IMA Journal of Numerical Analysis (2019) **00**, 1–27
doi:10.1093/imanum/drz007

Adaptive restart of accelerated gradient methods under local quadratic growth condition

Olivier Fercoq*
LTCI, Télécom ParisTech, Université Paris-Saclay, 46 rue Barrault, 75013 Paris, France
*Corresponding author: olivier.fercoq@telecom-paristech.fr

and

Zheng Qu
Department of Mathematics, The University of Hong Kong, Run Run Shaw Building, Pokfulam Road, Hong Kong, China
zhengqu@hku.hk

[Received on 20 April 2018; revised on 27 September 2018]

By analyzing accelerated proximal gradient methods under a local quadratic growth condition, we show that restarting these algorithms at any frequency gives a globally linearly convergent algorithm. This result was previously known only for long enough frequencies. Then as the rate of convergence depends on the match between the frequency and the quadratic error bound, we design a scheme to automatically adapt the frequency of restart from the observed decrease of the norm of the gradient mapping. Our algorithm has a better theoretical bound than previously proposed methods for the adaptation to the quadratic error bound of the objective. We illustrate the efficiency of the algorithm on Lasso, regularized logistic regression and total variation denoising problems.

Keywords: accelerated gradient descent; restarting; quadratic growth condition; unknown error bound.

1. Introduction

1.1 *Motivation*

The proximal gradient method aims at minimizing composite convex functions of the form

$$F(x) = f(x) + \psi(x), \quad x \in \mathbb{R}^n,$$

where f is differentiable with Lipschitz gradient and ψ may be nonsmooth, but has an easily computable proximal operator. For a mild additional computational cost, accelerated gradient methods transform the proximal gradient method, for which the optimality gap $F(x_k) - F^\star$ decreases as $O(1/k)$, into an algorithm with 'optimal' $O(1/k^2)$ complexity (Nesterov, 1983). Accelerated variants include the dual accelerated proximal gradient (APG; Nesterov, 2005, 2013), the APG method (Tseng, 2008) and fast iterative soft-thresholding algorithm (FISTA; Beck & Teboulle, 2009). Gradient-type methods, also called first-order methods, are often used to solve large-scale problems because of their good scalability and easiness of implementation that facilitates parallel and distributed computations.

When solving a convex problem whose objective function satisfies a local quadratic error bound (this is a generalization of strong convexity), classical (nonaccelerated) gradient and coordinate descent methods automatically have a linear rate of convergence, i.e., $F(x_k) - F^\star \in O((1-\mu)^k)$ for a problem dependent $0 < \mu < 1$ (Drusvyatskiy & Lewis, 2018; Necoara *et al.*, 2018), whereas one needs to know explicitly the strong convexity (or error bound) parameter in order to set accelerated gradient and accelerated coordinate descent methods to have a linear rate of convergence; see, for instance, Nesterov (2012, 2013), Lee & Sidford (2013), Lin *et al.* (2015a,b), Necoara & Clipici (2016) and Necoara *et al.* (2018). Setting the algorithm with an incorrect parameter may result in a slower algorithm, sometimes even slower than if we had not tried to set an acceleration scheme (O'Donoghue & Candes, 2012). This is a major drawback of the method because, in general, the strong convexity parameter is difficult to estimate.

In the context of accelerated gradient method with unknown strong convexity parameter, Nesterov (2013) proposed a restarting scheme that adaptively approximates the strong convexity parameter. The same idea was exploited by Lin & Xiao (2015) for sparse optimization. Nesterov (2013) also showed that, instead of deriving a new method designed to work better for strongly convex functions, one can restart the accelerated gradient method and get a linear convergence rate. However, the restarting frequency he proposed still depends explicitly on the strong convexity of the function, and so O'Donoghue & Candes (2012) introduced some heuristics to adaptively restart the algorithm and obtain good results in practice.

1.2 *Contributions*

In this paper, we show that, if the objective function is convex and satisfies a local quadratic error bound, we can restart accelerated gradient methods at *any* frequency and get a linearly convergent algorithm. The rate depends on an estimate of the quadratic error bound, and we show that for a wide range of this parameter, one obtains a faster rate than without acceleration. In particular, we do not require this estimate to be smaller than the actual value. In that way, our result supports and explains the practical success of arbitrary periodic restart for accelerated gradient methods.

Then as the rate of convergence depends on the match between the frequency and the quadratic error bound, we design a scheme to automatically adapt the frequency of restart from the observed decrease of the norm of the gradient mapping. The approach follows the lines of Nesterov (2013), Lin & Xiao (2015) and Liu & Yang (2017). We proved that, if our current estimate of the local error bound was correct, the norm of the gradient mapping would decrease at a prescribed rate. We just need to check this decrease and when the test fails, we have a certificate that the estimate was too large.

Our algorithm has a better theoretical bound than previously proposed methods for the adaptation to the quadratic error bound of the objective. In particular, we can make use of the fact that our study shows that the norm of the gradient mapping will decrease even when we had a wrong estimate of the local error bound.

In Section 2 we recall the main convergence results for accelerated gradient methods and show that a fixed restart leads to a linear convergence rate. In Section 3 we present our adaptive restarting rule. Finally, we present numerical experiments on the lasso, logistic regression and total variation (TV) denoising problems in Section 4.

2. Accelerated gradient schemes

2.1 *Problem and assumptions*

We consider the following optimization problem:

$$\min_{x\in\mathbb{R}^n} \; F(x) := f(x) + \psi(x), \tag{2.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a differentiable convex function and $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a proper, closed and convex function. We denote by $F^\star$ the optimal value of (2.1) and assume that the optimal solution set $\mathscr{X}_\star$ is nonempty. Throughout the paper $\|\cdot\|$ denotes the Euclidean norm. For any positive vector $v \in \mathbb{R}^n_+$, we denote by $\|\cdot\|_v$ the weighted Euclidean norm

$$\|x\|_v^2 \overset{\text{def}}{=} \sum_{i=1}^n v_i (x^i)^2$$

and $\operatorname{dist}_v(x, \mathscr{X}_\star)$ the distance of x to the closed convex set $\mathscr{X}_\star$ with respect to the norm $\|\cdot\|_v$. In addition, we assume that ψ is simple, in the sense that the proximal operator defined as

$$\operatorname{prox}_{v,\psi}(x) := \arg\min_y \left\{ \frac{1}{2}\|x-y\|_v^2 + \psi(y) \right\} \tag{2.2}$$

is easy to compute, for any positive vector $v \in \mathbb{R}^n_+$. We also make the following *smoothness* and *local quadratic error bound* assumption.

ASSUMPTION 2.1 There is a positive vector $L \in \mathbb{R}^n_+$ such that

$$f(x) \leqslant f(y) + \langle \nabla f(y), x-y \rangle + \frac{1}{2}\|x-y\|_L^2, \quad \forall x, y \in \mathbb{R}^n. \tag{2.3}$$

ASSUMPTION 2.2 For any $x_0 \in \operatorname{dom}(F)$, there is $\mu > 0$ such that

$$F(x) \geqslant F^\star + \frac{\mu}{2}\operatorname{dist}_L^2(x, \mathscr{X}_\star), \quad \forall x \in \left[F \leqslant F(x_0)\right], \tag{2.4}$$

where $[F \leqslant F(x_0)]$ denotes the set of all x such that $F(x) \leqslant F(x_0)$. We denote by $\mu_F(L, x_0)$ the largest μ satisfying (2.4).

Assumption 2.2 is also referred to as *local quadratic growth condition*. It is known that Assumptions 2.1 and 2.2 guarantee the linear convergence of proximal gradient method (Drusvyatskiy & Lewis, 2018; Necoara *et al.*, 2018) with complexity bound $O(\log(1/\varepsilon)/\mu_F(L, x_0))$.

2.2 *Accelerated gradient schemes*

We first recall in Algorithms 1 and 2 two classical APG schemes. For identification purpose we refer to them respectively as FISTA (Beck & Teboulle, 2009) and APG (Tseng, 2008). As pointed out in Tseng (2008), the accelerated schemes were first proposed by Nesterov (2004).

REMARK 2.3 We have written the algorithms in a unified framework to emphasize their similarities. Practical implementations usually consider only two variables: (x_k, y_k) for FISTA and (y_k, z_k) for APG.

REMARK 2.4 The vector L used in the proximal operation step (Line 4) of Algorithms 1 and 2 should satisfy Assumption 2.1. If such L is not known *a priori*, we can incorporate a line search procedure; see, for example, Nesterov (2013).

Algorithm 1 $\hat{x} \leftarrow \text{FISTA}(x_0, K)$ (Beck & Teboulle, 2009)

```
Set $\theta_0 = 1$ and $z_0 = x_0$.
for $k = 0, 1, \cdots, K-1$ do
    $y_k = (1-\theta_k)x_k + \theta_k z_k$
    $x_{k+1} = \arg\min_{x \in \mathbb{R}^n} \left\{ \langle \nabla f(y_k), x - y_k \rangle + \frac{1}{2}\|x - y_k\|_L^2 + \psi(x) \right\}$
    $z_{k+1} = z_k + \frac{1}{\theta_k}(x_{k+1} - y_k)$
    $\theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}$
end for
$\hat{x} \leftarrow x_{k+1}$
```

Algorithm 2 $\hat{x} \leftarrow \text{APG}(x_0, K)$ (Tseng, 2008)

```
Set $\theta_0 = 1$ and $z_0 = x_0$.
for $k = 0, 1, \cdots, K-1$ do
    $y_k = (1-\theta_k)x_k + \theta_k z_k$
    $z_{k+1} = \arg\min_{z \in \mathbb{R}^n} \left\{ \langle \nabla f(y_k), z - y_k \rangle + \frac{\theta_k}{2}\|z - z_k\|_L^2 + \psi(z) \right\}$
    $x_{k+1} = y_k + \theta_k(z_{k+1} - z_k)$
    $\theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}$
end for
$\hat{x} \leftarrow x_{k+1}$
```

The most simple restarted accelerated gradient method has a *fixed restarting frequency*. This is Algorithm 3, which restarts periodically Algorithm 1 or 2. Here the restarting period is fixed to be some integer $K \geqslant 1$. In Section 3 we will also consider *adaptive restarting frequency* with a varying restarting period.

Algorithm 3 FixedRES(x_0, K)

```
for $t = 0, 1, \cdots,$ do
    $x_{(t+1)K} \leftarrow \text{APG}(x_{tK}, K)$ or $x_{(t+1)K} \leftarrow \text{FISTA}(x_{tK}, K)$
end for
```

2.3 *Convergence results for accelerated gradients methods*

2.3.1 *Basic results.* In this section we gather a list of known results, shared by FISTA and APG, which will be used later to build restarted methods. Although all the results presented in this subsection have been proved or can be derived easily from existing results, for completeness all the proof is given in the appendix. We first recall the following properties on the sequence $\{\theta_k\}$.

LEMMA 2.5 (Tseng, 2008). The sequence (θ_k) defined by $\theta_0 = 1$ and $\theta_{k+1} = \frac{\sqrt{\theta_k^4+4\theta_k^2}-\theta_k^2}{2}$ satisfies

$$\frac{1}{k+1} \leqslant \theta_k \leqslant \frac{2}{k+2} \tag{2.5}$$

$$\frac{1-\theta_{k+1}}{\theta_{k+1}^2} = \frac{1}{\theta_k^2}, \quad \forall k = 0, 1, \dots \tag{2.6}$$

$$\theta_{k+1} \leqslant \theta_k, \quad \forall k = 0, 1, \dots. \tag{2.7}$$

We shall also need the following relation of the sequences. The iterates of Algorithms 1 and 2 satisfy for all $k \geqslant 1$,

$$x_{k+1} = (1-\theta_k)x_k + \theta_k z_{k+1}. \tag{2.8}$$

It is known that the sequence of objective values $\{F(x_k)\}$ generated by accelerated schemes, in contrast to that generated by proximal gradient schemes, does not decrease monotonically. However, this sequence is always upper bounded by the initial value $F(x_0)$. Following Lin & Xiao (2015), we refer to this result as the *nonblowout* property of accelerated schemes.

PROPOSITION 2.6 The iterates of Algorithms 1 and 2 satisfy for all $k \geqslant 1$,

$$F(x_k) \leqslant F(x_0).$$

Proof. The proof for Algorithm 1 can be found in Lin & Xiao (2015). The proof for Algorithm 2 can be found in the appendix. □

The nonblowout property of accelerated schemes can be found in many papers; see, for example, Lin & Xiao (2015) and Wen *et al.* (2017). It will be repeatedly used in this paper to derive the linear convergence rate of restarted methods. Finally, recall the following fundamental property for accelerated schemes.

PROPOSITION 2.7 (Tseng, 2008; Beck & Teboulle, 2009). The iterates of Algorithms 1 and 2 satisfy for all $k \geqslant 1$,

$$\frac{1}{\theta_{k-1}^2}\left(F(x_k) - F^\star\right) + \frac{1}{2}\|z_k - x^\star\|_L^2 \leqslant \frac{1}{2}\|x_0 - x^\star\|_L^2, \tag{2.9}$$

where $x^\star$ is an arbitrary point in $\mathscr{X}_\star$.

Again, Proposition 2.7 is not new and can be easily derived from existing results; see the appendix for a proof. We derive from Proposition 2.7 a direct corollary.

Corollary 2.8 The iterates of Algorithms 1 and 2 satisfy for all $k \geqslant 1$,

$$\frac{1}{\theta_{k-1}^2}\left(F(x_k) - F^\star\right) + \frac{1}{2}\operatorname{dist}_L(z_k, \mathscr{X}_\star)^2 \leqslant \frac{1}{2}\operatorname{dist}_L(x_0, \mathscr{X}_\star)^2. \tag{2.10}$$

Remark 2.9 All the results presented above hold without Assumption 2.2.

2.3.2 *Conditional linear convergence.* One can derive directly from Corollary 2.8 and Assumption 2.2 the following decreasing property.

Corollary 2.10 (Necoara *et al.*, 2018). If $\hat{x}$ is the output of Algorithm 1 or 2 with input (x_0, K) and Assumptions 2.1 and 2.2 hold, then

$$F(\hat{x}) - F^\star \leqslant \frac{\theta_{K-1}^2}{\mu_F(L, x_0)}\left(F(x_0) - F^\star\right) \tag{2.11}$$

and

$$\operatorname{dist}_L(\hat{x}, \mathscr{X}_\star)^2 \leqslant \frac{\theta_{K-1}^2}{\mu_F(L, x_0)}\operatorname{dist}_L(x_0, \mathscr{X}_\star)^2. \tag{2.12}$$

Proposition 2.11 (Necoara *et al.*, 2018). Let $\{x_{tK}\}_{t \geqslant 0}$ be the sequence generated by Algorithm 3 with $K \geqslant 1$. If Assumptions 2.1 and 2.2 hold, then we have

$$F(x_{tK}) - F^\star \leqslant \left(\frac{\theta_{K-1}^2}{\mu_F(L, x_0)}\right)^t \left(F(x_0) - F^\star\right), \quad \forall t \geqslant 1$$

and

$$\operatorname{dist}_L(x_{tK}, \mathscr{X}_\star)^2 \leqslant \left(\frac{\theta_{K-1}^2}{\mu_F(L, x_0)}\right)^t \operatorname{dist}_L(x_0, \mathscr{X}_\star)^2, \quad \forall t \geqslant 1.$$

Remark 2.12 Although Proposition 2.11 provides a guarantee of linear convergence when $\theta_{K-1}^2/\mu_F(L, x_0) < 1$, it does not give any interesting information in the case when $\theta_{K-1}^2/\mu_F(L, x_0) \geqslant 1$.

For any $\mu > 0$, define

$$K(\mu) := \left\lceil \frac{2e}{\sqrt{\mu}} - 1 \right\rceil. \tag{2.13}$$

By the right inequality of (2.5), letting the restarting period $K \geqslant K(\mu_F(L, x_0))$ implies

$$\frac{\theta_{K-1}^2}{\mu_F(L, x_0)} \leqslant e^{-2}. \tag{2.14}$$

Therefore, if we know in advance $\mu_F(L, x_0)$ and restart Algorithm 1 or 2 every $K(\mu_F(L, x_0))$ iterations, then after computing

$$K\big(\mu_F(L, x_0)\big) \ln \frac{\operatorname{dist}_L(x_0, \mathscr{X}_\star)}{\sqrt{\varepsilon}} = O\left(\sqrt{\mu_F^{-1}(L, x_0)} \ln \varepsilon^{-1}\right) \tag{2.15}$$

number of proximal gradient mappings, we get a point x such that $\operatorname{dist}_L(x, \mathscr{X}_\star)^2 \leqslant \varepsilon$.

2.3.3 *Unconditional linear convergence.* We now prove a contraction result on the distance to the optimal solution set.

THEOREM 2.13 If $\hat{x}$ is the output of Algorithm 1 or 2 with input (x_0, K) and Assumptions 2.1 and 2.2 hold, then

$$\operatorname{dist}_L(\hat{x}, \mathscr{X}_\star)^2 \leqslant \min\left(\frac{\theta_{K-1}^2}{\mu_F(L, x_0)}, \frac{1}{1 + \frac{\mu_F(L, x_0)}{2\theta_{K-1}^2}}\right) \operatorname{dist}_L(x_0, \mathscr{X}_\star)^2. \tag{2.16}$$

Proof. For simplicity we denote $\mu_F = \mu_F(L, x_0)$. Note that by Assumption 2.2 and Proposition 2.6 we have for all $k \in \mathbb{N}$,

$$F(x_k) - F^\star \geqslant \frac{\mu_F}{2} \operatorname{dist}_L(x_k, \mathscr{X}_\star)^2. \tag{2.17}$$

For all $k \in \mathbb{N}$, let us denote $x_k^\star$ the projection of x_k onto $\mathscr{X}_\star$. For all $k \in \mathbb{N}$ and $\sigma_k \in [0, 1]$, we have

$$\begin{aligned}
\frac{1}{2}\|x_{k+1} - x_{k+1}^\star\|_L^2 &= \frac{\sigma_k}{2}\|x_{k+1} - x_{k+1}^\star\|_L^2 + \frac{1-\sigma_k}{2}\|x_{k+1} - x_{k+1}^\star\|_L^2 \\
&\leqslant \frac{\sigma_k}{2}\|x_{k+1} - x_{k+1}^\star\|_L^2 + \frac{1-\sigma_k}{2}\|x_{k+1} - \theta_k x_0^\star - (1-\theta_k)x_k^\star\|_L^2 \\
&\leqslant \frac{\sigma_k}{\mu_F}\big(F(x_{k+1}) - F^\star\big) + \frac{1-\sigma_k}{2}\theta_k\|z_{k+1} - x_0^\star\|_L^2 + \frac{1-\sigma_k}{2}(1-\theta_k)\|x_k - x_k^\star\|_L^2,
\end{aligned}$$

where the first inequality follows from the convexity of $\mathscr{X}_\star$, and the second inequality follows from (2.8) and (2.17). Let us choose $\sigma_k = \frac{1}{1+\theta_k/\mu_F}$ so that

$$\frac{\sigma_k}{\mu_F} = (1 - \sigma_k)\frac{\theta_k}{\theta_k^2}.$$

We proceed as

$$\begin{aligned}\frac{1}{2}\operatorname{dist}_L(x_{k+1},\mathscr{X}_\star)^2 &\leqslant (1-\sigma_k)\theta_k\left(\frac{1}{\theta_k^2}\big(F(x_{k+1})-F^\star\big)+\frac{1}{2}\left\|z_{k+1}-x_0^\star\right\|_L^2\right)\\ &\quad+\frac{(1-\sigma_k)(1-\theta_k)}{2}\left\|x_k-x_k^\star\right\|_L^2\\ &\overset{(2.9)}{=}(1-\sigma_k)\left(\frac{\theta_k}{2}\left\|x_0-x_0^\star\right\|_L^2+\frac{(1-\theta_k)}{2}\left\|x_k-x_k^\star\right\|_L^2\right)\\ &=\frac{1}{1+\mu_F/\theta_k}\left(\frac{\theta_k}{2}\operatorname{dist}_L(x_0,\mathscr{X}_\star)^2+\frac{(1-\theta_k)}{2}\operatorname{dist}_L(x_k,\mathscr{X}_\star)^2\right).\end{aligned}\tag{2.18}$$

Denote $\Delta_k=\operatorname{dist}_L(x_k,\mathscr{X}_\star)^2$. Remark that

$$\Delta_1\leqslant\frac{1}{1+\mu_F/\theta_0}\Delta_0\leqslant\frac{1}{1+0.5\mu_F/\theta_0^2}\Delta_0$$

so that we may prove that $\Delta_k\leqslant\frac{1}{1+0.5\mu_F/\theta_{k-1}^2}\Delta_0$ by induction. To this end, let us assume that $\Delta_k\leqslant\frac{1}{1+0.5\mu_F/\theta_{k-1}^2}\Delta_0$. Then using (2.18),

$$\begin{aligned}\Delta_{k+1}&\leqslant\frac{1}{1+\mu_F/\theta_k}\Big(\theta_k\Delta_0+(1-\theta_k)\Delta_k\Big)\\ &\leqslant\frac{1}{1+\mu_F/\theta_k}\left(\theta_k\Delta_0+\frac{1-\theta_k}{1+0.5\mu_F/\theta_{k-1}^2}\Delta_0\right)\\ &=\frac{\theta_k(1+0.5\mu_F/\theta_{k-1}^2)+(1-\theta_k)}{(1+\mu_F/\theta_k)\big(1+0.5\mu_F/\theta_{k-1}^2\big)}\Delta_0.\end{aligned}$$

Using (2.6) one can then easily check that

$$\frac{\theta_k\big(1+0.5\mu_F/\theta_{k-1}^2\big)+(1-\theta_k)}{(1+\mu_F/\theta_k)\big(1+0.5\mu_F/\theta_{k-1}^2\big)}\leqslant\frac{1}{1+0.5\mu_F/\theta_k^2}\Leftrightarrow 0\leqslant 2\theta_k^3+\mu_F(1-\theta_k)$$

and so inequality (2.16) comes by combining this with Proposition 2.11. □

Theorem 2.13 allows us to derive immediately an explicit linear convergence rate of Algorithm 3, regardless of the choice of $K\geqslant 1$.

COROLLARY 2.14 Let $\{x_{tK}\}_{t\geqslant 0}$ be the sequence generated by Algorithm 3 with fixed restarting period $K \geqslant 1$. If Assumptions 2.1 and 2.2 hold, then

$$\mathrm{dist}_L(x_{tK}, \mathscr{X}_\star)^2 \leqslant \left(\min\left(\frac{\theta_{K-1}^2}{\mu_F(L,x_0)}, \frac{1}{1+\frac{\mu_F(L,x_0)}{2\theta_{K-1}^2}}\right)\right)^t \mathrm{dist}_L(x_0, \mathscr{X}_\star)^2.$$

REMARK 2.15 When $K = 1$, the rate suggested by Corollary 2.14 is

$$\mathrm{dist}_L(x_t, \mathscr{X}_\star)^2 \leqslant \left(\frac{1}{1+\frac{\mu_F(L,x_0)}{2}}\right)^t \mathrm{dist}_L(x_0, \mathscr{X}_\star)^2,$$

which is of the same order as proximal gradient mapping. When, instead, we take $K = K(\mu_F(L,x_0))$ as defined in (2.13), then we obtain

$$\mathrm{dist}_L(x_{tK}, \mathscr{X}_\star)^2 \leqslant e^{-2}\,\mathrm{dist}_L(x_0, \mathscr{X}_\star)^2.$$

By Corollary 2.14, the number of proximal mappings needed to reach an ε accuracy on the distance is bounded by

$$\frac{2K}{\log\left(1+\frac{\mu_F(L,x_0)}{2\theta_{K-1}^2}\right)}\log\frac{\mathrm{dist}_L(x_0, \mathscr{X}_\star)^2}{\varepsilon}.$$

In particular, if we choose $K \sim 1/\sqrt{\mu_F(L,x_0)}$ then we get an iteration complexity bound

$$O(1/\sqrt{\mu_F(L,x_0)}\log(1/\varepsilon)). \tag{2.19}$$

REMARK 2.16 In Wen *et al.* (2017), the local linear convergence of the sequence generated by FISTA with arbitrary (fixed or adaptive) restarting frequency was proved. Our Theorem 2.13 not only yields the global linear convergence of such sequence, but also gives an explicit bound on the convergence rate. Also, note that although an asymptotic linear convergence rate can be derived from the proof of Lemma 3.6 in Wen *et al.* (2017), it can be checked that the asymptotic rate in Wen *et al.* (2017) is not as good as ours. In fact, an easy calculation shows that even restarting with optimal period $K \sim 1/\sqrt{\mu_F}$, their asymptotic rate only leads to the complexity bound $O(1/\mu_F^2 \log(1/\varepsilon))$. Moreover, our restarting scheme is more flexible because the internal block in Algorithm 3 can be replaced by any scheme that satisfies all the properties presented in Section 2.3.1.

3. Adaptive restarting of accelerated gradient schemes

Although Theorem 2.13 guarantees a linear convergence of the restarted method (Algorithm 3), it requires the knowledge of $\mu_F(L,x_0)$ to attain the complexity bound (2.19). In this section, we show how to combine Corollary 2.14 with Nesterov's adaptive restart method, first proposed in Nesterov (2013), in order to obtain a complexity bound close to (2.19) that does not depend on a guess on $\mu_F(L,x_0)$.

3.1 *Bounds on gradient mapping norm*

We first show the following inequalities that generalize similar ones in Nesterov (2013). Hereinafter, we define the proximal mapping

$$T(x) := \arg\min_{y\in\mathbb{R}^n} \left\{ \langle \nabla f(x), y - x\rangle + \frac{1}{2}\|y-x\|_L^2 + \psi(y) \right\}.$$

Proposition 3.1 If Assumptions 2.1 and 2.2 hold, then for any $x \in \mathbb{R}^n$ we have

$$\|T(x)-x\|_L^2 \leqslant 2\left(F(x) - F(T(x))\right) \leqslant 2\left(F(x)-F^\star\right), \tag{3.1}$$

$$\operatorname{dist}_L(T(x), \mathscr{X}_\star) \leqslant \frac{4}{\mu_F(L,x)}\|x - T(x)\|_L \tag{3.2}$$

and

$$F(T(x)) - F^\star \leqslant \frac{8\|x-T(x)\|_L^2}{\mu_F(L,x)}. \tag{3.3}$$

Proof. The inequality (3.1) follows directly from Nesterov (2013, Theorem 1). By the convexity of F, for any $q \in \partial F(T(x))$ and $x^\star \in \mathscr{X}_\star$,

$$\langle q, T(x) - x^\star\rangle \geqslant F(T(x)) - F^\star, \quad \forall\, q \in \partial F(T(x)), x^\star \in \mathscr{X}_\star.$$

Furthermore, by Nesterov (2013, Theorem 1), for any $q \in \partial F(T(x))$ and $x^\star \in \mathscr{X}_\star$,

$$2\|x - T(x)\|_L \left\|T(x) - x^\star\right\|_L \geqslant \langle q, T(x) - x^\star\rangle.$$

Therefore,

$$2\|x - T(x)\|_L \operatorname{dist}_L(T(x), \mathscr{X}_\star) \geqslant F(T(x)) - F^\star. \tag{3.4}$$

By (3.1), we know that $F(T(x)) \leqslant F(x)$ and in view of (2.4),

$$F(T(x)) - F^\star \geqslant \frac{\mu_F(L,x)}{2}\operatorname{dist}_L^2(T(x), \mathscr{X}_\star).$$

Combining the latter two inequalities, we get

$$2\|x - T(x)\|_L \geqslant \frac{\mu_F(L,x)}{2}\operatorname{dist}_L(T(x), \mathscr{X}_\star). \tag{3.5}$$

Plugging (3.5) back to (3.4), we get (3.3). □

Remark 3.2 Condition (3.2) is usually referred to as an *error bound condition*. It was proved in Drusvyatskiy & Lewis (2018) that under Assumption 2.1, the error bound condition is equivalent to the quadratic growth condition (2.4).

Corollary 3.3 Suppose Assumptions 2.1 and 2.2 hold and denote $\mu_F = \mu_F(L, x_0)$. Then the iterates of Algorithm 3 satisfy for all $t \geqslant 1$,

$$\|T(x_{tK}) - x_{tK}\|_L^2 \leqslant \theta_{K-1}^2 \left(\min \left(\frac{\theta_{K-1}^2}{\mu_F}, \frac{1}{1 + \frac{\mu_F}{2\theta_{K-1}^2}} \right) \right)^{t-1} \operatorname{dist}_L(x_0, \mathscr{X}_\star)^2. \tag{3.6}$$

Moreover, if there is $u_0 \in \mathbb{R}^n$ such that $x_0 = T(u_0)$ and denote $\mu'_F = \mu_F(L, u_0)$, then

$$\|T(x_{tK}) - x_{tK}\|_L^2 \leqslant \frac{16}{\mu'_F} \left(\frac{\theta_{K-1}^2}{\mu'_F} \right) \left(\min \left(\frac{\theta_{K-1}^2}{\mu'_F}, \frac{1}{1 + \frac{\mu'_F}{2\theta_{K-1}^2}} \right) \right)^{t-1} \|x_0 - u_0\|_L^2. \tag{3.7}$$

Proof. By (3.1) and (2.10), we have

$$\|T(x_{tK}) - x_{tK}\|_L^2 \leqslant 2\big(F(x_{tK}) - F^\star\big) \leqslant \theta_{K-1}^2 \operatorname{dist}_L \big(x_{(t-1)K}, \mathscr{X}_\star\big)^2.$$

Now applying Corollary 2.14 we get (3.6). If in addition $x_0 = T(u_0)$ then, in view of (3.2),

$$\operatorname{dist}_L(x_0, \mathscr{X}_\star)^2 \leqslant \left(\frac{4}{\mu'_F} \right)^2 \|x_0 - u_0\|_L^2.$$

The second inequality (3.7) then follows from combining the latter inequality, (3.6) and the fact that $\mu'_F \leqslant \mu_F$. □

3.2 *Adaptively restarted algorithm*

Inequality (3.7) provides a way to test whether our guess on $\mu_F(L, x_0)$ is too large. Indeed, let $x_0 = T(u_0)$. Then if $\mu \leqslant \mu_F(L, u_0) \leqslant \mu_F(L, x_0)$ and we run Algorithm 3 with $K = K(\mu)$, then necessarily we have

$$\|T(x_{tK}) - x_{tK}\|_L^2 \leqslant \frac{16}{\mu} \left(\frac{\theta_{K-1}^2}{\mu} \right) \left(\min \left(\frac{\theta_{K-1}^2}{\mu}, \frac{1}{1 + \frac{\mu}{2\theta_{K-1}^2}} \right) \right)^{t-1} \|x_0 - u_0\|_L^2. \tag{3.8}$$

It is essential that both sides of (3.8) are computable, so that we can check this inequality for each estimate μ. If (3.8) does not hold then we know that $\mu > \mu_F(L, u_0)$. The idea was originally proposed by Nesterov (2013) and later generalized in Lin & Xiao (2015), where instead of restarting, they incorporate the estimate μ into the update of θ_k. As a result, the complexity analysis only works for strongly convex objective function and seems not to hold under Assumption 2.2.

Our adaptively restarted algorithm is described in Algorithm 4. We start from an initial estimate μ_0 and restart Algorithm 1 or 2 with period $K(\mu_s)$ defined in (2.13). Note that by (2.14),

$$\min\left(\frac{\theta^2_{K_s-1}}{\mu_s}, \frac{1}{1+\frac{\mu_s}{2\theta^2_{K_s-1}}}\right) = \frac{\theta^2_{K_s-1}}{\mu_s}.$$

At the end of each restarting period, we test condition (3.8), the opposite of which is given by the first inequality at Line 13. If it holds then we continue with the same estimate μ_s, and thus the same restarting period, otherwise we decrease μ_s by one half and repeat. Our stopping criteria is based on the norm of proximal gradient, same as in related work by Lin & Xiao (2015) and Liu & Yang (2017).

Algorithm 4 $(\hat{x}, \hat{s}, \hat{N}) \leftarrow \mathrm{AdaRES}(x_0)$

```
1.  Parameters: $\varepsilon$, $\mu_0$
2.  $s \leftarrow -1, t_s \leftarrow 0$
3.  $x_{s,t_s} \leftarrow x_0$
4.  $x_{s+1,0} \leftarrow T(x_{s,t_s})$
5.  repeat
6.      $s \leftarrow s+1$
7.      $C_s \leftarrow \frac{16\|x_{s,0}-x_{s-1,t_{s-1}}\|^2_L}{\mu_s}$
8.      $K_s \leftarrow K(\mu_s)$
9.      $t \leftarrow 0$
10.     repeat
11.         $x_{s,t+1} \leftarrow$ Algorithm 1 $(x_{s,t}, K_s)$ or $x_{s,t+1} \leftarrow$ Algorithm 2 $(x_{s,t}, K_s)$
12.         $t \leftarrow t+1$
13.     until $\|T(x_{s,t}) - x_{s,t}\|^2_L > C_s(\theta^2_{K_s-1}/\mu_s)^t$ or $\|T(x_{s,t}) - x_{s,t}\|^2_L \leqslant \varepsilon$
14.     $t_s \leftarrow t$
15.     $x_{s+1,0} \leftarrow T(x_{s,t_s})$
16.     $\mu_{s+1} \leftarrow \mu_s/2$
17. until $\|x_{s+1,0} - x_{s,t_s}\|^2_L \leqslant \varepsilon$
18. $\hat{x} \leftarrow T(x_{s,t_s}), \hat{s} \leftarrow s, \hat{N} \leftarrow 1 + \sum_{s=0}^{\hat{s}}(t_s K_s + 1)$
```

REMARK 3.4 Although Line 13 of Algorithm 4 requires to compute the proximal gradient mapping $T(x_{s,t})$, one should remark that this $T(x_{s,t})$ is in fact given by the first iteration of Algorithm 1 $(x_{s,t}, K_s)$

or Algorithm 2 $(x_{s,t}, K_s)$ (this holds because $y_0 = x_0 = z_0$). Hence, except for $t = t_s$, the computation of $T(x_{s,t})$ does not incur additional computational cost. Therefore, the output $\hat{N}$ of Algorithm 4 records the total number of proximal gradient mappings needed to get $\|T(x) - x\|_L^2 \leqslant \varepsilon$.

We first show the following nonblowout property for Algorithm 4.

LEMMA 3.5 For any $-1 \leqslant s \leqslant \hat{s}$ and $0 \leqslant t \leqslant t_s$, we have

$$F(T(x_{s,t})) \leqslant F(x_{s,t}) \leqslant F(x_0).$$

Proof. Since $x_{s,t+1}$ is the output of Algorithm 1 or 2 with input $x_{s,t}$, we know by Proposition 2.6 that

$$F(x_{s,t_s}) \leqslant F(x_{s,t_s-1}) \leqslant \ldots \leq F(x_{s,0}).$$

By (3.1),

$$F(x_{s+1,0}) = F(T(x_{s,t_s})) \leqslant F(x_{s,t_s}) \leqslant F(x_{s,0}).$$

The right inequality then follows by induction since $x_{-1,0} = x_0$. The left inequality follows from (3.1). □

LEMMA 3.6 For any $0 \leqslant s \leqslant \hat{s}$, if $\mu_s \leqslant \mu_F(L, x_0)$ then

$$F(x_{s,t}) - F^\star \leqslant \left(\frac{\theta_{K_s-1}^2}{\mu_s}\right)^t \big(F(x_{s,0}) - F^\star\big), \ \ \forall\, 1 \leqslant t \leqslant t_s. \tag{3.9}$$

Proof. This is a direct application of Proposition 2.11 and Lemma 3.5. □

From Lemmas 3.5 and 3.6, we obtain immediately the following key results for the complexity bound of Algorithm 4.

COROLLARY 3.7 Let C_s be the constant defined in Line 7 of Algorithm 4. We have

$$C_s \leqslant \frac{32\big(F(x_0) - F^\star\big)}{\mu_s}, \ \ \forall\, s \geqslant 0. \tag{3.10}$$

If for any $0 \leqslant s \leqslant \hat{s}$ we have $\mu_s \leqslant \mu_F(L, x_0)$, then

$$\big\|T(x_{s,t}) - x_{s,t}\big\|_L^2 \leqslant C_s \left(\frac{\theta_{K_s-1}^2}{\mu_s}\right)^t, \ \ \forall\, 0 \leqslant t \leqslant t_s \tag{3.11}$$

and

$$\big\|T(x_{s,t}) - x_{s,t}\big\|_L^2 \leqslant 2e^{-2t}\big(F(x_0) - F^\star\big), \ \ \forall\, 0 \leqslant t \leqslant t_s. \tag{3.12}$$

Proof. The bound on C_s follows from (3.1) applied at $x_{s-1,t_{s-1}}$ and Lemma 3.5. The second bound can be derived from (3.1), Lemma 3.6 and (3.3),

$$\begin{aligned}\left\|T(x_{s,t})-x_{s,t}\right\|_L^2 &\leqslant 2\left(F(x_{s,t})-F^\star\right)\leqslant 2\left(\frac{\theta_{K_s-1}^2}{\mu_s}\right)^t\left(F(x_{s,0})-F^\star\right)\\ &=2\left(\frac{\theta_{K_s-1}^2}{\mu_s}\right)^t\left(F(T(x_{s-1,t_{s-1}}))-F^\star\right)\\ &\leqslant \frac{16\left\|x_{s-1,t_{s-1}}-T(x_{s-1,t_{s-1}})\right\|_L^2}{\mu_F(L,x)}\left(\frac{\theta_{K_s-1}^2}{\mu_s}\right)^t.\end{aligned}$$

For the third one, it suffices to apply Lemma 3.5, Lemma 3.6 together with (3.1) and the fact that

$$\frac{\theta_{K_s-1}^2}{\mu_s}\leqslant e^{-2},\quad \forall\, s\geqslant 0. \tag{3.13}$$

□

PROPOSITION 3.8 Consider Algorithm 4. If $\mu_0\leqslant\mu_F(L,x_0)$, then $\hat{s}=0$ and

$$t_0\leqslant\left\lceil \ln\sqrt{\frac{2(F(x_0)-F^\star)}{\varepsilon}}\right\rceil. \tag{3.14}$$

Proof. If $\mu_0\leqslant\mu_F(L,x_0)$, by (3.11), the inner loop terminates if and only if

$$\left\|T(x_{0,t_0})-x_{0,t_0}\right\|_L^2\leqslant\varepsilon.$$

Therefore, $\hat{s}=0$. Then (3.14) is derived from (3.12). □

THEOREM 3.9 Suppose Assumptions 2.1 and 2.2 hold. Consider the adaptively restarted accelerated gradient Algorithm 4. If the initial estimate of the local quadratic error bound satisfies $\mu_0\leqslant\mu_F(L,x_0)$ then the number of iterations $\hat{N}$ is bounded by

$$\hat{N}\leqslant\left\lceil\frac{2e}{\sqrt{\mu_0}}-1\right\rceil\left\lceil \ln\sqrt{\frac{2(F(x_0)-F^\star)}{\varepsilon}}\right\rceil+2. \tag{3.15}$$

If $\mu_0>\mu_F(L,x_0)$, then

$$\begin{aligned}\hat{N}\leqslant 1&+\left\lceil\log_2\frac{\mu_0}{\mu_F(L,x_0)}\right\rceil+\frac{2e}{\sqrt{2}-1}\left(\sqrt{\frac{2}{\mu_F(L,x_0)}}-\sqrt{\frac{1}{\mu_0}}\right)\left\lceil\ln\sqrt{\frac{32(F(x_0)-F^\star)}{\varepsilon\mu_F(L,x_0)}}\right\rceil\\ &+\frac{2\sqrt{2}e}{\sqrt{\mu_F(L,x_0)}}\left\lceil\ln\sqrt{\frac{2(F(x_0)-F^\star)}{\varepsilon}}\right\rceil.\end{aligned} \tag{3.16}$$

Proof. The first case is a direct application of Proposition 3.8.

Let us first concentrate on the case $\mu_0 > \mu_F(L, x_0)$. For simplicity we denote $\mu_F = \mu_F(L, x_0)$. Define $\bar{l} = \lceil \log_2 \mu_0 - \log_2 \mu_F \rceil \geqslant 1$. Then $\mu_{\bar{l}} \leqslant \mu_F$ and by Proposition 3.8, we know that $\hat{s} \leqslant \bar{l}$ and if $\hat{s} = \bar{l}$,

$$t_{\bar{l}} \leqslant \left\lceil \ln \sqrt{\frac{2(F(x_0) - F^\star)}{\varepsilon}} \right\rceil. \tag{3.17}$$

Now we consider t_s for $0 \leqslant s \leqslant \bar{l} - 1$. Note that $\varepsilon < \|T(x_{s,t}) - x_{s,t}\|_L^2 \leqslant C_s(\theta^2_{K_s-1}/\mu_s)^t$ cannot hold for t, satisfying

$$C_s\left(\theta^2_{K_s-1}/\mu_s\right)^t \leqslant \varepsilon. \tag{3.18}$$

By (3.10),

$$C_s \leqslant \frac{32\left(F(x_0) - F^\star\right)}{\mu_F}, \quad 0 \leqslant s \leqslant \bar{l} - 1. \tag{3.19}$$

In view of (3.13) and (3.19), (3.18) holds for any $t \geqslant 0$ such that

$$32\left(F(x_0) - F^\star\right) e^{-2t} \leqslant \varepsilon \mu_F.$$

Therefore,

$$t_s \leqslant \left\lceil \ln \sqrt{\frac{32(F(x_0) - F^\star)}{\varepsilon \mu_F}} \right\rceil, \quad 0 \leqslant s \leqslant \bar{l} - 1. \tag{3.20}$$

By the definition (2.13),

$$\begin{aligned} K_0 + \cdots + K_{\bar{l}-1} &= \sum_{s=0}^{\bar{l}-1} \left\lceil \frac{2e}{\sqrt{\mu_s}} - 1 \right\rceil \leqslant \sum_{s=0}^{\bar{l}-1} 2e \frac{\sqrt{2^s}}{\sqrt{\mu_0}} \\ &= \frac{2e/\sqrt{\mu_0}}{\sqrt{2} - 1} \left(\sqrt{2^{\bar{l}}} - 1\right) \leqslant \frac{2e}{\sqrt{2} - 1} \left(\sqrt{\frac{2}{\mu_F}} - \sqrt{\frac{1}{\mu_0}}\right). \end{aligned}$$

Therefore,

$$\sum_{s=0}^{\bar{l}-1} (t_s K_s + 1) \leqslant \frac{2e}{\sqrt{2} - 1} \left(\sqrt{\frac{2}{\mu_F}} - \sqrt{\frac{1}{\mu_0}}\right) \left\lceil \ln \sqrt{\frac{32(F(x_0) - F^\star)}{\varepsilon \mu_F}} \right\rceil + \bar{l}. \tag{3.21}$$

TABLE 1 *Comparison of the iteration complexity of accelerated gradient methods with adaptation to the local error bound. Here μ_F is any constant satisfying (2.4) and is at least μ/L when either f or Ψ is μ strongly convex*

Algorithm	Complexity bound	Assumption
Nesterov (2013)	$O\Big(\frac{1}{\sqrt{\mu/L}}\ln\big(\frac{L}{\mu}\big)\ln\big(\frac{L}{\mu\varepsilon}\big)\Big)$	μ-strong convexity of Ψ
Lin & Xiao (2015)	$O\Big(\frac{1}{\sqrt{\mu/L}}\ln\big(\frac{L}{\mu}\big)\ln\big(\frac{L}{\mu\varepsilon}\big)\Big)$	μ-strong convexity of f
Liu & Yang (2017)	$O\Big(\frac{1}{\sqrt{\mu_F}}\big(\ln\big(\frac{1}{\mu_F}\big)\big)^2\ln\big(\frac{1}{\varepsilon}\big)\Big)$	local quadratic error bound (2.4)[†]
This work	$O\Big(\frac{1}{\sqrt{\mu_F}}\ln\big(\frac{1}{\mu_F\varepsilon}\big)\Big)$	local quadratic error bound (2.4)

[†] The complexity bound is given only for the case of quadratic error bound for comparison purposes, but the paper actually deals with a more general assumption (3.22).

Combining (3.17) and (3.21), we get

$$\begin{aligned}\hat{N} &\leqslant 1+\sum_{k=0}^{\bar{l}}(t_sK_s+1)\\ &\leqslant 1+\bar{l}+\frac{2e}{\sqrt{2}-1}\left(\sqrt{\frac{2}{\mu_F}}-\sqrt{\frac{1}{\mu_0}}\right)\left\lceil \ln\sqrt{\frac{32(F(x_0)-F^\star)}{\varepsilon\mu_F}}\right\rceil\\ &\quad+\frac{2e}{\sqrt{\mu_{\bar{l}}}}\left\lceil \ln\frac{2(F(x_0)-F^\star)}{\varepsilon}\right\rceil.\end{aligned}$$

Then (3.16) follows by noting that $\mu_{\bar{l}}\geqslant \mu_F/2$. □

In Table 1 we compare the worst-case complexity bound of four algorithms that adaptively restart accelerated algorithms. Note that the algorithms proposed by Nesterov (2013) and Lin & Xiao (2015) require respectively the strong convexity of Ψ and f. Also, when the strong convexity is unknown, it is unclear whether we can transfer the strong convexity from one to another. However, the algorithm of Liu & Yang (2017) also applies to the case when local Hölderian error bound condition holds. The latter condition requires the existence of $\theta\geqslant 1$ and a constant $\mu_F(L,x_0,\theta)>0$ such that

$$F(x)\geqslant F^\star+\frac{\mu_F(L,x_0,\theta)}{2}\operatorname{dist}_L(x,\mathcal{X}_\star)^\theta,\quad \forall x\in\big[F\leqslant F(x_0)\big]. \tag{3.22}$$

When $\theta=2$, we recover the local quadratic growth condition. In this case, the algorithm of Liu & Yang (2017) has a complexity bound $\tilde{O}(1/\sqrt{\mu_F})$, where $\tilde{O}$ hides logarithm terms. For fair comparison, we compute and add back the logarithm terms based on their proof and get the bound $O(\frac{1}{\sqrt{\mu_F}}\ln\big(\frac{1}{\mu_F}\big)^2\ln\big(\frac{1}{\varepsilon}\big))$.

We can see that the analysis of our algorithm leads to a worst-case complexity that is $\log(1/\mu_F)$ times better than the previous work.

REMARK 3.10 We note that the complexity bound (3.16) increases with the constant L used in Algorithm 1 or 2. Therefore, for better efficiency we should use the smallest L that is available to us and satisfies Assumption 2.1.

3.3 *A stricter test condition*

The test condition (Line 13) in Algorithm 4 can be further strengthened as follows.

For simplicity denote $\mu_F = \mu_F(L, x_0)$ and

$$\alpha_s(\mu) := \min\left(\frac{\theta_{K_s-1}^2}{\mu}, \frac{1}{1+\frac{\mu}{2\theta_{K_s-1}^2}}\right).$$

Let any $0 \leqslant s' \leqslant s \leqslant \hat{s}$ and $1 \leqslant t \leqslant t_s$. Then for the same reason as (3.7), we have

$$\left\|T(x_{s,t}) - x_{s,t}\right\|_L^2 \leqslant \frac{16}{\mu_F}\left(\frac{\theta_{K_s-1}^2}{\mu_F}\right)\left(\alpha_s(\mu_F)\right)^{t-1}\prod_{j=s-1}^{s'}(\alpha_j(\mu_F))^{t_j}\|x_{s',0} - x_{s'-1,t_{s'-1}}\|_L^2.$$

This suggests to replace Line 7 of Algorithm 4 by

$$C_s \leftarrow \frac{16}{\mu_s}\min\left\{\prod_{j=s-1}^{s'}\left(\alpha_j(\mu_s)\right)^{t_j}\|x_{s',0} - x_{s'-1,t_{s'-1}}\|_L^2 : 0 \leqslant s' \leqslant s\right\}. \tag{3.23}$$

As we only decrease the value of C_s, all the theoretical analysis holds and Theorem 3.9 is still true with the new C_s defined in (3.23). Moreover, this change allows us to identify more quickly a too large μ_s, and thus can improve the practical performance of the algorithm.

Furthermore, if we find that

$$\left\|T(x_{s,t}) - x_{s,t}\right\|_L^2 > C_s(\theta_{K_s-1}^2/\mu_s)^t$$

then before running Algorithm 1 or 2 with $\mu_{s+1} := \mu_s/2$, we can first do a test on μ_{s+1}, i.e., check the condition

$$\begin{aligned}\left\|T(x_{s,t}) - x_{s,t}\right\|_L^2 \\ \leqslant \min\left\{\frac{16}{\mu_{s+1}}\left(\frac{\theta_{K_s-1}^2}{\mu_{s+1}}\right)\left(\alpha_s(\mu_{s+1})\right)^{t-1}\prod_{j=s-1}^{s'}(\alpha_j(\mu_{s+1}))^{t_j}\|x_{s',0} - x_{s'-1,t_{s'-1}}\|_L^2 : 0 \leqslant s' \leqslant s\right\}.\end{aligned} \tag{3.24}$$

If (3.24) holds then we go to Line 10. Otherwise, μ_{s+1} is still too large and we decrease it further by one half.

Our experiments use this stricter test condition since its computational cost is negligible compared to the cost of proximal gradient steps.

3.4 *Looking for an ε solution*

Instead of an x such that $\|T(x)-x\|^2 \leqslant \varepsilon$, we may be interested in an x such that $F(x)-F^\star \leqslant \varepsilon'$, which is an ε'solution.

In view of (3.3), if $\|x-T(x)\|_L^2 \leqslant \frac{\varepsilon'\mu_F(L,x)}{8}$ then $F(T(x))-F^\star \leqslant \varepsilon'$. As a result, except from the fact that we cannot terminate the algorithm in Line 17, Algorithm 4 is applicable with $\varepsilon = \frac{\varepsilon'\mu_F(L,x)}{8}$.

By plugging $\varepsilon = \frac{\varepsilon'\mu_F(L,x)}{8}$ in Theorem 3.9, we then obtain an ε' solution after a number of iterations at most equal to

$$1+\left\lceil \log_2 \frac{\mu_0}{\mu_F(L,x_0)} \right\rceil + \left\lceil \frac{2e}{\sqrt{2}-1}\left(\sqrt{\frac{2}{\mu_F(L,x_0)}}-\sqrt{\frac{1}{\mu_0}}\right)\right\rceil \left\lceil \ln\sqrt{\frac{2^8(F(x_0)-F^\star)}{\varepsilon'\mu_F(L,x_0)^2}} \right\rceil$$
$$+\frac{2\sqrt{2}e}{\sqrt{\mu_F(L,x_0)}}\left\lceil \ln\sqrt{\frac{16(F(x_0)-F^\star)}{\varepsilon'\mu_F(l,x_0)}}\right\rceil.$$

Note that compared to the result of Theorem 3.9, we only add a constant factor.

4. Numerical experiments

In this section we present some numerical results to demonstrate the effectiveness of the proposed algorithms. We apply Algorithm 4 to solve regression and classification problems that typically take the form of

$$\min_{x\in\mathbb{R}^n} F(x) := g(Ax)+\psi(x), \tag{4.1}$$

where $A \in \mathbb{R}^{m\times n}$, $g:\mathbb{R}^m \to \mathbb{R}$ has Lipschitz continuous gradient and $\psi:\mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ is simple. The model includes in particular the L^1-regularized least squares problem (Lasso) and the L^1-L^2-regularized logistic regression problem. Note that the following problem is dual to (4.1):

$$\max_{y\in\mathbb{R}^m} G(y) := -\psi^*(A^\top y)-g^*(-y),$$

where g^* (resp. ψ^*) denotes the convex conjugate function of g (resp. ψ). We define the primal-dual gap associated with a point $x \in \mathrm{dom}(\psi)$ as

$$F(x)-G\big(-\alpha(x)A^\top\nabla g(Ax)\big), \tag{4.2}$$

where $\alpha(x)$ is the largest $\alpha \in [0,1]$ such that $G(-\alpha A^\top\nabla g(Ax)) < +\infty$. When g and ψ are polyhedral, one can easily get closed form solutions for this one-dimensional sub-problem. Otherwise, we may approximate $\alpha(x)$ by dichotomy. Note that $x \in \mathrm{dom}(\psi)$ is an optimal solution of (4.1) if and only if the associated primal-dual gap (4.2) equals 0.

We compare five methods: gradient descent (GD), FISTA (Beck & Teboulle, 2009), AdapAPG (Lin & Xiao, 2015), AdaAGC (Liu & Yang, 2017) and AdaRES (Algorithm 4 using FISTA in Line 11). We plot the primal-dual gap (4.2) vs. running time. Note that GD and FISTA do not depend on the initial guess of the value $\mu_F(L, x_0)$. In all our experiments we start with $x_0 = 0 \in \mathbb{R}^n$.

4.1 *Lasso problem*

We present in Fig. 1 the experimental results for solving the L^1-regularized least squares problem (Lasso)

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|_2^2 + \frac{\|A^\top b\|_\infty}{\lambda_1}\|x\|_1 \tag{4.3}$$

on the dataset cpusmall_scale[2] with $n = 12$ and $m = 8192$. The value L is set to be trace$(A^\top A)$. We test with $\lambda_1 = 10^4$, $\lambda_1 = 10^5$ and $\lambda_1 = 10^6$. For each value of λ_1, we vary the initial guess μ_0 from 0.1 to 10^{-5}. Compared with AdaAPG and AdaAGC, Algorithm 4 seems to be more efficient and less sensitive to the guess of μ_F.

In Table 2 we fix $\lambda_1 = 10^5$ and test with five different datasets[1,2] and three different initial guess $\mu_0 = 10^{-1}$, 10^{-3}, 10^{-5}. For each experiment we stop the algorithm either when the running time exceeds 3000 s or when the primal-dual gap defined as in (4.2) is smaller than $10^{-10}F(x_0)$. We report in row ‘P-D Gap’ the smallest primal-dual gap value obtained in 3000 s and in row ‘Time’ the computational time when this value is first attained. We mark with ‘-’ the experiments for which the required precision has not been reached before 3000 s. It is not surprising to see that the adaptively restarted methods sometimes can be much slower than FISTA, due to the bad guess of μ_0. However, comparing the three adaptive methods, AdaRES either reaches the precision in smallest time or finds the smallest primal-dual gap in 3000 s, for each dataset and each μ_0.

4.2 *Logistic regression*

We present in Fig. 2 the experimental results for solving the L^1-L^2 regularized logistic regression problem

$$\min_{x\in\mathbb{R}^n} \frac{\lambda_1}{2\|A^\top b\|_\infty}\sum_{j=1}^{m}\log\left(1+\exp\left(b_j a_j^\top x\right)\right) + \|x\|_1 + \frac{\lambda_2}{2}\|x\|^2 \tag{4.4}$$

on the dataset dorothea[3] with $n = 100{,}000$ and $m = 800$. In our experiments, we set

$$\lambda_2 = \frac{L}{10n},$$

[1] Provided by Michael Revow, https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/.

[2] Some of the datasets are for multi-class classification. We use them directly on the regression problem for test purpose.

[3] Provided by DuPont Pharmaceuticals Research Laboratories and KDD Cup 2001, http://archive.ics.uci.edu/ml/datasets/dorothea.

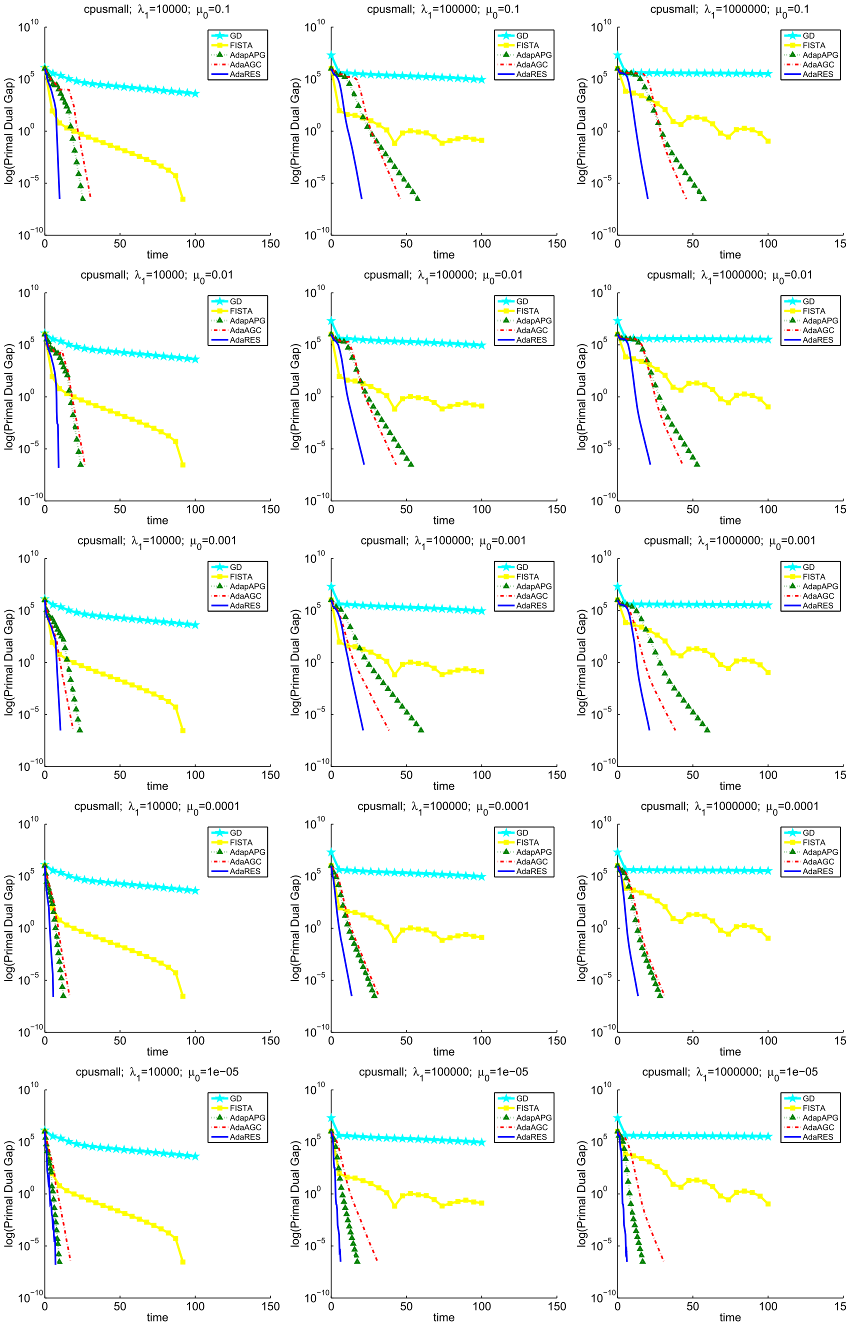

FIG. 1. Experimental results on the Lasso problem (4.3) and the dataset cpusmall. Column-wise: we solve the same problem with a different *a priori* on the quadratic error bound. Row-wise: we use the same *a priori* on the quadratic error bound, but the weight of the 1-norm is varying.

TABLE 2 *Experimental results on the Lasso problem (4.3) with* $\lambda_1 = 10^5$

Instance	m/n		GD	FISTA	AdaAGC			AdapAPG			AdaRES		
					$\mu_0 = 10^{-1}$	$\mu_0 = 10^{-3}$	$\mu_0 = 10^{-5}$	$\mu_0 = 10^{-1}$	$\mu_0 = 10^{-3}$	$\mu_0 = 10^{-5}$	$\mu_0 = 10^{-1}$	$\mu_0 = 10^{-3}$	$\mu_0 = 10^{-5}$
connect4	67557	Time(s)	–	–	–	–	–	–	–	–	–	–	–
	126	P-D Gap	15070	**0.0063**	2309.3	374.84	43.007	288.41	289.71	43.085	37.117	15.631	37.117
usps	2007	Time(s)	–	–	–	–	–	–	–	–	–	–	–
	256	P-D Gap	10269	**0.5198**	5144.2	671.37	107.03	589.37	435.49	154.44	8.7657	7.8612	83.237
vehicle	846	Time(s)	319.35	13.727	8.7084	7.7071	5.438	10.939	11.501	3.3014	3.9141	4.0882	**1.1663**
	18	P-D Gap	3.1e−07	2.5e−07	3.1e−07	3.1e−07	3.1e−07	3.1e−07	3.1e−07	3.1e−07	3.1e−07	3.1e−07	1.9e−07
triazine	186	Time(s)	–	60.134	33.503	38.742	18.112	28.572	28.17	23.708	12.168	**10.725**	13.351
	60	P-D Gap	2.9e−07	3.7e−09	4.1e−09	4.1e−09	4.1e−09	4.1e−09	4.1e−09	4.1e−09	4.1e−09	4.1e−09	4.1e−09
sector	3207	Time(s)	–	–	–	–	–	–	–	–	–	–	–
	55197	P-D Gap	10837	**2.2**	1616.8	1055.3	706.5	783.9	762.8	439.7	57.9	48.2	235.2

For each instance, we indicate in bold the smallest running time.

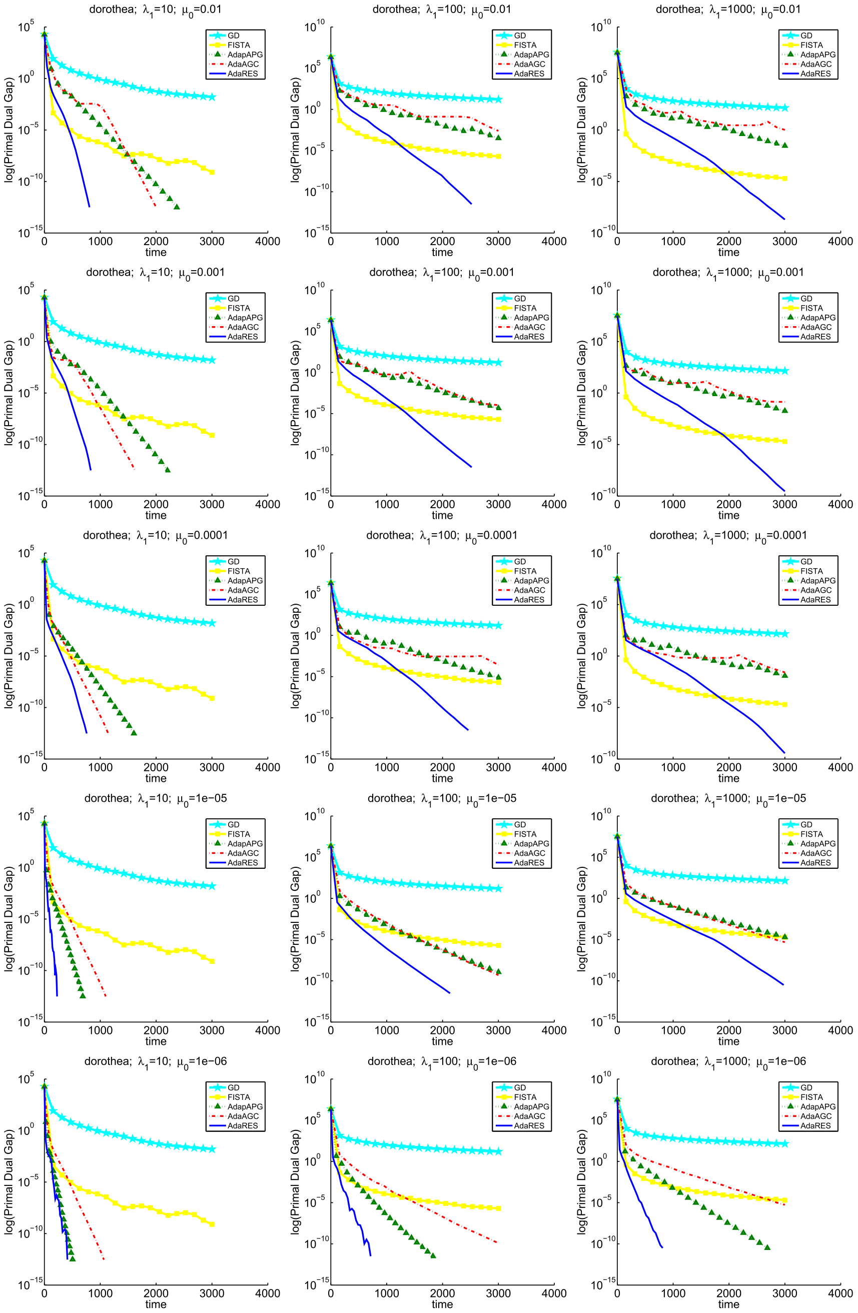

FIG. 2. Experimental results on the logistic regression problem (4.4) and the dataset dorothea. Column-wise: we solve the same problem with a different *a priori* on the quadratic error bound. Row-wise: we use the same *a priori* on the quadratic error bound, but the weight of the regularization is varying.

where

$$L := \frac{\lambda_1}{8\|A^\top b\|_\infty} \sum_{j=1}^{n} \sum_{i=1}^{m} (b_j A_{ij})^2$$

is an upper bound of the Lipschitz constant of the function f given by

$$f(x) = \frac{\lambda_1}{2\|A^\top b\|_\infty} \sum_{j=1}^{m} \log\left(1 + \exp(b_j a_j^\top x)\right).$$

Thus, $\mu_F \geqslant 1/(10n) = 10^{-6}$. We test with $\lambda_1 = 10$, $\lambda_1 = 100$ and $\lambda_1 = 1000$. For each value of λ_1, we vary the initial guess μ_0 from 0.01 to 10^{-6}. On this problem Algorithm 4 also outperforms AdaAPG and AdaAGC in all the cases.

In Table 3 we fix $\lambda_1 = 10^3$ and test with five different datasets[2] and three different initial guess $\mu_0 = 10^{-2}, 10^{-4}, 10^{-6}$. As the datasets have different n, we set

$$\lambda_2 = \frac{L}{\max(10n, 10^6)},$$

so that $\mu_F \geqslant 10^{-6}$. The same conclusion as for the Lasso problem (Section 4.1) can be drawn.

4.3 *TV denoising*

We took an image of a lampshade with $p_1 \times p_2 = 706 \times 1087$ pixels. As the scene was not properly lit, there is image noise on the picture. Hence, we consider a TV denoising problem to remove the noise

$$\min_{u\in\mathbb{R}^n} \frac{1}{2}\|b-u\|_2^2 + \lambda_1\|Au\|_1. \tag{4.5}$$

The vector $b \in \mathbb{R}^m$ is an image in vector format (all the columns are stacked), A is the discrete gradient matrix (it has at most two nonzeros elements per row) and $n = 2m = 2p_1p_2$.

As the TV denoising objective cannot be written as $f(x) + \psi(x)$ with a simple ψ, we instead solve the dual TV denoising problem given by

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\left\|A^\top x + b\right\|_2^2 + I_{\lambda_1 B_{2,\infty}}(x). \tag{4.6}$$

$I_{\lambda_1 B_{2,\infty}}$ is the convex indicator function of the ball $\lambda_1 B_{2,\infty}$. This convex indicator amounts to constraining x such that $\forall\, i \in \{1,\dots,m\}, \sqrt{x_{2i}^2 + x_{2i+1}^2} \leqslant \lambda_1$.

As in the previous cases, we compare the performance of Algorithm 4, AdaAPG and AdaAGC in Fig. 3. Here also, Algorithm 4 is faster than the other adaptive accelerated gradient methods and for each level of regularization, there is a choice of μ_0 such that Algorithm 4 is faster than FISTA.

TABLE 3 *Experimental results on the logistic regression problem (4.4) with* $\lambda_1 = 10^3$

Instance	m/n		GD	FISTA	AdaAGC			AdapAPG			AdaRES		
					$\mu_0 = 10^{-2}$	$\mu_0 = 10^{-4}$	$\mu_0 = 10^{-6}$	$\mu_0 = 10^{-2}$	$\mu_0 = 10^{-4}$	$\mu_0 = 10^{-6}$	$\mu_0 = 10^{-2}$	$\mu_0 = 10^{-4}$	$\mu_0 = 10^{-6}$
leukemia	34	Time(s)	–	312.46	586.19	978.08	986.94	1796.7	1671.5	483.85	205.71	194.8	**63.283**
	7129	P-D Gap	1.5495	9.0e−08	9.0e−08	4.89e−04	9.0e−08	9.0e−08	9.0e−08	9.0e−08	9.0e−08	9.0e−08	8.91e−08
madelon	600	Time(s)	–	–	–	–	–	–	–	1778.3	748.83	661.16	**358.3**
	500	P-D Gap	10.414	4.2e−08	0.033	0.486	4.7e−08	8.2e−04	4.6e−04	4.0e−09	4.0e−09	4.0e−09	4.0e−09
gisette	1000	Time(s)	–	–	–	–	–	–	–	–	–	–	–
	5000	P-D Gap	16880	**0.067**	5963.1	168.11	148.83	4691.9	149.72	18.6	29.1	0.775	**0.068**
mushrooms	8124	Time(s)	–	162.62	680.95	554.13	261.32	631.3	537.21	154.06	261.13	278.82	**62.57**
	112	P-D Gap	0.03	4.5e−08	4.5e−08	4.5e−08	4.5e−08	4.5e−08	4.5e−08	4.5e−08	4.5e−08	4.5e−08	4.5e−08
phishing	11055	Time(s)	2306.6	446.28	232.79	128.97	123.5	222.88	102.31	333.57	105.62	**43.367**	115.1
	68	P-D Gap	8.5e−07	8.4e−07	8.4e−07	8.4e−07	8.4e−07	8.4e−07	8.5e−07	8.1e−07	8.4e−07	8.4e−07	8.3e−07

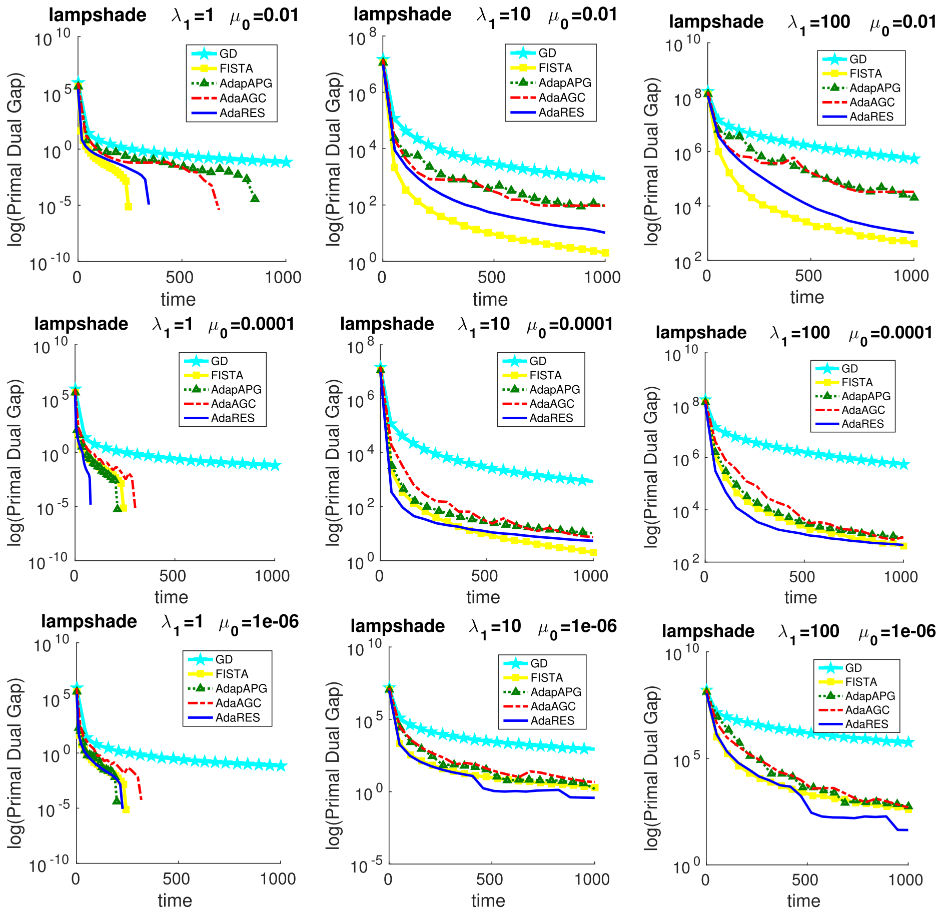

FIG. 3. Experimental results on the TV denoising problem (4.6) on the image lampshade. Column-wise: we solve the same problem with a different *a priori* on the quadratic error bound. Row-wise: we use the same *a priori* on the quadratic error bound, but the weight of the regularization is varying.

5. Conclusion

In this work, we show that global linear convergence is guaranteed if we restart at any frequency accelerated gradient methods under a local quadratic growth condition. We then propose an adaptive restarting strategy based on the decrease of the norm of proximal gradient mapping. Compared with similar methods dealing with unknown local error bound condition number, our algorithm has a better worst-case complexity bound and practical performance.

Our algorithm can be further extended to a more general setting when Hölderian error bound (3.22) is satisfied. Another avenue of research is that the accelerated coordinate descent method (Fercoq & Richtárik, 2015) faces the same issue as full gradient methods: to get an accelerated rate of convergence, one needs to estimate the strong convexity coefficient (Lin *et al.*, 2015b). In Fercoq & Qu (2016), an algorithm with fixed periodic restart was proposed. We may also consider adaptive restart for the accelerated coordinate descent method, to get more efficiency in large-scale computation.

Funding

EPSRC (EP/K02325X/1 *Accelerated Coordinate Descent Methods for Big Data Optimization* to O.F.); Centre for Numerical Algorithms and Intelligent Software (EP/G036136/1 and the Scottish Funding Council to O.F.); the Orange/Telecom ParisTech think tank Phi-TAB; ANR (ANR-11-LABX-0056-LMH, LabEx LMH as part of the Investissement d'avenir project to O.F.); Hong Kong Research Grants Council Early Career Scheme (27303016 to Z.Q.); this research was conducted using the HKU Information Technology Services research computing facilities that are supported in part by the Hong Kong UGC Special Equipment Grant (SEG HKU09).

References

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

Drusvyatskiy, D. & Lewis, A. S. (2018) Error bounds, quadratic growth, and linear convergence of proximal methods. *Math. Oper. Res.*, **43**, 919–948.

Fercoq, O. & Richtárik, P. (2015) Accelerated, parallel and proximal coordinate descent. *SIAM J. Optim.*, **25**, 1997–2023.

Fercoq, O. & Qu, Z. (2016) Restarting accelerated gradient methods with a rough strong convexity estimate, *Preprint arXiv:1609.07358*.

Lee, Y. T. & Sidford, A. (2013) Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science* (FOCS '13). Washington, DC: IEEE Computer Society, pp. 147–156.

Lin, H., Mairal, J. & Harchaoui, Z. (2015a) A universal catalyst for first-order optimization. *Advances in Neural Information Processing Systems* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama & R. Garnett eds), vol. 28. Vancouver: Curran Associates, Inc., pp. 3384–3392.

Lin, Q., Lu, Z. & Xiao, L. (2015b) An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM J. Optim.*, **25**, 2244–2273.

Lin, Q. & Xiao, L. (2015) An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization. *Comput. Optim. Appl.*, **60**, 633–674.

Liu, M. & Yang, T. (2017) Adaptive accelerated gradient converging method under holderian error bound condition. *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett eds), vol. 30. California: Curran Associates, Inc., pp. 3104–3114.

Necoara, I. & Clipici, D. (2016) Parallel random coordinate descent method for composite minimization: convergence analysis and error bounds. *SIAM J. Optim.*, **26**, 197–226.

Necoara, I., Nesterov, Y. & Glineur, F. (2018) Linear convergence of first order methods for non-strongly convex optimization. *Math. Program.*

Nesterov, Y. (1983) A method of solving a convex programming problem with convergence rate $O\left(1/k^2\right)$. *Soviet Math. Dokl.*, **27**, 372–376.

Nesterov, Y. (2004) *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Kluwer Academic Publishers.

Nesterov, Y. (2005) Smooth minimization of nonsmooth functions. *Math. Program.*, **103**, 127–152.

Nesterov, Y. (2012) Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM J. Optim.*, **22**, 341–362.

Nesterov, Y. (2013) Gradient methods for minimizing composite functions. *Math. Program.*, **140**, 125–161.

O'Donoghue, B. & Candes, E. (2012) Adaptive restart for accelerated gradient schemes. *Found. Comput. Math.*, **15**, 715–732.

Tseng, P. (2008) On accelerated proximal gradient methods for convex-concave optimization. *SIAM J. Optim.* (submitted).

Wen, B., Chen, X. & Pong, T. K. (2017) Linear convergence of proximal gradient algorithm with extrapolation for a class of nonconvex nonsmooth minimization problems. *SIAM J. Optim.*, **27**, 124–145.

Appendix A. Proof of Proposition 2.6

Proof of Proposition 2.6. The proof for FISTA can be found in Lin & Xiao (2015).

Next consider the iterates of APG. By Tseng (2008, Proposition 1), for any $x \in \mathbb{R}^n$, the iterates of APG satisfy the following property:

$$F(x_{k+1}) - F(x) + \frac{\theta_k^2}{2}\|x - z_{k+1}\|_v^2 \leqslant (1-\theta_k)\big(F(x_k) - F(x)\big) + \frac{\theta_k^2}{2}\|x - z_k\|_v^2. \tag{A.1}$$

By taking $x = x_k$, we obtain

$$F(x_{k+1}) - F(x_k) \leqslant \frac{\theta_k^2}{2}\|x_k - z_k\|_v^2 - \frac{\theta_k^2}{2}\|x_k - z_{k+1}\|_v^2.$$

The update of APG implies

$$x_{k+1} = (1-\theta_k)x_k + \theta_k z_{k+1},$$

which yields

$$x_{k+1} - z_{k+1} = (1-\theta_k)(x_k - z_{k+1}).$$

Therefore, for any $k \geqslant 1$,

$$F(x_{k+1}) - F(x_k) \leqslant \frac{\theta_k^2}{2}\|x_k - z_k\|_v^2 - \frac{\theta_k^2}{2(1-\theta_k)^2}\|x_{k+1} - z_{k+1}\|_v^2. \tag{A.2}$$

Applying (A.2) recursively, using the decreasing property (2.7) and (A.1) for the first step, we obtain

$$F(x_{k+1}) - F(x_0) \leqslant F(x_1) - F(x_0) + \frac{1}{2}\|x_1 - z_1\|_v^2 \leqslant \frac{1}{2}\|x_0 - z_0\|_v^2 = 0.$$

□