leq \epsilon_{k-1}/2$ **then**
        let $\mathbf{x}_k = \mathbf{x}_{\tau+1}^k$ 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: $\mathbf{x}_K$

Thus $f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_0)$. To prove the third inequality, we let $\mathbf{x} = \mathbf{x}_* \in \Omega_*$ in the first inequality, we have

$$f(\mathbf{x}_{t+1}) + \frac{\delta}{2}\|\mathbf{x}_{t+1} - \mathbf{x}_0\|_2^2 - f(\mathbf{x}_*) - \frac{\delta}{2}\|\mathbf{x}_0 - \mathbf{x}_*\|_2^2 \leq \left(1 - \sqrt{\frac{\delta}{L+\delta}}\right)^t [f(\mathbf{x}_0) - f(\mathbf{x}_*)] .$$

Then we have

$$\|\mathbf{x}_{t+1} - \mathbf{x}_0\|_2^2 \leq \|\mathbf{x}_0 - \mathbf{x}_*\|_2^2 + \left(1 - \sqrt{\frac{\delta}{L+\delta}}\right)^t \frac{L}{\delta}\|\mathbf{x}_0 - \mathbf{x}_*\|_2^2.$$

If $t \geq \sqrt{\frac{L+\delta}{\delta}} \log\left(\frac{L}{\delta}\right)$, we have

$$\begin{aligned}\|\mathbf{x}_{t+1} - \mathbf{x}_0\|_2^2 &\leq \|\mathbf{x}_0 - \mathbf{x}_*\|_2^2 + \left(1 - \sqrt{\frac{\delta}{L+\delta}}\right)^t \frac{L}{\delta}\|\mathbf{x}_0 - \mathbf{x}_*\|_2^2 \\ &\leq \|\mathbf{x}_0 - \mathbf{x}_*\|_2^2 + \|\mathbf{x}_0 - \mathbf{x}_*\|_2^2 = 2\|\mathbf{x}_0 - \mathbf{x}_*\|_2^2.\end{aligned}$$

■

Next, we prove the following theorem.

Theorem 6 *Under the same condition as in Theorem 5, we have*

$$\|\nabla f(\mathbf{x}_{t+1})\|_2 \leq \sqrt{L(L+\delta)}\left(1-\sqrt{\frac{\delta}{L+\delta}}\right)^{t/2}\|\mathbf{x}_0-\mathbf{x}_*\|_2 + \sqrt{2}\delta\|\mathbf{x}_0-\mathbf{x}_*\|_2.$$

Proof Let $\mathbf{x}_{t+1}^+ = \mathbf{x}_{t+1} - \frac{1}{L+\delta}\nabla f_\delta(\mathbf{x}_{t+1})$ in the first inequality in Theorem 5, we have

$$f_\delta(\mathbf{x}_{t+1}) - f_\delta(\mathbf{x}_{t+1}^+) \leq \left(1-\sqrt{\frac{\delta}{L+\delta}}\right)^t [f(\mathbf{x}_0) - f(\mathbf{x}_{t+1}^+)] \leq \left(1-\sqrt{\frac{\delta}{L+\delta}}\right)^t [f(\mathbf{x}_0) - f(\mathbf{x}_*)],$$

where the last inequality uses $f(\mathbf{x}_{t+1}^+) \geq f(\mathbf{x}_*)$. Applying Corollary 1 we have

$$\frac{1}{2(L+\delta)}\|\nabla f_\delta(\mathbf{x}_{t+1})\|_2^2 \leq f_\delta(\mathbf{x}_{t+1}) - f_\delta(\mathbf{x}_{t+1}^+) \leq \left(1-\sqrt{\frac{\delta}{L+\delta}}\right)^t [f(\mathbf{x}_0) - f(\mathbf{x}_*)].$$

Then we have

$$\begin{aligned}\|\nabla f_\delta(\mathbf{x}_{t+1})\|_2 &\leq \sqrt{2(L+\delta)}\left(1-\sqrt{\frac{\delta}{L+\delta}}\right)^{t/2}\sqrt{f(\mathbf{x}_0) - f(\mathbf{x}_*)} \\ &\leq \sqrt{L(L+\delta)}\left(1-\sqrt{\frac{\delta}{L+\delta}}\right)^{t/2}\|\mathbf{x}_0-\mathbf{x}_*\|_2,\end{aligned}$$

where the last inequality uses the smoothness of $f(\mathbf{x})$. To proceed, we have

$$\begin{aligned}\|\nabla f(\mathbf{x}_{t+1})\|_2 = \|\nabla f_\delta(\mathbf{x}_{t+1}) - \delta(\mathbf{x}_{t+1}-\mathbf{x}_0)\|_2 &\leq \|\nabla f_\delta(\mathbf{x}_{t+1})\|_2 + \delta\|\mathbf{x}_{t+1}-\mathbf{x}_0\|_2 \\ &\leq \|\nabla f_\delta(\mathbf{x}_{t+1})\|_2 + \sqrt{2}\delta\|\mathbf{x}_0-\mathbf{x}_*\|_2.\end{aligned}$$

■

Finally, we can prove the main theorem of this section.

Theorem 7 *Suppose $f(\mathbf{x}_0) - f_* \leq \xi_0$, $f(\mathbf{x})$ satisfies HEB on $\mathcal{S}_{\xi_0}$ with $\theta \in (0,1]$ and $c_0 \leq c$. Let $\epsilon_0 = \|\nabla f(\mathbf{x}_0)\|_2$ and $K = \lceil \log_2(\frac{\epsilon_0}{\epsilon})\rceil$, $p = (1-2\theta)/(1-\theta)$ for $\theta \in (0, 1/2]$. The iteration complexity of the Algorithm 4 for having $\|\nabla f(\mathbf{x}_K)\|_2 \leq \epsilon$ is $\widetilde{O}\left(\sqrt{L}c^{\frac{1}{2(1-\theta)}}\max(\frac{1}{\epsilon^{p/2}}, \log(\varepsilon_0/\epsilon)\right)$, where $\widetilde{O}(\cdot)$ suppresses a log term depending on c, c_0, L, γ.*

Proof

We can easily induce that $f(\mathbf{x}_k) - f_* \leq \xi_0$ from Theorem 5. Let $t_k = \lceil 2\sqrt{\frac{L+\delta_k}{\delta_k}}\log\frac{\sqrt{L(L+\delta_k)}}{\delta_k}\rceil$. Applying Theorem 6 to the k-the stage of adaAGC, we have

$$\begin{aligned}&\|\nabla f(\mathbf{x}_{t_k+1}^k)\|_2 \leq \sqrt{L(L+\delta_k)}\left(1-\sqrt{\frac{\delta_k}{L+\delta_k}}\right)^{t_k/2}\|\mathbf{x}_{k-1}-\mathbf{x}_*\|_2 + \sqrt{2}\delta_k\|\mathbf{x}_{k-1}-\mathbf{x}_*\|_2 \\ &\leq \sqrt{L(L+\delta_k)}\left(1-\sqrt{\frac{\delta_k}{L+\delta_k}}\right)^{t_k/2} c^{\frac{1}{(1-\theta)}}\|\nabla f(\mathbf{x}_{k-1})\|_2^{\frac{\theta}{(1-\theta)}} + \sqrt{2}\delta_k c^{\frac{1}{(1-\theta)}}\|\nabla f(\mathbf{x}_{k-1})\|_2^{\frac{\theta}{(1-\theta)}},\end{aligned}$$

where the last inequality follows Lemma 1. Note that at each stage, we check two conditions (i) $\|\nabla f(\mathbf{x}_{\tau+1}^k)\|_2 \leq \epsilon_{k-1}/2$ and (ii) $\tau = t_k$. If the first condition satisfies first, we proceed to the next stage. If the second condition satisfies first, then we can claim that $c_e \leq c$ and then we increase c_e by a factor $\gamma > 1$ and then restart the same stage. To verify the claim, assume $c_e > c$ and the second condition satisfies first, i.e., $\tau = t_k$ but $\|\nabla f(\mathbf{x}_{\tau+1}^k)\|_2 > \epsilon_{k-1}/2$. We will deduce a contradiction. To this end, we use

$$\begin{aligned}\|\nabla f(\mathbf{x}_{t_k+1}^k)\|_2 &\leq \left(\sqrt{L(L+\delta_k)}\left(1-\sqrt{\frac{\delta_k}{L+\delta_k}}\right)^{t_k/2} + \sqrt{2}\delta_k\right) c^{\frac{1}{1-\theta}} \|\nabla f(\mathbf{x}_{k-1})\|_2^{\frac{\theta}{1-\theta}} \\ &\leq (\delta_k + \sqrt{2}\delta_k) c^{\frac{1}{1-\theta}} \|\nabla f(\mathbf{x}_{k-1})\|_2^{\frac{\theta}{1-\theta}} \\ &\leq 3\delta_k c^{\frac{1}{1-\theta}} \|\nabla f(\mathbf{x}_{k-1})\|_2^{\frac{\theta}{1-\theta}} = \frac{3\epsilon_{k-1}^{\frac{1-2\theta}{1-\theta}}}{6c_e^{1/(1-\theta)}} c^{\frac{1}{1-\theta}} \epsilon_{k-1}^{\frac{\theta}{1-\theta}} \leq \epsilon_{k-1}/2,\end{aligned}$$

where the last inequality follows that $c_e > c$. This contradicts to the assumption that $\|\nabla f(\mathbf{x}_{\tau+1}^k)\|_2 > \epsilon_{k-1}/2$, which verifies our claim.

Since c_e is increased by a factor $\gamma > 1$ whenever condition (ii) holds first. Thus with at most $\lceil \log_\gamma(c/c_0) \rceil$ times condition (ii) holds first. Similarly with at most $\lceil \log_2 \epsilon_0/\epsilon \rceil$ times that condition (i) holds first before the algorithm terminates. We let T_k denote the total number of iterations in order to make condition (i) satisfies in stage k. First, we can see that $c_e \leq \gamma c$.

Let $\delta_k' = \frac{\epsilon_{k-1}^{\frac{1-2\theta}{1-\theta}}}{6(\gamma c)^{1/(1-\theta)}}$ and $t_k' = \lceil 2\sqrt{\frac{L+\delta_k'}{\delta_k'}} \log \frac{\sqrt{L(L+\delta_k')}}{\delta_k'} \rceil$. Let s_k denote the number of cycles in each stage in order to have $\|\nabla f(\mathbf{x}_{\tau+1}^k)\|_2 \leq \epsilon_k$. Then $s_k \leq \log_\gamma(c/c_0) + 1$. The total number of iterations of across all stages is bounded by $\sum_{k=1}^K s_k t_k$, which is bounded by

$$\sum_{k=1}^K s_k t_k \leq (1 + \log_\gamma(c/c_0)) \sum_{k=1}^K t_k'.$$

Plugging the value of t_k', we can deduce the iteration complexity in Theorem 7 for $\theta \in (0, 1/2]$.

■

6. Adaptive Accelerated Gradient Converging Methods for Smooth Composite Optimization

In this section, we generalize the results in previous section to smooth composite optimization problem

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

Different from last section, we will use the proximal gradient $G(\mathbf{x}_t)$ as a measure for restart and termination in adaAGC. Similar to last section, we first present a key lemma for our development that serves the foundation of the adaptive regularization and conditional restarting.

Lemma 2 *Assume $F(\mathbf{x})$ satisfies HEB for any $\mathbf{x} \in \mathcal{S}_\xi$ with $\theta \in (0, 1]$. If $\theta \in (0, 1/2]$ then we have for any $\mathbf{x} \in \mathcal{S}_\xi$*

$$D(\mathbf{x}, \Omega_*) \leq \frac{2}{L}\|G(\mathbf{x})\|_2 + c^{\frac{1}{1-\theta}} 2^{\frac{\theta}{1-\theta}} \|G(\mathbf{x})\|_2^{\frac{\theta}{1-\theta}}.$$

If $\theta \in (1/2, 1]$, we have for any $\mathbf{x} \in \mathcal{S}_\xi$

$$D(\mathbf{x}, \Omega_*) \leq \left(\frac{2}{L} + 2c^2\xi^{2\theta-1}\right)\|G(\mathbf{x})\|_2.$$

Proof The conclusion is trivial when $\mathbf{x} \in \Omega_*$, so we only need to consider the case when $\mathbf{x} \notin \Omega_*$. Define $P_{\eta F}(\mathbf{x}) = \arg\min_{\mathbf{u}} \frac{1}{2}\|\mathbf{u} - \mathbf{x}\|_2^2 + \eta F(\mathbf{u})$.

We first prove for $\theta \in (0, 1/2]$. It is not difficult to see that $\frac{1}{\eta}(\mathbf{x} - P_{\eta F}(\mathbf{x})) \in \partial F(P_{\eta F}(\mathbf{x}))$.

$$\begin{aligned}
D(\mathbf{x}, \Omega_*) &\leq \|\mathbf{x} - P_{\eta F}(\mathbf{x})\|_2 + D(P_{\eta F}(\mathbf{x}), \Omega_*) \\
&\leq \|\mathbf{x} - P_{\eta F}(\mathbf{x})\|_2 + c^{\frac{1}{1-\theta}} \|\partial F(P_{\eta F}(\mathbf{x}))\|_2^{\frac{\theta}{1-\theta}} \\
&\leq \|\mathbf{x} - P_{\eta F}(\mathbf{x})\|_2 + \frac{c^{\frac{1}{1-\theta}}}{\eta^{\frac{\theta}{1-\theta}}} \|\mathbf{x} - P_{\eta F}(\mathbf{x})\|_2^{\frac{\theta}{1-\theta}} \\
&\leq \eta(1 + L\eta)\|G_\eta(\mathbf{x})\|_2 + c^{\frac{1}{1-\theta}}(1 + \eta L)^{\frac{\theta}{1-\theta}} \|G_\eta(\mathbf{x})\|_2^{\frac{\theta}{1-\theta}},
\end{aligned}$$

where the second inequality uses the result in Lemma 1 and the last inequality follows Proposition 13, which asserts that $\|\mathbf{x} - P_{\eta F}(\mathbf{x})\|_2 \leq \eta(1 + L\eta)\|G_\eta(\mathbf{x})\|_2$. Plugging the value $\eta = 1/L$, we have the result.

Next, we prove for $\theta \in (1/2, 1]$. For any $\mathbf{x} \in S_\xi$, we have $P_{\eta F}(\mathbf{x}) \in S_\xi$ and

$$\begin{aligned}
&D(P_{\eta F}(\mathbf{x}), \Omega_*) \leq c(F(P_{\eta F}(\mathbf{x})) - F_*)^\theta \\
&= c(F(P_{\eta F}(\mathbf{x})) - F_*)^{1-\theta}(F(P_{\eta F}(\mathbf{x})) - F_*)^{2\theta-1} \leq c^2\|\partial F(P_{\eta F}(\mathbf{x}))\|_2(F(\mathbf{x}) - F_*)^{2\theta-1} \\
&\leq c^2\|\partial F(P_{\eta F}(\mathbf{x}))\|_2\xi^{2\theta-1} \leq c^2(1 + L\eta)\|G_\eta(\mathbf{x})\|_2\xi^{2\theta-1} \leq 2c^2\xi^{2\theta-1}\|G_\eta(\mathbf{x})\|_2,
\end{aligned}$$

where the second inequality holds because the inequality (24) holds for any $\theta \in (0, 1]$ (by Lemma 1), $F(P_{\eta F}(\mathbf{x})) \leq F(\mathbf{x}) \leq \xi$, the fourth inequality holds since $\|G_\eta(\mathbf{x})\|_2 \geq \frac{1}{1+L\eta}\|(\mathbf{x} - P_{\eta F}(\mathbf{x}))/\eta\|_2 \geq \frac{1}{1+L\eta}\|\partial F(P_{\eta F}(\mathbf{x}))\|_2$ (by Proposition 13), and the last inequality holds by taking $\eta = 1/L$.

So for $\theta \in (1/2, 1]$ and $\eta = 1/L$, we have

$$\begin{aligned}
D(\mathbf{x}, \Omega_*) &\leq \|\mathbf{x} - P_{\eta F}(\mathbf{x})\|_2 + D(P_{\eta F}(\mathbf{x}), \Omega_*) \\
&\leq (\frac{2}{L} + 2c^2\xi^{2\theta-1})\|G(\mathbf{x})\|_2.
\end{aligned}$$

■

A building block of the proposed algorithm is to solve a problem of the following style:

$$F_\delta(\mathbf{x}) = F(\mathbf{x}) + \frac{\delta}{2}\|\mathbf{x} - \mathbf{x}_0\|_2^2, \tag{16}$$

which consists of a L-smooth function $f(\mathbf{x})$ and a δ-strongly convex function $g_\delta(\mathbf{x}) = g(\mathbf{x}) + \frac{\delta}{2}\|\mathbf{x}-\mathbf{x}_0\|_2^2$. We present some technical results for employing the Algorithm 1 (i.e., Nesterov's ADG) to solve the above problem.

Theorem 8 *By running the Algorithm 1 for minimizing $f(\mathbf{x})+g_\delta(\mathbf{x})$ with an initial solution $\mathbf{x}_0$, then for any $\mathbf{x} \in \mathbb{R}^d$ and $t \geq 0$,*

$$F_\delta(\mathbf{x}_{t+1}) - F_\delta(\mathbf{x}) \leq \frac{L}{2}\|\mathbf{x}_0 - \mathbf{x}\|_2^2 \left[1 + \sqrt{\frac{\delta}{2L}}\right]^{-2t},$$

and $F(\mathbf{x}_{t+1}) \leq F(\mathbf{x}_0)$. If $t \geq \sqrt{\frac{L}{2\delta}} \log\left(\frac{L}{\delta}\right)$, we have $\|\mathbf{x}_{t+1} - \mathbf{x}_0\|_2 \leq \sqrt{2}\|\mathbf{x}_0 - \mathbf{x}_*\|_2$.

Proof Applying Proposition 6 to $F_\delta(\mathbf{x})$ yields

$$\begin{aligned} F(\mathbf{x}_{t+1}) - F(\mathbf{x}) + \frac{\delta}{2}\|\mathbf{x}_{t+1} - \mathbf{x}_0\|_2^2 &\leq \frac{\delta}{2}\|\mathbf{x} - \mathbf{x}_0\|_2^2 \\ &+ \frac{L}{2}\|\mathbf{x}_0 - \mathbf{x}\|_2^2 \left[1 + \sqrt{\frac{\delta}{2L}}\right]^{-2t}. \end{aligned} \tag{17}$$

Then $F(\mathbf{x}_{t+1}) - F(\mathbf{x}_0) \leq 0$, and choose $\mathbf{x} = \mathbf{x}_*$ in the inequality (17), where $\mathbf{x}_* \in \Omega_*$, then we have

$$\begin{aligned} &\|\mathbf{x}_{t+1} - \mathbf{x}_0\|_2^2 \\ &\leq \|\mathbf{x}_0 - \mathbf{x}_*\|_2^2 + \frac{L}{\delta}\|\mathbf{x}_0 - \mathbf{x}_*\|_2^2 \left[1 + \sqrt{\frac{\delta}{2L}}\right]^{-2t}. \end{aligned}$$

Under the condition $t \geq \sqrt{\frac{L}{2\delta}} \log\left(\frac{L}{\delta}\right)$ we have $\|\mathbf{x}_{t+1} - \mathbf{x}_0\|_2 \leq \sqrt{2}\|\mathbf{x}_0 - \mathbf{x}_*\|_2$. ■

Theorem 9 *Under the same condition as in Theorem 8, for $t \geq \sqrt{\frac{L}{2\delta}} \log\left(\frac{L}{\delta}\right)$ we have*

$$\begin{aligned} \|G(\mathbf{x}_{t+1})\|_2 \leq &\sqrt{L(L+\delta)}\|\mathbf{x}_0 - \mathbf{x}_*\|_2 \left[1 + \sqrt{\delta/(2L)}\right]^{-t} \\ &+ 2\sqrt{2}\delta\|\mathbf{x}_* - \mathbf{x}_0\|_2. \end{aligned}$$

Proof Let $\mathbf{x}_\delta^*$ be the optimal solution to $\min_{\mathbf{x}\in\mathbb{R}^d} F_\delta(\mathbf{x})$ and $\mathbf{x}_*$ denote an optimal solution to $\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x})$. Thanks to the strong convexity of $F_\delta(\mathbf{x})$, we have $F_\delta(\mathbf{x}_*) - F_\delta(\mathbf{x}_\delta^*) \geq \frac{\delta}{2}\|\mathbf{x}_* - \mathbf{x}_\delta^*\|_2^2$. Then

$$F(\mathbf{x}_*) - F(\mathbf{x}_\delta^*) + \delta/2\|\mathbf{x}_* - \mathbf{x}_0\|_2^2 - \delta/2\|\mathbf{x}_\delta^* - \mathbf{x}_0\|_2^2 \geq \delta/2\|\mathbf{x}_* - \mathbf{x}_\delta^*\|_2^2.$$

Since $F(\mathbf{x}_*) - F(\mathbf{x}_\delta^*) \leq 0$, it implies $\|\mathbf{x}_\delta^* - \mathbf{x}_0\|_2 \leq \|\mathbf{x}_* - \mathbf{x}_0\|_2$. By Corollary 1, we have

$$\frac{\eta}{2}\|G_\eta^\delta(\mathbf{x}_{t+1})\|_2^2 \leq F_\delta(\mathbf{x}_{t+1}) - F_\delta(\mathbf{x}_\delta^*) \leq \frac{L}{2}\|\mathbf{x}_0 - \mathbf{x}_\delta^*\|_2^2\left[1+\sqrt{\delta/(2L)}\right]^{-2t},$$

where $\eta \leq 1/(L+\delta)$ and G_η^δ is a proximal gradient of $F_\delta(\mathbf{x})$ defined as $G_\eta^\delta(\mathbf{x}) = \frac{1}{\eta}\left(\mathbf{x} - \mathbf{x}_\eta^+(\delta)\right)$ and

$$\mathbf{x}_\eta^+(\delta) = \arg\min_{\mathbf{y}}\left\{\eta(\nabla f(\mathbf{x}) + \delta(\mathbf{x}-\mathbf{x}_0))^\top(\mathbf{y}-\mathbf{x}) + \eta g(\mathbf{y}) + \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2\right\}.$$

Recall that $\mathbf{x}_\eta^+ = P_{\eta g}(\mathbf{x} - \eta\nabla f(\mathbf{x}))$. It is not difficult to derive that $\|\mathbf{x}_\eta^+ - \mathbf{x}_\eta^+(\delta)\|_2 \leq 2\eta\delta\|\mathbf{x}-\mathbf{x}_0\|_2$ (by Lemma 3 in the appendix). Since $G_\eta(\mathbf{x}) = \frac{1}{\eta}(\mathbf{x} - \mathbf{x}_\eta^+)$, we have

$$\|G_\eta(\mathbf{x})\|_2 \leq \|G_\eta^\delta(\mathbf{x})\|_2 + \|\mathbf{x}_\eta^+ - \mathbf{x}_\eta^+(\delta)\|_2/\eta \leq \|G_\eta^\delta(\mathbf{x})\|_2 + 2\delta\|\mathbf{x}-\mathbf{x}_0\|_2.$$

Let $\eta = 1/(L+\delta)$, we have

$$\begin{aligned}\|G_\eta(\mathbf{x}_{t+1})\|_2 &\leq 2\delta\|\mathbf{x}_{t+1}-\mathbf{x}_0\|_2 + \sqrt{L/\eta}\|\mathbf{x}_0 - \mathbf{x}_\delta^*\|_2\left[1+\sqrt{\delta/(2L)}\right]^{-t} \\ &\leq 2\sqrt{2}\delta\|\mathbf{x}_* - \mathbf{x}_0\|_2 + \sqrt{L(L+\delta)}\|\mathbf{x}_0 - \mathbf{x}_*\|_2\left[1+\sqrt{\delta/(2L)}\right]^{-t}.\end{aligned}$$

where we use the inequality $\|\mathbf{x}_\delta^* - \mathbf{x}_0\|_2 \leq \|\mathbf{x}_* - \mathbf{x}_0\|_2$. Since $\|G_\eta(\mathbf{x})\|_2$ is a monotonically decreasing function of η (by the Proposition 1) , then $\|G(\mathbf{x})\|_2 \leq \|G_\eta(\mathbf{x})\|_2$ for $\eta = 1/(L+\delta) \leq 1/L$. Then

$$\|G(\mathbf{x}_{t+1})\|_2 \leq \sqrt{L(L+\delta)}\|\mathbf{x}_0 - \mathbf{x}_*\|_2\left[1+\sqrt{\delta/(2L)}\right]^{-t} + 2\sqrt{2}\delta\|\mathbf{x}_* - \mathbf{x}_0\|_2.$$

■

Finally, we present the proposed adaptive accelerated gradient converging (adaAGC) method for solving the smooth composite optimization in Algorithm 5 and prove the main theorem of this section. The adaAGC runs with multiple stages ($k = 1, \ldots, K$). We start with an initial guess c_0 of the parameter c in the HEB. With the current guess c_e of c, at the k-th stage adaAGC employs ADG to solve a problem of (16) with an adaptive regularization parameter δ_k being

$$\delta_k = \begin{cases} \min\left(\frac{L}{32}, \frac{\varepsilon_{k-1}^{\frac{1-2\theta}{1-\theta}}}{16c_e^{1/(1-\theta)}2^{\frac{\theta}{1-\theta}}}\right) & \text{if } \theta \in (0, 1/2] \\ \min\left(\frac{L}{32}, \frac{1}{32c_e^2\epsilon_0^{2\theta-1}}\right) & \text{if } \theta \in (1/2, 1] \end{cases} \tag{18}$$

The step 16 specifies the condition for restarting with an increased value of c_e. When the flow enters step 17 before step 14 for each s, it means that the current guess c_e is not sufficiently large, then we increase c_e and repeat the same process (next iteration for s). We refer to this machinery as conditional restarting.

Algorithm 5 adaAGC for solving (2)

Input: $\mathbf{x}_0 \in \Omega$ and c_0 and $\gamma > 1$
Let $c_e = c_0$ and $\varepsilon_0 = \|G(\mathbf{x}_0)\|_2$,
for $k = 1, \ldots, K$ **do**
  for $s = 1, \ldots,$ **do**
    Let δ_k be given in (18) and $g_{\delta_k}(\mathbf{x}) = g(\mathbf{x}) + \frac{\delta_k}{2}\|\mathbf{x} - \mathbf{x}_{k-1}\|_2^2$
    $A_0 = 0$, $\mathbf{v}_0 = \mathbf{x}_{k-1}$, $\mathbf{x}_0^k = \mathbf{x}_{k-1}$
    for $t = 0, \ldots$ **do**
      Let a_{t+1} be the root of $\frac{a^2}{A_t + a} = 2\frac{1+\delta_k A_t}{L}$
      Set $A_{t+1} = A_t + a_{t+1}$
      Set $\mathbf{y}_t = \frac{A_t}{A_{t+1}}\mathbf{x}_t^k + \frac{a_{t+1}}{A_{t+1}}\mathbf{v}_t$
      Compute $\mathbf{x}_{t+1}^k = P_{g_{\delta_k}/L}(\mathbf{y}_t - \nabla f(\mathbf{y}_t)/L)$
      Compute $\mathbf{v}_{t+1} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{x}_{k-1}\|_2^2 + \sum_{\tau=1}^{t+1} a_\tau \nabla f(\mathbf{x}_\tau^k)^\top \mathbf{x} + A_{t+1} g_{\delta_k}(\mathbf{x})$
      if $\|G(\mathbf{x}_{t+1}^k)\|_2 \le \varepsilon_{k-1}/2$ **then**
        let $\mathbf{x}_k = \mathbf{x}_{t+1}^k$ and $\varepsilon_k = \varepsilon_{k-1}/2$.
        break the enclosing two for loops
      else if $\tau = \lceil \sqrt{\frac{2L}{\delta_k}} \log \frac{\sqrt{L(L+\delta_k)}}{\delta_k} \rceil$ **then**
        let $c_e = \gamma c_e$ and break the enclosing for loop
      end if
    end for
  end for
end for
Output: $\mathbf{x}_K$

Theorem 10 *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 iteration complexity of Algorithm 5 for having $\|G(\mathbf{x}_K)\|_2 \le \epsilon$ is $\widetilde{O}\left(\sqrt{L}c^{\frac{1}{2(1-\theta)}} \max(\frac{1}{\epsilon^{p/2}}, \log(\varepsilon_0/\epsilon)\right)$ if $\theta \in (0, 1/2]$, and $\widetilde{O}(\sqrt{L}c\epsilon_0^{\theta-1/2})$ if $\theta \in (1/2, 1]$, where $\widetilde{O}(\cdot)$ suppresses a log term depending on c, c_0, L, γ.*

Proof

- We first prove the case when $\theta \in (0, 1/2]$. We can easily induce that $F(\mathbf{x}_k) - F_* \le \epsilon_0$ from Theorem 8. Let $t_k = \lceil \sqrt{\frac{2L}{\delta_k}} \log \frac{\sqrt{L(L+\delta_k)}}{\delta_k} \rceil$. Applying Theorem 9 to the k-the stage of adaAGC and using Lemma 2, we have

$$\begin{aligned} \|G(\mathbf{x}_{t_k+1}^k)\|_2 \le &(\sqrt{L(L+\delta_k)} \left[1 + \sqrt{\frac{\delta_k}{2L}}\right]^{-t_k} + 2\sqrt{2}\delta_k) \\ &\times (\frac{2}{L}\|G(\mathbf{x}_{k-1})\|_2 + c^{\frac{1}{(1-\theta)}} 2^{\frac{\theta}{(1-\theta)}} \|G(\mathbf{x}_{k-1})\|_2^{\frac{\theta}{(1-\theta)}}), \end{aligned} \tag{19}$$

Note that at each stage, we check two conditions (i) $\|G(\mathbf{x}_{\tau+1}^k)\|_2 \le \varepsilon_{k-1}/2$ and (ii) $\tau = t_k$. If the first condition satisfies first, we proceed to the next stage (k increases by 1). If the second condition satisfies first, then we can claim that $c_e \leq c$ and then we increase c_e by a factor $\gamma > 1$ and then restart the same stage. To verify the claim, assume $c_e > c$ and the second condition satisfies first, i.e., $\tau = t_k$ but $\|G(\mathbf{x}_{\tau+1}^k)\|_2 > \varepsilon_{k-1}/2$. We will deduce a contradiction. To this end, we use (20) and note the value of t_k

$$\|G(\mathbf{x}_{t_k+1}^k)\|_2 \leq \left(\delta_k + 2\sqrt{2}\delta_k\right)\left(\frac{2}{L}\|G(\mathbf{x}_{k-1})\|_2 + c^{\frac{1}{(1-\theta)}} 2^{\frac{\theta}{(1-\theta)}} \|G(\mathbf{x}_{k-1})\|_2^{\frac{\theta}{(1-\theta)}}\right)$$
$$\leq 4\delta_k\left(\frac{2}{L}\|G(\mathbf{x}_{k-1})\|_2 + c^{\frac{1}{(1-\theta)}} 2^{\frac{\theta}{(1-\theta)}} \|G(\mathbf{x}_{k-1})\|_2^{\frac{\theta}{(1-\theta)}}\right)$$
$$\leq \frac{\epsilon_{k-1}}{4} + \frac{c^{\frac{1}{(1-\theta)}} 2^{\frac{\theta}{(1-\theta)}} \epsilon_{k-1}}{4c_e^{\frac{1}{(1-\theta)}} 2^{\frac{\theta}{(1-\theta)}}} \leq \varepsilon_{k-1}/2 = \varepsilon_k,$$

where the last inequality follows that $c_e > c$. This contradicts to the assumption that $\|G(\mathbf{x}_{\tau+1}^k)\|_2 > \varepsilon_{k-1}/2$, which verifies our claim.

Since c_e is increased by a factor $\gamma > 1$ whenever condition (ii) holds first, so within at most $\lceil \log_\gamma(c/c_0) \rceil$ times condition (ii) holds first. Similarly with at most $\lceil \log_2 \varepsilon_0/\epsilon \rceil$ times that condition (i) holds first before the algorithm terminates. We let T_k denote the total number of iterations in order to make condition (i) satisfies in stage k. First, we can see that $c_e \leq \gamma c$. Let $\delta_k' = \min(\frac{L}{32}, \frac{\varepsilon_{k-1}^p}{16(\gamma c 2^\theta)^{1/(1-\theta)}}) \leq \delta_k$ and $t_k' = \lceil \sqrt{\frac{2L}{\delta_k'}} \log \frac{\sqrt{L(L+\delta_k')}}{\delta_k'} \rceil$. Let s_k denote the number of cycles in each stage in order to have $\|G(\mathbf{x}_{\tau+1}^k)\|_2 \leq \varepsilon_k$. Then $s_k \leq \log_\gamma(c/c_0) + 1$. The total number of iterations of across all stages is bounded by $\sum_{k=1}^K s_k t_k$, which is bounded by

$$\sum_{k=1}^K s_k t_k \leq (1 + \log_\gamma(c/c_0)) \sum_{k=1}^K t_k'$$

Plugging the value of t_k', we can deduce the iteration complexity in Theorem 10 for $\theta \in (0, 1/2]$.

- We consider the proof when $\theta \in (1/2, 1]$. Similar to the proof for $\theta \in (0, 1/2]$, we can easily induce that $F(\mathbf{x}_k) - F_* \leq \epsilon_0$ from Theorem 8. Let $t_k = \lceil \sqrt{\frac{2L}{\delta_k}} \log \frac{\sqrt{L(L+\delta_k)}}{\delta_k} \rceil$. Applying Theorem 9 to the k-the stage of adaAGC and using Lemma 2, we have

$$\|G(\mathbf{x}_{t_k+1}^k)\|_2 \leq \left(\sqrt{L(L+\delta_k)}\left[1 + \sqrt{\frac{\delta_k}{2L}}\right]^{-t_k} + 2\sqrt{2}\delta_k\right)\left(\frac{2}{L} + 2c^2\xi^{2\theta-1}\right)\|G(\mathbf{x}_{k-1})\|_2, \tag{20}$$

Note that at each stage, we check two conditions (i) $\|G(\mathbf{x}_{\tau+1}^k)\|_2 \leq \varepsilon_{k-1}/2$ and (ii) $\tau = t_k$. If the first condition satisfies first, we proceed to the next stage (k increases by 1). If the second condition satisfies first, then we can claim that $c_e \leq c$ and then we increase c_e by a factor $\gamma > 1$ and then restart the same stage. To verify the claim, assume $c_e > c$ and the second condition satisfies first, i.e., $\tau = t_k$ but

$\|G(\mathbf{x}^k_{\tau+1})\|_2 > \varepsilon_{k-1}/2$. We will deduce a contradiction. To this end, we use (20) and note the value of t_k, we have

$$\|G(\mathbf{x}^k_{t_k+1})\|_2 \leq 4\delta_k \left(\frac{2}{L} + 2c^2\xi^{2\theta-1}\right)\|G(\mathbf{x}_{k-1})\|_2$$
$$\leq \frac{\epsilon_{k-1}}{4} + \frac{8c^2\xi^{2\theta-1}}{32c_e^2\epsilon_0^{2\theta-1}}\epsilon_{k-1} \leq \frac{\epsilon_{k-1}}{2} = \epsilon_k,$$

where the last inequality follows that $c_e > c$ and $\xi \leq \epsilon_0$. This contradicts to the assumption that $\|G(\mathbf{x}^k_{\tau+1})\|_2 > \varepsilon_{k-1}/2$, which verifies our claim.

Since c_e is increased by a factor $\gamma > 1$ whenever condition (ii) holds first, so within at most $\lceil\log_\gamma(c/c_0)\rceil$ times condition (ii) holds first. Similarly with at most $\lceil\log_2 \varepsilon_0/\epsilon\rceil$ times that condition (i) holds first before the algorithm terminates. We let T_k denote the total number of iterations in order to make condition (i) satisfies in stage k. First, we can see that $c_e \leq \gamma c$. Let $\delta'_k = \min(\frac{L}{32}, \frac{1}{32(\gamma c)^2\epsilon_0^{2\theta-1}}) \leq \delta_k$ and $t'_k = \lceil\sqrt{\frac{2L}{\delta'_k}}\log\frac{\sqrt{L(L+\delta'_k)}}{\delta'_k}\rceil$. Let s_k denote the number of cycles in each stage in order to have $\|G(\mathbf{x}^k_{\tau+1})\|_2 \leq \varepsilon_k$. Then $s_k \leq \log_\gamma(c/c_0) + 1$. The total number of iterations of across all stages is bounded by $\sum_{k=1}^K s_k t_k$, which is bounded by

$$\sum_{k=1}^K s_k t_k \leq (1 + \log_\gamma(c/c_0)) \sum_{k=1}^K t'_k.$$

Plugging the value of t'_k, we can deduce the iteration complexity in Theorem 10 for $\theta \in (1/2, 1]$.

■

Before ending this section, we would like to remark that if the smoothness parameter L is unknown, one can also employ the backtracking technique pairing with each update to search for L (Nesterov, 2007).

7. Applications

In this section, we present some applications in machine learning and corollaries of our main theorems. In particular, we consider the regularized problems with a smooth loss:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n \ell(\mathbf{x}^\top\mathbf{a}_i, b_i) + \lambda R(\mathbf{x}), \tag{21}$$

where $(\mathbf{a}_i, b_i), i = 1, \ldots, n$ denote a set of training examples, $R(\mathbf{x})$ could be the ℓ_1 norm ($\|\mathbf{x}\|_1$), the ℓ_∞ norm ($\|\mathbf{x}\|_\infty$), or a general form $\|\mathbf{x}\|_p^s$ where $p \geq 1$ and $s \in \mathbb{N}$, or a huber norm (Zadorozhnyi et al., 2016) where $R(\mathbf{x}) = \sum_{i=1}^d h(x_i)$ and $h(x_i)$ is the huber function

$$h(x) = \begin{cases} \delta(|x| - \frac{\delta}{2}) & \text{if } |x| \geq \delta/2 \\ \frac{\delta^2}{2} & \text{otherwise} \end{cases}. \tag{22}$$

We can also consider a broader family of problems that aim to learn a set of models (e.g., in multi-class, multi-label and multi-task learning):

$$\min_{\mathbf{x}_1,\ldots,\mathbf{x}_K\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\ell(\mathbf{x}_k^\top \mathbf{a}_i, b_{ik}) + \lambda\sum_{k=1}^{K}\|\mathbf{x}\|_p, \tag{23}$$

where the regularizer $\sum_{k=1}^{K}\|\mathbf{x}\|_p$ is known as $\ell_{1,p}$ norm.

Next, we present several results about the HEB condition to cover a broad family of loss functions that enjoy the faster convergence of PG and adaAPC.

Corollary 2 *Assume the loss function $\ell(z, b)$ is nonnegative, convex, smooth and semi-algebraic, the the problems in (23) and (21) with $R(\mathbf{x}) = \|\mathbf{x}\|_p^s$ or the huber norm, where $s \in \mathbb{N}$ and $p \geq 1$ is a rational number, satisfy the HEB condition with $\theta \in (0, 1]$ on any sublevel set S_ξ with $\xi > 0$. Hence PG have a global convergence speed of $o(1/t)$.*

Remark: Because of the regularization, the objective function is coercive and proper. The ℓ_p norm with p being a rational number is a semi-algebraic function (Bolte et al., 2014). Finite sum of semi-algebraic functions and composition of semi-algebraic functions are also semi-algebraic. Then we can use Proposition 7 and Theorem 4 to prove the above corollary.

Corollary 3 *Assume the loss function $\ell(z, b)$ is nonnegative, convex, smooth and piecewise quadratic, then the problems in (21) and (23) with ℓ_1 norm, ℓ_∞ norm, Huber norm and $\ell_{1,\infty}$ norm regularization satisfy the HEB condition with $\theta = 1/2$ on any sublevel set S_ξ with $\xi > 0$. Hence adaAGC has a global linear convergence in terms of the proximal gradient's norm and a square root dependence on the condition number.*

Remark: The above corollary follows directly from Proposition 8 and Theorem 10. If the loss function is a logistic loss and the regularizer is a polyhedral function (e.g., ℓ_1, ℓ_∞ and $\ell_{1,\infty}$ norm), we can prove the same result as above using Proposition 11. Examples of convex, smooth and piecewise convex quadratic loss functions include: square loss: $\ell(z, b) = (z-b)^2$ for $b \in \mathbb{R}$; squared hinge loss: $\ell(z, b) = \max(0, 1 - bz)^2$ for $b \in \{1, -1\}$; and huber loss: $\ell(z, b) = h(z - b)$ with $h(x)$ defined in (22).

Finally, it is worth mentioning that the result in Corollary 3 is more general and better than many previous work. For example, Xiao and Zhang (2013) and Lin and Xiao (2014) only considered the lasso problem consisting of a square loss and ℓ_1 norm and derived a linear convergence under the *restricted eigen-value condition*. The PG has been shown to have a linear convergence for solving the lasso problem (Bolte et al., 2015; Necoara et al., 2015; Karimi et al., 2016; Zhang, 2016a; Drusvyatskiy and Lewis, 2016), but it has a linear dependence on the condition number. Many works have considered the structured smooth problem $f(\mathbf{x}) = h(A\mathbf{x}) + g(\mathbf{x})$, where $h(\cdot)$ is a strongly convex function on any compact set (Hou et al., 2013; Zhou et al., 2015; So, 2013; Luo and Tseng, 1992a,b, 1993). Note that this structured family does not cover squared hinge loss and huber loss, and mostly the convergence results in these work are local convergence (i.e., asymptotic convergence) instead of global convergence.

Table 2: Squared hinge loss with ℓ_1 norm regularization

Algorithm	dataset	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-7}$
PG	splice	2201	2201	2201	2201
adaAGC	splice	2123	2123	2123	2123
PG	german.numer	1014	1492	1971	2450
adaAGC	german.numer	762	1010	1338	1586

Table 3: Square loss with ℓ_1 norm regularization

Algorithm	dataset	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-7}$
PG	bodyfat	366637	1110329	1871925	1948897
adaAGC	bodyfat	15414	26174	40526	40905
PG	cpusmall	109298	159908	170915	170915
adaAGC	cpusmall	9571	12623	13571	13571

Table 4: Huber loss with ℓ_1 norm regularization

Algorithm	dataset	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-7}$
PG	bodyfat	258723	423181	602043	681488
adaAGC	bodyfat	16976	16980	23844	25702
PG	cpusmall	74387	112702	159461	190640
adaAGC	cpusmall	26958	32070	36698	38222

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

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)

8. Experimental Results

We conduct some experiments to demonstrate the effectiveness of adaAGC for solving problems of type (2). Specifically, we compare adaAGC and PG with option II for optimizing the squared hinge loss (classification), square loss (regression), huber loss ($\delta = 1$) (regression) with ℓ_1 and ℓ_∞ regularization, which are cases of (21), and we also consider the ℓ_1 constrained ℓ_p norm regression (11) with varying p. We use four datasets from the LibSVM website (Fan and Lin, 2011), which are splice ($n = 1000, d = 60$), german.numer ($n = 1000, d = 24$) for classification, and bodyfat ($n = 252, d = 14$), cpusmall ($n = 8192, d = 12$) for regression. For problems covered by (21), we fix $\lambda = \frac{1}{n}$, and the parameter s in (11) is set to $s = 100$.

We use the backtracking in both PG and adaAGC to search for the smoothness parameter. In adaAGC, we set $c_0 = 10, \gamma = 2$. For fairness, each algorithm starts at the

Table 6: Squared hinge loss with ℓ_∞ regularization

Algorithm	dataset	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-7}$
PG	splice	3514	3724	3724	3724
adaAGC	splice	2336	2456	2456	2456
PG	german.numer	898	898	898	898
adaAGC	german.numer	742	742	742	742

Table 7: Squared loss with ℓ_∞ regularization

Algorithm	dataset	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-7}$
PG	bodyfat	542414	652613	778869	800050
adaAGC	bodyfat	23226	24990	30646	30864
PG	cpusmall	139505	204120	210874	210874
adaAGC	cpusmall	10828	14276	15020	15020

Table 8: Huber loss with ℓ_∞ regularization

Algorithm	dataset	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-7}$
PG	bodyfat	419316	531999	651092	709486
adaAGC	bodyfat	15744	18072	23684	25391
PG	cpusmall	75346	171052	240050	270540
adaAGC	cpusmall	27225	36745	41461	42925

same initial point, which is set to be zero, and we stop each algorithm when the norm of its proximal gradient is less than a prescribed threshold ϵ and **report the total number of proximal mappings**. The results are presented in the Tables 2, 3, 4, 5, 6, 7 and 8, which clearly show that adaAGC converges considerably faster than PG. It is notable that for some problems (see Table 2, 6) the number of proximal mappings is the same value for achieving different precision ϵ. This is because that value is the minimum number of proximal mappings such that the magnitude of the proximal gradient suddenly becomes zero. In Table 5, the numbers in parenthesis indicate the increasing factor in the number of proximal mappings compared to the base case $p = 2$, which show that increasing factors of adaAGC are approximately the square root of that of PG and hence are consistent with our theory.

9. Conclusions

In this paper, we have considered smooth composite optimization problems under a general Hölderian error bound condition. We have established adaptive iteration complexity to the Hölderian error bound condition of proximal gradient and accelerated proximal gradient methods. To eliminate the dependence on the unknown parameter in the error bound condition and enjoy the faster convergence of accelerated proximal gradient method, we have developed a parameter-free adaptive accelerated gradient converging method using the magnitude of the (proximal) gradient as a measure for restart and termination. We have also considered a broad family of norm regularized problems in machine learning and showed faster convergence of the proposed adaptive accelerated gradient converging method.

Appendix

We present some lemmas and propositions that are useful to our analysis.

Proposition 12 *(Bolte et al., 2015, Theorem 5 in v3) Let $f : H \to (-\infty, +\infty]$ be a proper, convex and lower semi-continuous with $\min f = f_*$. Let $r_0 > 0$, $\varphi \in \{\varphi \in C^0[0, r_0) \cap C^1(0, r_0), \varphi(0) = 0, \varphi \text{ is concave}, \varphi > 0\}$, $c > 0, \rho > 0$, and $\bar{x} \in \arg\min f$. If $s\varphi'(s) \geq c\varphi(s)$ for all $s \in (0, r_0)$, and $\varphi(f(x) - f_*) \geq D(x, \arg\min f)$ for all $x \in [0 < f < r_0] \cap B(\bar{x}, \rho)$, then $\varphi'(f(x) - f_*)\|\partial f(x)\|_2 \geq c$ for all $x \in [0 < f < r_0] \cap B(\bar{x}, \rho)$.*

The following proposition is a rephrase of Theorem 3.5 in (Drusvyatskiy and Lewis, 2016).

Proposition 13 *If f is L-smooth and convex, g is proper, convex and lower semi-continuous, $F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$, $\eta > 0$, and define*

$$P_{\eta F}(\mathbf{x}) = \arg\min_{\mathbf{u}} \frac{1}{2}\|\mathbf{u} - \mathbf{x}\|_2^2 + \eta F(\mathbf{u}).$$

Then the following inequality holds:

$$\left\|\frac{1}{\eta}(\mathbf{x} - P_{\eta F}(\mathbf{x}))\right\|_2 \leq (1 + L\eta)\|G_\eta(\mathbf{x})\|_2.$$

A. Perturbation of a Strongly Convex Problem

Lemma 3 *Let $h(\mathbf{x})$ be a σ-strongly convex function, $\mathbf{x}_a^*$ and $\mathbf{x}_b^*$ be the optimal solutions to the following problems.*

$$\mathbf{x}_a^* = \min_{\mathbf{x}\in\mathbb{R}^d} \mathbf{a}^\top \mathbf{x} + h(\mathbf{x}).$$

$$\mathbf{x}_b^* = \min_{\mathbf{x}\in\mathbb{R}^d} \mathbf{b}^\top \mathbf{x} + h(\mathbf{x}).$$

Then

$$\|\mathbf{x}_a^* - \mathbf{x}_b^*\|_2 \leq \frac{2\|\mathbf{a} - \mathbf{b}\|_2}{\sigma}.$$

Proof Let $H_a(\mathbf{x}) = h(\mathbf{x}) + \mathbf{a}^\top\mathbf{x}$ and $H_b(\mathbf{x}) = h(\mathbf{x}) + b^\top\mathbf{x}$. By the strong convexity of $h(\mathbf{x})$, we have

$$\begin{aligned}
&\frac{\sigma}{2}\|\mathbf{x}_a^* - \mathbf{x}_b^*\|_2^2 \\
&\leq H_a(\mathbf{x}_b^*) - H_a(\mathbf{x}_a^*) \\
&= H_b(\mathbf{x}_b^*) + (\mathbf{a} - \mathbf{b})^\top\mathbf{x}_b^* - H_b(\mathbf{x}_a^*) - (\mathbf{a} - \mathbf{b})^\top\mathbf{x}_a^* \\
&\leq (\mathbf{a} - \mathbf{b})^\top(\mathbf{x}_b^* - \mathbf{x}_a^*) \leq \|\mathbf{x}_a^* - \mathbf{x}_b^*\|_2\|\mathbf{a} - \mathbf{b}\|_2,
\end{aligned}$$

where we use the fact $H_b(\mathbf{x}_b^*) \leq H_b(\mathbf{x}_a^*)$. From the above inequality, we can get $\|\mathbf{x}_a^* - \mathbf{x}_b^*\|_2 \leq \frac{2\|\mathbf{a}-\mathbf{b}\|_2}{\sigma}$. ∎

B. Proof of Lemma 1

Proof The conclusion is trivial if $\mathbf{x} \in \Omega_*$. Otherwise, the proof follows Proposition 12. In particular, if we define $\varphi(s) = cs^\theta$, then $D(\mathbf{x}, \Omega_*) \leq \varphi(f(\mathbf{x}) - f_*)$ for any $\mathbf{x} \in \{\mathbf{x} : 0 < f(\mathbf{x}) - f_* \leq \xi\}$ and φ satisfies $s\varphi'(s) \geq \theta\varphi(s)$. By Proposition 12, we have

$$\varphi'(f(\mathbf{x}) - f_*)\|\partial f(\mathbf{x})\|_2 \geq \theta,$$

i.e.,

$$c\|\partial f(\mathbf{x})\|_2 \geq (f(\mathbf{x}) - f_*)^{1-\theta}. \tag{24}$$

When $\theta = 1$, we have $\|\partial f(\mathbf{x})\|_2 \geq 1/c$ for $\mathbf{x} \notin \Omega_*$. As a result, when $\theta \in (0, 1)$

$$D(\mathbf{x}, \Omega_*) \leq c(f(\mathbf{x}) - f_*)^\theta \leq c^{\frac{1}{1-\theta}}\|\partial f(\mathbf{x})\|_2^{\frac{\theta}{1-\theta}},$$

and when $\theta = 1$

$$D(\mathbf{x}, \Omega_*) \leq c(f(\mathbf{x}) - f_*) \leq c^2\xi\|\partial f(\mathbf{x})\|_2.$$

∎

References

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2:183–202, 2009.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146: 459–494, 2014.

Jerome Bolte, Trong Phong Nguyen, Juan Peypouquet, and Bruce Suter. From error bounds to the complexity of first-order descent methods for convex functions. *CoRR*, abs/1510.08234, 2015.

Dmitriy Drusvyatskiy and Adrian S. Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *arXiv:1602.06661*, 2016.

Rong-En Fan and Chih-Jen Lin. Libsvm data: Classification, regression and multi-label. *URL: http://www. csie. ntu. edu. tw/cjlin/libsvmtools/datasets*, 2011.

Olivier Fercoq and Zheng Qu. Restarting accelerated gradient methods with a rough strong convexity estimate. *CoRR*, abs/1609.07358, 2016.

Pinghua Gong and Jieping Ye. Linear convergence of variance-reduced projected stochastic gradient without strong convexity. *CoRR*, abs/1406.1102, 2014.

Ke Hou, Zirui Zhou, Anthony Man-Cho So, and Zhi-Quan Luo. On the linear convergence of the proximal gradient method for trace norm regularization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 710–718, 2013.

Hamed Karimi, Julie Nutini, and Mark W. Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases - European Conference (ECML-PKDD)*, pages 795–811, 2016.

Guoyin Li. Global error bounds for piecewise convex polynomials. *Math. Program.*, 137 (1-2):37–64, 2013.

Qihang Lin and Lin Xiao. An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization. In *Proceedings of the International Conference on Machine Learning, (ICML)*, pages 73–81, 2014.

Zhi-Quan Luo and Paul Tseng. On the convergence of coordinate descent method for convex differentiable minization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992a.

Zhi-Quan Luo and Paul Tseng. On the linear convergence of descent methods for convex essenially smooth minization. *SIAM Journal on Control and Optimization*, 30(2):408–425, 1992b.

Zhi-Quan Luo and Paul Tseng. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46:157–178, 1993.

I. Necoara, Yu. Nesterov, and F. Glineur. Linear convergence of first order methods for non-strongly convex optimization. *CoRR*, abs/1504.06298, 2015.

Y. Nesterov. How to make the gradients small. *Optima 88*, 2012.

Yu. Nesterov. Gradient methods for minimizing composite objective function. Core discussion papers, Universite catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2007.

Yurii Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). 1983.

Yurii Nesterov. *Introductory lectures on convex optimization : a basic course.* Applied optimization. Kluwer Academic Publ., 2004. ISBN 1-4020-7553-7.

H. Nyquist. The optimal lp norm estimator in linear regression models. *Communications in Statistics - Theory and Methods*, 12(21):2511–2524, 1983.

Brendan Odonoghue and Emmanuel Candes. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15(3):715–732, 2015.

R. Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. on Control and Optimization*, 14, 1976.

Anthony Man-Cho So. Non-asymptotic convergence analysis of inexact gradient methods for machine learning without strong convexity. *CoRR*, abs/1309.0113, 2013.

P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 2008.

Po-Wei Wang and Chih-Jen Lin. Iteration complexity of feasible descent methods for convex optimization. *Journal of Machine Learning Research*, 15(1):1523–1548, 2014a.

Po-Wei Wang and Chih-Jen Lin. Iteration complexity of feasible descent methods for convex optimization. *Journal of Machine Learning Research*, 15(1):1523–1548, 2014b.

Lin Xiao and Tong Zhang. A proximal-gradient homotopy method for the sparse least-squares problem. *SIAM Journal on Optimization*, 23(2):1062–1091, 2013.

W. H. Yang. Error bounds for convex polynomials. *SIAM Journal on Optimization*, 19(4): 1633–1647, 2009.

Oleksandr Zadorozhnyi, Gunthard Benecke, Stephan Mandt, Tobias Scheffer, and Marius Kloft. Huber-norm regularization for linear prediction models. In *Machine Learning and Knowledge Discovery in Databases - European Conference (ECML-PKDD)*, pages 714–730, 2016.

Hui Zhang. New analysis of linear convergence of gradient-type methods via unifying error bound conditions. *CoRR*, abs/1606.00269, 2016a.

Hui Zhang. The restricted strong convexity revisited: analysis of equivalence to error bound and quadratic growth. *Optimization Letters*, pages 1–17, 2016b.

Zirui Zhou and Anthony Man-Cho So. A unified approach to error bounds for structured convex optimization problems. *CoRR*, abs/1512.03518, 2015.

Zirui Zhou, Qi Zhang, and Anthony Man-Cho So. L1p-norm regularization: Error bounds and convergence rate analysis of first-order methods. In *Proceedings of the 32nd International Conference on Machine Learning, (ICML)*, pages 1501–1510, 2015.