An Adaptive Accelerated Proximal Gradient Method and its Homotopy Continuation for Sparse Optimization

Your Name EMAIL@YOURDOMAIN.EDU
Your Fantastic Institute, 314159 Pi St., Palo Alto, CA 94306 USA

Your CoAuthor's Name EMAIL@COAUTHORDOMAIN.EDU
Their Fantastic Institute, 27182 Exp St., Toronto, ON M6H 2T1 CANADA

Abstract

We first propose an adaptive accelerated proximal gradient (APG) method for minimizing strongly convex composite functions with unknown convexity parameters. This method incorporates a restarting scheme to automatically estimate the strong convexity parameter and achieves a nearly optimal iteration complexity. Then we consider the ℓ_1-regularized least-squares (ℓ_1-LS) problem in the high-dimensional setting. Although such an objective function is not strongly convex, it has restricted strong convexity over sparse vectors. We exploit this property by combining the adaptive APG method with a homotopy continuation scheme, which generates a sparse solution path towards optimality. This method obtains a global linear rate of convergence and its overall iteration complexity has a weaker dependency on the restricted condition number than previous work.

1. Introduction

We consider first-order methods for minimizing *composite* objective functions, i.e., the problem of

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \Big\{\phi(x) \triangleq f(x) + \Psi(x)\Big\}, \tag{1}$$

where $f(x)$ and $\Psi(x)$ are lower-semicontinuous, proper convex functions (Rockafellar, 1970, Section 7). We assume that f is differentiable on an open set containing $\operatorname{dom}\Psi$ and its gradient ∇f is Lipschitz continuous on $\operatorname{dom}\Psi$, i.e., there exists a constant L_f such that

$$\|\nabla f(x)-\nabla f(y)\|_2 \le L_f\|x-y\|_2,\ \forall\, x,y\in\operatorname{dom}\Psi. \tag{2}$$

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

We also assume $\Psi(x)$ is *simple* (Nesterov, 2013), meaning that for any $y\in\operatorname{dom}\Psi$, the following auxiliary problem can be solved efficiently or in closed-form:

$$T_L(y) = \arg\min_x \left\{\nabla f(y)^T x + \frac{L}{2}\|x-y\|_2^2 + \Psi(x)\right\}. \tag{3}$$

This is the case, e.g., when $\Psi(x)=\lambda\|x\|_1$ for any $\lambda>0$, or $\Psi(x)$ is the indicator function of a closed convex set that admits an easy projection mapping.

The so-called *proximal gradient* (PG) method simply uses (3) as its update rule: $x^{(k+1)} = T_L(x^{(k)})$, for $k=0,1,2,\ldots$, where L is set to L_f or determined by a linear search procedure. The iteration complexity for the PG method is $O(L_f/\epsilon)$ (Nesterov, 2004; 2013), which means, to obtain an ϵ-optimal solution (whose objective value is within ϵ of the optimum), the PG method needs $O(L_f/\epsilon)$ iterations. A far better iteration complexity, $O\big(\sqrt{L_f/\epsilon}\big)$, can be obtained by accelerated proximal gradient (APG) methods (Nesterov, 2013; Beck & Teboulle, 2009; Tseng, 2008).

The iteration complexities above imply that both PG and APG methods have a sublinear convergence rate. However, if f is strongly convex, i.e., there exists a constant $\mu_f>0$ (the *convexity parameter*) such that

$$f(x) \ \ge\ f(y) + \langle\nabla f(y), x-y\rangle + \frac{\mu_f}{2}\|x-y\|_2^2, \tag{4}$$

for all $x,y\in\operatorname{dom}\Psi$, then both PG and APG methods will achieve a linear convergence rate with the iteration complexities being $O(\kappa_f\log(1/\epsilon))$ and $O(\sqrt{\kappa_f}\log(1/\epsilon))$ (Nesterov, 2004; 2013), respectively. Here, $\kappa_f = L_f/\mu_f$ is called *condition number* of the function f. Since κ_f is typically a large number, the iteration complexity of the APG methods can be significanly better than that of the PG method for ill-conditioned problems. However, in order to obtain this better complexity, the APG methods need to use the convexity parameter μ_f, or a lower bound of it,

explicitly in their updates. In many applications, an effective lower bound of μ_f can be hard to estimate.

To address this problem, our first contribution in this paper is an adaptive APG method for solving problem (1) when f is strongly convex but μ_f is unknown. This method incorporates a restart scheme that can automatically estimate μ_f on the fly and achieves an iteration complexity of $O\left(\sqrt{\kappa_f}\log\kappa_f\cdot\log(1/\epsilon)\right)$.

Even if f is not strongly convex ($\mu_f = 0$), problem (1) may have special structure that may still allow the development of first-order methods with linear convergence. This is the case for the ℓ_1-*regularized least-squares* (ℓ_1-LS) problem, defined as

$$\underset{x}{\text{minimize}} \quad \phi_\lambda(x) \triangleq \frac{1}{2}\|Ax-b\|_2^2 + \lambda\|x\|_1, \qquad (5)$$

where $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ are the problem data, and $\lambda > 0$ is a regularization parameter. The problem has important applications in machine learning, signal processing, and statistics; see, e.g., Tibshirani (1996); Chen et al. (1998); Bruckstein et al. (2009). We are especially interested in solving this problem in the high-dimensional case ($m < n$) and when the solution, denoted as $x^\star(\lambda)$, is sparse.

In terms of the general model in (1), we have $f(x) = (1/2)\|Ax-b\|_2^2$ and $\Psi(x) = \lambda\|x\|_1$. Here $f(x)$ has a constant Hessian $\nabla^2 f(x) = A^TA$, and we have $L_f = \rho_{\max}(A^TA)$ and $\mu_f = \rho_{\min}(A^TA)$ where $\rho_{\max}(\cdot)$ and $\rho_{\min}(\cdot)$ denote the largest and smallest eigenvalues, respectively, of a symmetric matrix. Under the assumption $m < n$, the matrix A^TA is singular, hence $\mu_f = 0$ (i.e., f is not strongly convex). Therefore, we only expect sublinear convergence rates (at least globally) when using first-order optimization methods.

Nevertheless, even in the case of $m < n$, when the solution $x^\star(\lambda)$ is sparse, the PG method often exhibits fast convergence when it gets close to the optimal solution. Indeed, local linear convergence can be established for the PG method provided that the active submatrix (columns of A corresponding to the nonzero entries of the sparse iterates) is well conditioned (Luo & Tseng, 1992; Hale et al., 2008; Bredies & Lorenz, 2008). To explain this more formally, we define the *restricted eigenvalues* of A at the sparsity level s as

$$\begin{aligned}\rho_+(A,s) &= \sup\left\{\frac{x^TA^TAx}{x^Tx} : x\neq 0,\ \|x\|_0\le s\right\},\\ \rho_-(A,s) &= \inf\left\{\frac{x^TA^TAx}{x^Tx} : x\neq 0,\ \|x\|_0\le s\right\},\end{aligned} \qquad (6)$$

where s is a positive integer and $\|x\|_0$ denotes the number of nonzero entries of a vector $x \in \mathbb{R}^n$. From the above definitions, we have

$$\mu_f \ \le\ \rho_-(A,s) \ \le\ \rho_+(A,s) \ \le\ L_f, \qquad \forall\, s > 0.$$

As discussed before, we have $\mu_f = 0$ for $m < n$. But it is still possible that $\rho_-(A,s) > 0$ holds for some $s < m$. In this case, we say that the matrix A satisfies the *restricted eigenvalue condition* at the sparsity level s. Let $\mathrm{supp}(x) = \{j : x_j \neq 0\}$, and assume that $x, y \in \mathbb{R}^n$ satisfy $|\mathrm{supp}(x)\cup\mathrm{supp}(y)| \le s$. Then it can be shown (Xiao & Zhang, 2013, Lemma 3) that

$$f(x) \ \ge\ f(y) + \langle\nabla f(y), x-y\rangle + \frac{\rho_-(A,s)}{2}\|x-y\|_2^2.$$

The above inequality gives the notion of *restricted strong convexity* (cf. strong convexity defined in (4)). Intuitively, if the iterates of the PG method become sparse and their supports do not fluctuate much from each other, then restricted strong convexity leads to (local) linear convergence. This is exactly what happens when the PG method speeds up while getting close to the optimal solution.

Moreover, such a local linear convergence can be exploited by a homotopy continuation strategy to obtain much faster global convergence (Hale et al., 2008; Wright et al., 2009; Xiao & Zhang, 2013). The basic idea is to solve the ℓ_1-LS problem (5) with a large value of λ first, and then gradually decreases the value of λ until the target regularization is reached. For each value of λ, Xiao & Zhang (2013) employ the PG method to solve (5) up to an adequate precision, and then use the resulting approximate solution to warm start the PG method for (5) with the next value of λ. It is shown (Xiao & Zhang, 2013) that under suitable assumptions for sparse recovery (mainly the restricted eigenvalue condition), an appropriate homotopy strategy can ensure all iterates of the PG method be sparse, hence linear convergence at each stage can be established. As a result, the overall iteration complexity of such a proximal-gradient homotopy (PGH) method is $\widetilde{O}\big(\kappa_s\log(1/\epsilon)\big)$ where κ_s denotes the *restricted condition number* at some sparsity level $s > 0$, i.e.,

$$\kappa_s \ \triangleq\ \kappa(A,s) \ =\ \frac{\rho_+(A,s)}{\rho_-(A,s)}, \qquad (7)$$

and the notation $\widetilde{O}(\cdot)$ hides additional $\log(\kappa_s)$ factors.

Our second contribution in this paper is to show that, by using the adaptive APG method developed in this paper in a homotopy continuation scheme, we can further improve the iteration complexity for solving the ℓ_1-LS problem to $\widetilde{O}\big(\sqrt{\kappa_{s'}}\log(1/\epsilon)\big)$, where the sparsity level s' is slightly larger than the one for the PGH

method. We note that this result is not a trivial extension from the convergence results for the PGH method in Xiao & Zhang (2013). In particular, the adaptive APG method does not have the property of monotone decreasing, which was important for the analysis of the PGH method. In order to overcome this difficulty, we had to show a "non-blowout" property of our adaptive APG method, which is interesting in its own right.

2. An APG method for minimizing strongly convex functions

The main iteration of the APG method is based on a composite gradient mapping introduced by Nesterov in (Nesterov, 2013). For any fixed point y and a given constant $L > 0$, we define a local model of $\phi(x)$ around y using a quadratic approximation of f but keeping Ψ intact:

$$\psi_L(y; x) = f(y) + \nabla f(y)^T(x - y) + \frac{L}{2}\|x - y\|_2^2 + \Psi(x).$$

According to (3), we have

$$T_L(y) = \arg\min_x \ \psi_L(y; x). \tag{8}$$

Then the *composite gradient mapping* of f at y is defined as

$$g_L(y) = L(y - T_L(y)).$$

Following (Nesterov, 2013), we also define a local Lipschitz parameter

$$S_L(y) = \frac{\|\nabla f(T_L(y)) - \nabla f(y)\|_2}{\|T_L(y) - y\|_2}.$$

With the machinery of composite gradient mapping, Nesterov (2004; 2013) developed several variants of the APG methods. As discussed in the introduction, compared to the PG method, the iteration complexity of the accelerated methods have a better dependence on the accuracy ϵ when f is not strongly convex, and a better dependence on the condition number κ_f when f is strongly convex. However, in contrast with the PG method, the better complexity bound of the APG method in the strongly convex case relies on the knowledge of the convexity parameter μ_f, or an effective lower bound of it, both of which can be hard to obtain in practice.

To address this problem, we propose an adaptive APG method that can be applied without knowing μ_f and still obtains a linear convergence rate. To do so, we first present an APG method in Algorithm 1 and in Algorithm 2 upon which the development of the adaptive APG method is based. We name this method scAPG, where "sc" stands for "strongly convex."

Algorithm 1 $\{\hat{x}, \hat{M}\} \leftarrow \texttt{scAPG}(x^{(0)}, L_0, \mu, \hat{\epsilon})$

parameter: $L_{\min} \geq \mu > 0$, $\gamma_{\text{dec}} \geq 1$
$x^{(-1)} \leftarrow x^{(0)}$
$\alpha_{-1} = 1$
repeat
  (for $k = 0, 1, 2, \ldots$)
  $\{x^{(k+1)}, M_k, \alpha_k, g^{(k)}, S_k\}$
    $\leftarrow \texttt{AccelLineSearch}(x^{(k)}, x^{(k-1)}, L_k, \mu, \alpha_{k-1})$
  $L_{k+1} \leftarrow \max\{L_{\min}, M_k/\gamma_{\text{dec}}\}$
until $\omega(x^{(k+1)}) \leq \hat{\epsilon}$
$\hat{x} \leftarrow x^{(k+1)}$
$\hat{M} \leftarrow M_k$

Algorithm 2 $\{x^{(k+1)}, M_k, \alpha_k, g^{(k)}, S_k\}$
  $\leftarrow \texttt{AccelLineSearch}(x^{(k)}, x^{(k-1)}, L_k, \mu, \alpha_{k-1})$

parameter: $\gamma_{\text{inc}} > 1$
$L \leftarrow L_k/\gamma_{\text{inc}}$
repeat
  $L \leftarrow L\gamma_{\text{inc}}$
  $\alpha_k \leftarrow \sqrt{\frac{\mu}{L}}$
  $y^{(k)} \leftarrow x^{(k)} + \frac{\alpha_k(1-\alpha_{k-1})}{\alpha_{k-1}(1+\alpha_k)}(x^{(k)} - x^{(k-1)})$
  $x^{(k+1)} \leftarrow T_L(y^{(k)})$
until $\phi(x^{(k+1)}) \leq \psi_L(y^{(k)}; x^{(k+1)})$
$M_k \leftarrow L$
$g^{(k)} \leftarrow M_k(y^{(k)} - x^{(k+1)})$
$S_k \leftarrow S_L(y^{(k)})$

To use this algorithm, we need to first choose an initial optimistic estimate $L_{\min}$ for the Lipschitz constant L_f: $0 < L_{\min} \leq L_f$, and two adjustment parameters $\gamma_{\text{dec}} \geq 1$ and $\gamma_{\text{inc}} > 1$. In addition, this method requires an input parameter $\mu > 0$, which is an estimate of the true convexity parameter μ_f. The scAPG method generates the following three sequences:

$$\begin{aligned} \alpha_k &= \sqrt{\frac{\mu}{M_k}}, \\ y^{(k)} &= x^{(k)} + \frac{\alpha_k(1-\alpha_{k-1})}{\alpha_{k-1}(1+\alpha_k)}(x^{(k)} - x^{(k-1)}), \\ x^{(k+1)} &= T_{M_k}(y^{(k)}). \end{aligned} \tag{9}$$

where M_k is found by the line-search procedure in Algorithm 2. The line search procedure starts with an estimated Lipschitz constant L_k, and increases its value by the factor γ_{inc} until $\phi(x^{(k+1)}) \leq \psi_{M_k}(y^{(k)}; x^{(k+1)})$, which is sufficient to guarantee the convergence. In each iteration of Algorithm 1, the scAPG method tries to start the line search at a smaller initial value by setting L_{k+1} to be $\min\{L_{\min}, M_k/\gamma_{\text{dec}}\}$.

The scAPG algorithm can be considered as an extension of the constant step scheme of Nesterov (2004) for

minimizing composite functions in (1) when $\mu_f > 0$. Indeed, if $M_k = L_f$, we have $\alpha_k = \sqrt{\mu_f/L_f}$ for all k and the update for $y^{(k)}$ becomes

$$y^{(k)} = x^{(k)} + \frac{\sqrt{L_f} - \sqrt{\mu_f}}{\sqrt{L_f} + \sqrt{\mu_f}}(x^{(k)} - x^{(k-1)}), \qquad (10)$$

which is the same as Algorithm (2.2.11) in Nesterov (2004). Note that, one can not directly apply Algorithm 1 or Nesterov's constant scheme to problems without strongly convexity by simply setting $\mu = 0$.

Another difference from Nesterov's method is that Algorithm 1 has an explicit stopping criterion based on the *optimality residue* $\omega(x^{(k+1)})$, which is defined as

$$\omega(x) \triangleq \min_{\xi \in \partial\Psi(x)} \|\nabla f(x) + \xi\|_\infty, \qquad (11)$$

where $\partial\Psi(x)$ is the subdifferential of Ψ at x. The optimality residue measures how close a solution x is to the optimality condition of (1) in the sense that $\omega(x^\star) = 0$ if and only if $x^\star$ is an solution to (1).

The following theorem states that, if μ is a positive lower bound of μ_f, the scAPG converges geometrically and it has an iteration complexity $O(\sqrt{\kappa_f}\log(1/\epsilon))$.

Theorem 1. *Suppose $x^\star$ is the optimal solution of (1) and $0 < \mu \le \mu_f$. Then Algorithm 1 guarantees that*

$$\phi(x^{(k)}) - \phi(x^\star) \le \tau_k\left[\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\|x^{(0)} - x^\star\|_2^2\right], \qquad (12)$$

$$\frac{\mu}{2}\|y^{(k)} - x^\star\|_2^2 \le \tau_k\left[\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\|x^{(0)} - x^\star\|_2^2\right], \qquad (13)$$

where

$$\tau_k = \begin{cases} 1 & k = 0, \\ \prod_{i=0}^{k-1}(1-\alpha_i) & k \ge 1. \end{cases} \qquad (14)$$

Moreover,

$$\tau_k \le \left(1 - \sqrt{\frac{\mu}{L_f\gamma_{\text{inc}}}}\right)^k.$$

In addition to the geometric convergence of $\phi(x^{(k)})$, this theorem states that the auxiliary sequence $y^{(k)}$ also converges to the unique optimizer $x^\star$ with a geometric rate.

If μ does not satisfies $\mu \le \mu_f$, Theorem 1 may not hold anymore. However, we can show that, in this case, Algorithm 1 will at least not blowout. More precisely, we show that $\phi(x^{(k)}) \le \phi(x^{(0)})$ for all $k \ge 1$ as long as $\mu \le L_{\min}$, which can be easily enforced in implementation of the algorithm.

Lemma 1. *Suppose $0 < \mu \le L_{\min}$. Then Algorithm 1 guarantees that*

$$\phi(x^{(k+1)}) \le \phi(x^{(0)}) - \frac{M_k}{2}\|x^{(k+1)} - x^{(k)}\|_2^2. \qquad (15)$$

The non-blowout property is also critical in our analysis of the homotopy method for solving the ℓ_1-LS problem presented in Section 4. In particular, it helps to show the sparsity of $x^{(k)}$ once $x^{(0)}$ is sparse. (All proofs for our results are given in the supporting materials).

3. An Adaptive APG method with restart

When applied to strongly convex minimization problems, Nesterov's constant step scheme (10) needs to use L_f and μ_f as input parameters. Thanks to the line-search technique, Algorithm 1 does not need to know L_f explicitly. However, it still need to know the convexity parameter μ_f or a nontrivial lower bound of it in order to guarantee the geometric convergence rate given in Theorem 1.

Compared to line search on L_f, estimating μ_f on-the-fly is much more sophisticated. Nesterov (2013) suggested a restarting scheme to estimate μ_f, which does not require any lower bound of μ_f, and can be shown to have linear convergence (up to a logarithmic factor). In this section, we adapt his restarting technique to Algorithm 1 and obtain an adaptive APG method. This method has the same convergence guarantees as Nesterov's scheme. However, there are two important differences, which we will elaborate on at the end of this section.

We first describe the basic idea of the restart scheme for estimating μ_f. Suppose we simply run Algorithm 1 with a guessed value μ. At each iteration, we can check if the inequality (12) is satisfied. If not, we must have $\mu > \mu_f$ according to Theorem 1, and therefore need to reduce μ to ensure Algorithm 1 converges in a linear rate. However, (12) can not be evaluated because $x^\star$ is unknown. Fortunately, we can show in the following lemma that, if $\mu \le \mu_f$, the norm of the gradient mapping $g^{(k)} = g_{M_k}(y^{(k)})$ generated in Algorithm 1 also decreases at a linear rate.

Lemma 2. *Suppose $0 < \mu \le \mu_f$ and the initial point $x^{(0)}$ of Algorithm 1 is obtained by calling Algorithm 2, i.e., $\{x^{(0)}, M_{-1}, \alpha_{-1}, g^{(-1)}, S_{-1}\} \leftarrow$ `AccelLineSearch`$(x^{\text{ini}}, x^{\text{ini}}, L_{\text{ini}}, \mu, 1)$ with an arbitrary $x^{\text{ini}} \in \mathbb{R}^n$ and $L_{\text{ini}} \ge L_{\min}$. Then, for any $k \ge 0$ in Algorithm 1, we have*

$$\|g_{M_k}(y^{(k)})\|_2 \le 2\sqrt{2\tau_k}\,\frac{M_k}{\mu}\left(1 + \frac{S_{-1}}{M_{-1}}\right)\|g^{(-1)}\|_2. \qquad (16)$$

Algorithm 3 $\{\hat{x}, \hat{M}, \hat{\mu}\} \leftarrow \texttt{AdapAPG}\,(x^{\rm ini}, L_{\rm ini}, \mu_0, \hat{\epsilon})$

parameter: $L_{\min} \geq \mu_0,\ \gamma_{\rm dec} \geq 1,\ \gamma_{\rm sc} > 1,\ \theta_{\rm sc} \in (0,1)$
$\{x^{(0)}, M_{-1}, \alpha_{-1}, g^{(-1)}, S_{-1}\}$
$\quad \leftarrow \texttt{AccelLineSearch}(x^{\rm ini}, x^{\rm ini}, L_{\rm ini}, \mu_0, 1)$
$x^{(-1)} \leftarrow x^{(0)},\ L_{-1} \leftarrow M_{-1},\ \mu \leftarrow \mu_0$
$\alpha_{-1} \leftarrow 1,\ \tau_0 \leftarrow 1, k \leftarrow 0$
repeat
  $\{x^{(k+1)}, M_k, \alpha_k, g^{(k)}, S_k\}$
  $\quad \leftarrow \texttt{AccelLineSearch}(x^{(k)}, x^{(k-1)}, L_k, \mu, \alpha_{k-1})$
  $\tau_{k+1} \leftarrow \tau_k(1-\alpha_k)$
  if *condition A holds,* **then**
    $x^{(0)} \leftarrow x^{(k+1)},\ x^{(-1)} \leftarrow x^{(k+1)},\ L_{-1} = M_k$
    $g^{(-1)} \leftarrow g^{(k)},\ M_{-1} \leftarrow M_k,\ S_{-1} \leftarrow S_k$
    $k \leftarrow 0$
  else
    if *condition B holds,* **then**
      $\mu \leftarrow \mu/\gamma_{\rm sc}$
      $k \leftarrow 0$
    else
      $L_{k+1} \leftarrow \max\{L_{\min}, M_k/\gamma_{\rm dec}\}$
      $k \leftarrow k+1$
    end if
  end if
until $\omega(x^{(k+1)}) \leq \hat{\epsilon}$
$\hat{x} \leftarrow x^{(k+1)},\ \hat{M} \leftarrow M_k,\ \hat{\mu} \leftarrow \mu$

Unlike the inequality (12), the inequality (16) can be checked explicitly and, if it does not hold, we know $\mu > \mu_f$ and need to reduce μ.

Now we are ready to develop the adaptive APG method. Let $\theta_{\rm sc} \in (0,1)$ be a desired shrinking factor. We check the following two conditions at iteration k of Algorithm 1:

- A: $\left\|g_{M_k}(y^{(k)})\right\|_2 \leq \theta_{\rm sc}\left\|g^{(-1)}\right\|_2$.
- B: $2\sqrt{2\tau_k}\,\frac{M_k}{\mu}\left(1+\frac{S_{-1}}{M_{-1}}\right) \leq \theta_{\rm sc}$.

If A is satisfied first, then we restart Algorithm 1 with $x^{(k+1)}$ as the new starting point, set $k=0$, and update the three quantities $g^{(-1)}$, S_{-1} and M_{-1} accordingly (again use $\alpha_{-1}=1$ and $\tau_0=1$). If A is not satisfied but B is satisfied first, it means that μ is larger than μ_f. In fact, if $\mu \leq \mu_f$, then combining condition B with Lemma 2 would imply that A also holds. This contradiction indicates that if B is satisfied first, we must have $\mu > \mu_f$, and we have to reduce μ, say by a factor $\gamma_{\rm sc} > 1$. In this case, we restart Algorithm 1 still at $x^{(0)}$ and keep $g^{(-1)}$, S_{-1} and M_{-1} unchanged. If neither conditions are satisfied, we continue Algorithm 1 to its next iterate until the optimality residue is smaller than a prescribed value. We present the above procedure formally in Algorithm 3, whose iteration complexity is given by the following theorem.

Theorem 2. *Assume $\mu_0 > \mu_f > 0$. Let $g^{\rm ini}$ denotes the first $g^{(-1)}$ computed by Algorithm 3, and $N_{\rm A}$ and $N_{\rm B}$ the number of times that conditions A and B are satisfied, respectively. Then $N_{\rm A} \leq \left\lceil \log_{1/\theta_{sc}}\left(\left(1+\frac{L_f}{L_{\min}}\right)\frac{\|g^{\rm ini}\|_2}{\hat{\epsilon}}\right)\right\rceil$ and $N_{\rm B} \leq \left\lceil \log_{\gamma_{\rm sc}}\left(\frac{\mu_0}{\mu_f}\right)\right\rceil$ and the total number of iterations is at most*

$$(N_{\rm A}+N_{\rm B})\sqrt{\frac{L_f\gamma_{\rm inc}\gamma_{sc}}{\mu_f}}\,\ln\left(8\left(\frac{L_f\gamma_{\rm inc}\gamma_{sc}}{\mu_f\theta_{sc}}\right)^2\left(1+\frac{L_f}{L_{\min}}\right)^2\right).$$

Note that if $0 < \mu_0 \leq \mu_f$, then $N_B = 0$.

The total number of iterations given in Theorem 2 is asymptotically

$$O\left(\kappa_f^{1/2}\log(\kappa_f)\log\left(\frac{\kappa_f}{\hat{\epsilon}}\right)\right) + O\left(\kappa_f^{1/2}\log(\kappa_f)\right).$$

This is the same complexity as for the restart scheme proposed by Nesterov for his accelerated dual gradient (ADG) method (Nesterov, 2013, Section 5.3). Despite using a similar restart scheme and having the same complexity bound, here we elaborate on some important differences between our method from Nesterov's.

- Nesterov's ADG method exploits strong convexity in Ψ instead of f. In order to use it under our assumption (that f is strongly convex), one needs to relocate a strong convexity term from f to Ψ, and this relocated term needs to be adjusted whenever the estimate μ is reduced.
- The restart scheme suggested in (Nesterov, 2013, Section 5.3) uses an extra line-search at each iteration, solely for the purpose of computing the gradient mapping at $x^{(k)}$. Our method directly use the gradient mapping at $y^{(k)}$, which does not require the extra line-search, therefore the computational cost per iteration is lower.

4. Homotopy continuation for sparse optimization

In this section, we focus on the ℓ_1-regularized least-squares (ℓ_1-LS) problem (5) in the high-dimensional setting i.e., with $m < n$. This is a special case of (1), but the function $f(x) = (1/2)\|Ax-b\|_2^2$ is not strongly convex when $m < n$. Therefore, we only expect a sublinear convergence rate (at least globally) when using traditional first-order optimization methods.

Nevertheless, as explained in the introduction, one can use a homotopy continuation strategy to obtain much faster convergence. The key idea is to solve the ℓ_1-LS

Algorithm 4 $\hat{x}^{(\mathrm{tgt})} \leftarrow \texttt{APGHomotopy}(A, b, \lambda_{\mathrm{tgt}}, \epsilon, L_0, \hat{\mu}_0)$

input: $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^n$, $L_0 \geq \hat{\mu}_0 > 0$
parameter: $\eta \in (0,1)$, $\delta \in (0,1)$
initialize: $\lambda_0 \leftarrow \|A^T b\|_\infty$, $\hat{x}^{(0)} \leftarrow 0$, $\hat{M}_0 \leftarrow L_0$
$N \leftarrow \lfloor \ln(\lambda_0/\lambda_{\mathrm{tgt}}) / \ln(1/\eta) \rfloor$
for $K = 0, 1, 2, \ldots, N-1$ **do**
  $\lambda_{K+1} \leftarrow \eta\lambda_K$
  $\hat{\epsilon}_{K+1} \leftarrow \delta\lambda_{K+1}$
  $\{\hat{x}^{(K+1)}, \hat{M}_{K+1}, \hat{\mu}_{K+1}\}$
    $\leftarrow \texttt{AdapAPG}\big(\hat{x}^{(K)}, \hat{M}_K, \hat{\mu}_K, \hat{\epsilon}_{K+1}, \lambda_{K+1}\big)$
end for
$\{\hat{x}^{(\mathrm{tgt})}, \hat{M}_{\mathrm{tgt}}\} \leftarrow \texttt{AdapAPG}\big(\hat{x}^{(N)}, \hat{M}_N, \hat{\mu}_N, \epsilon, \lambda_{\mathrm{tgt}}\big)$
return: $\hat{x}^{(\mathrm{tgt})}$

problem with a large regularization parameter λ_0 first, and then gradually decreases the value of λ until the target regularization is reached. For a fixed λ, adaptive APG method (Algorithm 3) is employed to solve the ℓ_1-LS problem up to an adequate precision, then the solution is used to warm start the next stage. We show that such a homotopy scheme guarantees that all iterates generated are sufficiently sparse, which implies restricted strong convexity. As a result, a linear rate of convergence can be established for each homotopy stage, and the overall complexity is $\widetilde{O}\big(\sqrt{\kappa_{s'}}\log(1/\epsilon)\big)$ with s' slightly larger than s in the complexity of PGH.

The APG homotopy method is presented in Algorithm 4. To avoid confusion over the notations, we use λ_{tgt} to denote the target regularization parameter in (5). The method starts with $\lambda_0 = \|A^T b\|_\infty$ which is the smallestλ such that the ℓ_1-LS problem has the trivial solution 0 (by examining the optimality condition). This method has two extra parameters $\eta \in (0,1)$ and $\delta \in (0,1)$. They control the algorithm as follows: The sequence of values for the regularization parameter is determined as $\lambda_k = \eta^k \lambda_0$ for $k = 1, 2, \ldots$, until the target value λ_{tgt} is reached. For each λ_k except λ_{tgt}, we solve problem (5) with a proportional precision $\delta\lambda_k$. For the last stage with λ_{tgt}, we solve to the absolute precision ϵ.

Our convergence analysis of the APG homotopy method is based on the following assumption, which involves the restricted eigenvalues defined in (6).

Assumption 1. *Suppose $b = A\bar{x} + z$. Let $\bar{S} = \mathrm{supp}(\bar{x})$ and $\bar{s} = |\bar{S}|$. There exist $\gamma > 0$ and $\delta' \in (0, 0.2]$ such that $\gamma > (1+\delta')/(1-\delta')$ and*

$$\lambda_{\mathrm{tgt}} \;\geq\; 4\max\left\{2,\; \frac{\gamma+1}{(1-\delta')\gamma - (1+\delta')}\right\}\|A^T z\|_\infty. \tag{17}$$

Moreover, we assume there exists an integer $\tilde{s}$ such that $\rho_-(A, \bar{s}+3\tilde{s}) > 0$ and

$$\tilde{s} \;>\; \frac{24\big(\gamma_{\mathrm{inc}}\rho_+(A, \bar{s}+3\tilde{s}) + 3\rho_+(A, \tilde{s})\big)}{\rho_-(A, \bar{s}+\tilde{s})}(1+\gamma)\bar{s}. \tag{18}$$

We also assume that $L_{\min} \leq \gamma_{\mathrm{inc}}\rho_+(A, \bar{s}+3\tilde{s})$.

We will show that by choosing the parameters η and δ in Algorithm 4 appropriately, these conditions also imply that all iterates along the solution path are sparse. We note that Assumption 1 is very similar to Assumption 1 in Xiao & Zhang (2013) (they differ only in the constants in the conditions), and interpretations and remarks made there also apply here. More specifically,

- The existence of $\tilde{s}$ satisfying the conditions like (18) is necessary and standard in sparse recovery analysis. It is closely related to the restricted isometry property (RIP) of Candès & Tao (2005) which assumes that there exist some $s > 0$, and $\nu \in (0,1)$ such that $\kappa(A, s) < (1+\nu)/(1-\nu)$.

- The RIP-like condition (18) can be much stronger than the corresponding conditions established in the sparse recovery literature (see, e.g., Li & Mo (2011) and references therein), which are only concerned about the recovery property of the optimal solution $x^\star$. In contrast, our condition needs to guarantee sparsity for all iterates along the solution path, thus is "dynamic" in nature. In particular, in addition to the matrix A, it also depends on algorithmic parameters γ_{inc}, η and δ (Theorem 4 will relate η to δ and δ').

Our first result below concerns the local linear convergence of Algorithm 3 when applied to solve the ℓ_1-LS problem at each stage of the homotopy method. Basically, if the starting point $x^{(0)}$ is sparse and the optimality condition is satisfied with adequate precision, then all iterates along the solution path are sparse. This implies that restricted strong convexity holds and Algorithm 3 actually has linear convergence.

Theorem 3. *Suppose Assumption 1 holds. If the initial point x^{ini} in Algorithm 3 satisfies*

$$\big\|x^{\mathrm{ini}}_{\bar{S}^c}\big\|_0 \leq \tilde{s}, \qquad \omega(x^{\mathrm{ini}}) \leq \delta'\lambda, \tag{19}$$

then for all $k \geq 0$, we have $\big\|x^{(k)}_{\bar{S}^c}\big\|_0 \leq \tilde{s}$. Moreover, all the three conclusions of Theorem 2 holds by replacing L_f and μ_f with $\rho_+(A, \bar{s}+3\tilde{s})$ and $\rho_-(A, \bar{s}+3\tilde{s})$, respectively.

Our next result gives the overall iteration complexity of the APG homotopy method in Algorithm 4. To

simplify presentation, we let $s' = \bar{s} + 3\tilde{s}$, and use the following notations:

$$\begin{aligned} \rho_+(s') &= \rho_+(A, \bar{s} + 3\tilde{s}), \\ \rho_-(s') &= \rho_-(A, \bar{s} + 3\tilde{s}), \\ \kappa_{s'} &= \kappa(A, \bar{s} + 3\tilde{s}) = \frac{\rho_+(A, \bar{s} + 3\tilde{s})}{\rho_-(A, \bar{s} + 3\tilde{s})}. \end{aligned}$$

Roughly speaking, if the parameters δ and η are chosen appropriately, then the total number of proximal-gradient steps in Algorithm 4 for finding an ϵ-optimal solution is $\widetilde{O}(\sqrt{\kappa_{s'}}\ln(1/\epsilon))$, which improves the $\widetilde{O}(\kappa_s \ln(1/\epsilon))$ complexity of PGH in the dependence on restricted condition number.

Theorem 4. *Suppose Assumption 1 holds for some δ', γ and $\tilde{s}$, and the parameters δ and η in Algorithm 4 are chosen such that $\frac{1+\delta}{1+\delta'} \le \eta < 1$. Let $N = \lfloor \ln(\lambda_0/\lambda_{\rm tgt}) / \ln \eta^{-1} \rfloor$ as in the algorithm. Then:*

1. *Condition (19) holds for each call of Algorithm 3. For $K = 0, \ldots, N-1$, the number of gradient steps in each call of Algorithm 3 is no more than*

$$\left(\log_{\frac{1}{\theta_{sc}}}\left(\frac{C}{\delta}\right) + D\right)\sqrt{\kappa_{s'}\gamma_{\rm inc}\gamma_{sc}} \times \ln\left(8\left(\frac{\kappa_{s'}\gamma_{\rm inc}\gamma_{sc}}{\theta_{sc}}\right)^2\left(1 + \frac{\rho_+(s')}{L_{\min}}\right)^2\right),$$

where $C = \left(1 + \frac{\rho_+(s')}{L_{\min}}\right)\sqrt{8\gamma_{\rm inc}\kappa_{s'}(1+\gamma)\bar{s}}$ and $D = \left\lceil \log_{\gamma_{sc}}\left(\frac{\hat{\mu}_0}{\rho_-(s')}\right)\right\rceil + 1$. It is independent of λ_K.

2. *For each $K \ge 0$, the outer iterates $\hat{x}^{(K)}$ satisfies*

$$\phi_{\lambda_{\rm tgt}}(\hat{x}^{(K)}) - \phi^\star_{\lambda_{\rm tgt}} \le \eta^{2(K+1)}\frac{4.5(1+\gamma)\lambda_0^2\bar{s}}{\rho_-(A, \bar{s}+\tilde{s})},$$

and the following bound on sparse recovery holds

$$\|\hat{x}^{(K)} - \bar{x}\|_2 \le \eta^{K+1}\frac{2\lambda_0\sqrt{\bar{s}}}{\rho_-(A, \bar{s}+\tilde{s})}.$$

3. *When Algorithm 4 terminates, the total number of proximal-gradient steps is $\widetilde{O}\left(\sqrt{\kappa_{s'}}\ln(1/\epsilon)\right)$, Moreover, the output $\hat{x}^{(\rm tgt)}$ satisfies*

$$\phi_{\lambda_{\rm tgt}}(\hat{x}^{(\rm tgt)}) - \phi^\star_{\lambda_{\rm tgt}} \le \frac{4(1+\gamma)\lambda_{\rm tgt}\bar{s}}{\rho_-(A, \bar{s}+\tilde{s})}\epsilon.$$

We note that our result is not a simple extension of those in Xiao & Zhang (2013). In particular, the AdapAPG method do not have the property of monotone decreasing, which is key for establishing the complexity of the PGH method in Xiao & Zhang (2013). Instead, our proof relies on the non-blowout property (Lemma 1) to show that all iterates along the solution path are sparse (details are given in the supporting materials).

5. Numerical experiments

In this section, we present preliminary numerical experiments to support our theoretical analysis. In addition to the PG and PGH methods (Xiao & Zhang, 2013), we also compare our method with FISTA (Beck & Teboulle, 2009) and its homotopy variants.

We implemented FISTA with an adaptive line-search over the Lipschitz constant L_f, but it does not use or estimate the convexity parameter μ_f. Hence it has a sublinear complexity $O(\sqrt{L_f/\epsilon})$. In our experiments, we also compare with a simple restart scheme for FISTA suggested by O'Donoghue & Candès (2012): restart FISTA whenever it exhibits nonmonotone behaviors. In particular, we implemented the *gradient* scheme: restart whenever $g_{L_k}(y^{(k-1)})^T(x^{(k)} - x^{(k-1)}) > 0$, where $x^{(k)}$ and $y^{(k)}$ are two sequences generated by FISTA, similar to those in our AdapAPG method. O'Donoghue & Candès (2012) show that for strongly convex pure quadratic functions, this restart scheme leads to the optimal complexity of $O\left(\sqrt{\kappa_f}\ln(1/\epsilon)\right)$. However, their analysis does *not* hold for the ℓ_1-LS problem or other non-quadratic functions. We call this method FISTA+RS (meaning FISTA with ReStart).

For our AdapAPG method (Algorithm 3) and APG homotopy method (Algorithm 4), we use the following values of the parameters unless otherwise stated:

parameters	$\gamma_{\rm inc}$	$\gamma_{\rm dec}$	$\theta_{\rm sc}$	$\gamma_{\rm sc}$	η	δ
values	2	2	0.1	10	0.8	0.2

To make the comparison clear, we generate an ill-conditioned random matrix A following the experimental setup in Agarwal et al. (2012):

- Generate a random matrix $B \in \mathbb{R}^{m\times n}$ with B_{ij} following i.i.d. standard normal distribution.
- Choose $\omega \in [0, 1)$, and for $i = 1, \ldots, m$, generate each row $A_{i,:}$ by $A_{i,1} = B_{i,1}/\sqrt{1-\omega^2}$ and $A_{i,j+1} = \omega A_{i,j} + B_{i,j}$ for $j = 2, \ldots, n$.

It can be shown that the eigenvalues of $\mathbf{E}[A^TA]$ lie within the interval $\left[\frac{1}{(1+\omega)^2}, \frac{2}{(1-\omega)^2(1+\omega)}\right]$. If $\omega = 0$, then $A = B$ and the covariance matrix A^TA is well conditioned. As $\omega \to 1$, it becomes progressively more ill-conditioned. In our experiments, we generate the matrix A with $m = 1000$, $n = 5000$, and $\omega = 0.9$.

Figure 1 shows the computational results of the four different methods: PG, FISTA, FISTA+RS, AdapAPG, and their homotopy continuation variants (denoted by "+H"). For each method, we initialize the Lipschitz constant by $L_0 = \max_{j\in\{1,\ldots,n\}}\|A_{:,j}\|_2^2$. For the AdapAPG method, we initialize the estimate of convexity

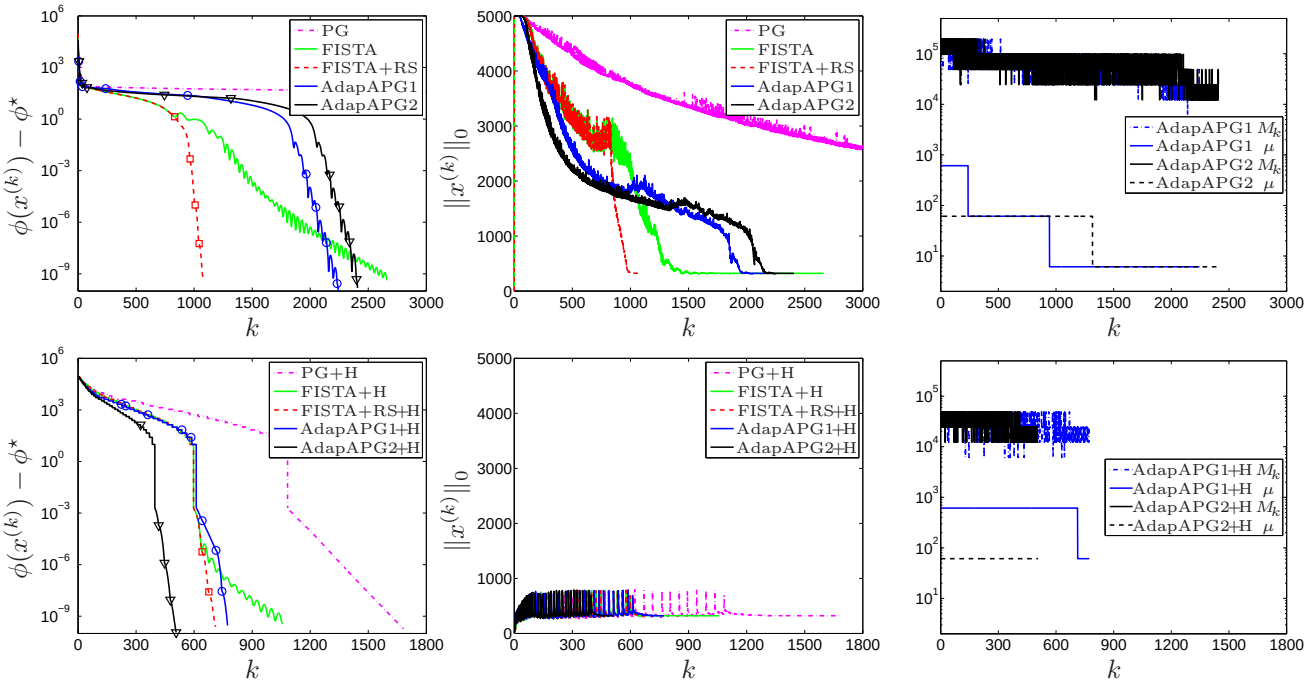

Figure 1. Solving an ill-conditioned ℓ_1-LS problem. AdapAPG1 starts with $\mu_0 = L_0/10$, and AdapAPG2 uses $\mu_0 = L_0/100$.

parameter with two different values, $\mu_0 = L_0/10$ and $\mu_0 = L_0/100$, and denote their results by AdapAPG1 and AdapAPG2, respectively.

From the top-left plot, we observe that PG, FISTA+RS and AdapAPG all go through a slow plateau before reaching fast local linear convergence. FISTA without restart does not exploit the strong convexity and is the slowest asymptotically. Their homotopy continuation variants shown in the bottom-left plot are much faster. Each vertical jump on the curves indicates a change in the value of λ in the homotopy scheme. In particular, it is clear that all except FISTA+H enter the final homotopy stage with fast linear convergence. In the final stage, the PGH method has a rather flat slope due to ill-conditioning of the A matrix; in contrast, FISTA+RS and AdapAPG have much steeper slopes due to their accelerated schemes. AdapAPG1 started with a modest slope, and then detected that the μ value was too big and reduced it by a factor of $\gamma_{\text{sc}} = 10$, which resulted in the same fast convergence rate as AdapAPG2 after that.

The two plots in the middle show the sparsity of each iterates along the solution paths of these methods. We observe that FISTA+RS and AdapAPG entered fast local convergence precisely when their iterates became sufficiently sparse, i.e., when $\|x^{(k)}\|_0$ became close to that of the final solution. In contrast, the homotopy variants of these algorithms kept all iterates sparse by using the warm start from previous stages. Therefore, restricted strong convexity hold along the whole path and linear convergence was maintained at each stage.

The right column shows the automatic tuning of the local Lipschitz constant M_k and the restricted convexity parameter μ. We see that the homotopy methods (bottom-right plot) have relatively smaller M_k and larger μ than the ones without using homotopy continuation (top-right plot), which means much better conditioning along the iterates. In particular, the homotopy AdapAPG method used fewer number of reductions of μ, for both initializations of μ_0.

Overall, we observe that for the ℓ_1-LS problem, the homotopy continuation scheme is very effective in speeding up different methods. Even with the overhead of estimating and tuning μ, the AdapAPG+H method is close in efficiency compared with the FISTA+RS+H method. If the initial guess of μ is not far off, then AdapAPG+H gives the best performance. Finally, we note that unlike the AdapAPG method, the optimal complexity of the FISTA+RS method has not been established for minimizing general strongly convex functions (including ℓ_1-LS). Although often quite competitive in practice, we have observed non-quadratic cases in which FISTA+RS demonstrate less desirable convergence (see examples in the supporting materials and also comments in O'Donoghue & Candès (2012)).

References

Agarwal, A., Negahban, S. N., and Wainwright, M. J. Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, 40(5):2452–2482, 2012.

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

Bredies, K. and Lorenz, D. A. Linear convergence of iterative soft-thresholding. *Journal of Fourier Analysis and Applications*, 14:813–837, 2008.

Bruckstein, A. M., Donoho, D. L., and Elad, M. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51 (1):34–81, 2009.

Candès, E. J. and Tao, T. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, December 2005.

Chen, S. S., Donoho, D. L., and Saunders, M. A. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

Hale, E. T., Yin, W., and Zhang, Y. Fixed-point continuation for ℓ_1-minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19(3): 1107–1130, 2008.

Li, S. and Mo, Q. New bounds on the restricted isometry constant δ_{2k}. *Applied and Computational Harmonic Analysis*, 31(3):460–468, 2011.

Luo, Z.-Q. and Tseng, P. On the linear convergence of descent methods for convex essentially smooth minimization. *SIAM Journal on Control and Optimization*, 30(2):408–425, 1992.

Nesterov, Y. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, Boston, 2004.

Nesterov, Y. Gradient methods for minimizing composite objective function. CORE discussion paper 2007/76, Center for Operations Research and Econometrics, Catholic University of Louvain, Belgium, September 2007.

Nesterov, Y. Gradient methods for minimizing composite functions. *Mathematical Programming, Series B*, 140:125–161, 2013.

O'Donoghue, B. and Candès, E. J. Adaptive restart for accelerated gradient schemes. Manuscript, April 2012. To appear in *Foundations of Computational Mathematics*.

Rockafellar, R. T. *Convex Analysis*. Princeton University Press, 1970.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58:267–288, 1996.

Tseng, P. On accelerated proximal gradient methods for convex-concave optimization. Manuscript, 2008.

Wright, S. J., Nowad, R. D., and Figueiredo, M. A. T. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, July 2009.

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

6. Appendix

6.1. Proof of Theorem 1

We will also need the following notations and results to prove Theorem 1.

Lemma 3. *Suppose $\mu \le \mu_f$ and the inequality $\phi(T_L(y)) \le \psi_L(y; T_L(y))$ holds for y. Then, for any $x \in \mathbb{R}^n$, we have*

$$\phi(x) \ge \phi(T_L(y)) + \langle g_L(y), x - y\rangle + \frac{1}{2L}\|g_L(y)\|^2 + \frac{\mu}{2}\|x - y\|^2. \tag{20}$$

We omit the proof of this lemma since it is almost identical to that of (Nesterov, 2004, Theorem 2.2.7), in which Ψ is restricted to be the indicator function of a closed convex set. A variant of this lemma corresponding to $\mu = 0$ appeared in (Beck & Teboulle, 2009, Lemma 2.3).

The proof of Theorem 1 is based on the notion of *estimate sequence* developed by Nesterov (Nesterov, 2004). We first give its definition and a few lemmas that are necessary for our proof.

Definition 1. *(Nesterov, 2004, Definition 2.2.1) A pair of sequences $\{V_k(x)\}_{k\ge0}$ and $\{\tau_k\}_{k\ge0}$, $\tau_k \ge 0$, is called an estimate sequence of the function $\phi(x)$ if*

$$\tau_k \to 0$$

and for any $x \in \mathbb{R}^n$ and all $k \ge 0$, we have

$$V_k(x) \;\le\; (1 - \tau_k)\phi(x) + \tau_k V_0(x). \tag{21}$$

Lemma 4. *(Nesterov, 2004, Lemma 2.2.1) Suppose $x^\star$ is an optimal solution to (1). Let the pair $\{V_k(x)\}_{k\ge0}$ and $\{\tau_k\}_{k\ge0}$ be an estimate sequence of $\phi(x)$. If we have some sequence $\{x_k\}_{k\ge0}$ satisfying*

$$\phi(x^{(k)}) \le V_k^\star := \min_{x\in\mathbb{R}^n} V_k(x), \tag{22}$$

then

$$\phi(x^{(k)}) - \phi(x^\star) \le \tau_k \left[V_0(x^\star) - \phi(x^\star)\right]. \tag{23}$$

Lemma 5. *Assume that $f(x)$ has Lipschitz continuous gradient and is strongly convex with convexity parameter $\mu_f > 0$. Moreover, assume $0 < \mu \le \mu_f$ and*

1. *$\{y^{(k)}\}_{k\ge0}$ is an arbitrary sequence in $\mathbb{R}^n$,*
2. *$\{M_k\}_{k\ge0}$ is a sequence such that $\phi\big(T_{M_k}(y^{(k)})\big) \le \psi_{M_k}\big(y^{(k)}; T_{M_k}(y^{(k)})\big)$,*
3. *$\{\alpha_k\}_{k\ge0}$ is a sequence that satisfies $\alpha_k \in (0,1)$ and $\sum_{k=0}^\infty \alpha_k = \infty$.*

Define the sequence $\{V_k(x)\}_{k\ge0}$ by letting $V_0(x)$ be an arbitrary function on $\mathbb{R}^n$ and for $k \ge 0$,

$$\begin{aligned} V_{k+1}(x) \;&=\; (1-\alpha_k)V_k(x) \\ &\quad + \alpha_k\left[\phi\big(T_{M_k}(y^{(k)})\big) + \left\langle g_{M_k}(y^{(k)}), x - y^{(k)}\right\rangle + \frac{1}{2M_k}\left\|g_{M_k}(y^{(k)})\right\|_2^2 + \frac{\mu}{2}\|x - y^{(k)}\|_2^2\right], \end{aligned} \tag{24}$$

and define the sequence $\{\tau_k\}_{k\ge0}$ by setting $\tau_0 = 1$ and

$$\tau_{k+1} = \tau_k(1 - \alpha_k), \qquad k \ge 0. \tag{25}$$

Then the pair $\{V_k(x)\}_{k\ge0}$ and $\{\tau_k\}_{k\ge0}$ is an estimate sequence of $\phi(x)$.

Proof. First we show that the inequality (21) holds for all $k \ge 0$. It holds for $k = 0$ since $\tau_0 = 1$. Suppose it holds for some $k \ge 0$. Then the assumption on $\{M_k\}_{k\ge0}$ and Lemma 3 imply

$$\begin{aligned} V_{k+1}(x) \;&\le\; (1-\alpha_k)V_k(x) + \alpha_k\phi(x) \\ &=\; (1 - (1-\alpha_k)\tau_k)\,\phi(x) + (1-\alpha_k)(V_k(x) - (1-\tau_k)\,\phi(x)) \\ &\le\; (1 - (1-\alpha_k)\tau_k)\,\phi(x) + (1-\alpha_k)\tau_k V_0(x) \\ &=\; (1-\tau_{k+1})\,\phi(x) + \tau_{k+1}V_0(x). \end{aligned}$$

In addition, we note that the sequence $\{\tau_k\}_{k\geq 0}$ defined by (25) is the same as the one given in (14), and the assumptions $\alpha_k \in (0,1)$ and $\sum_{k=0}^{\infty}\alpha_k = \infty$ ensures $\tau_k \to 0$. Therefore, by Definition 1, $\{V_k(x)\}_{k\geq 0}$ and $\{\tau_k\}_{k\geq 0}$ is an estimate sequence of $\phi(x)$. □

Lemma 6. *Let $V_0(x) = \phi(x^{(0)}) + \frac{\mu}{2}\|x - x^{(0)}\|_2^2$ where $x^{(0)}$ is an arbitrary point in $\mathbb{R}^n$. If we choose $\alpha_k = \sqrt{\frac{\mu}{M_k}}$ for $k \geq 0$, then the sequence $\{V_k(x)\}_{k\geq 0}$ defined by (24) can be written as*

$$V_k(x) = V_k^\star + \frac{\mu}{2}\|x - v^{(k)}\|_2^2, \tag{26}$$

where the sequences $\{v^{(k)}\}$ and $\{V_k^\star\}$ are defined as $v^{(0)} = x^{(0)}$, $V_0^\star = \phi(x^{(0)})$, and for $k \geq 0$,

$$\begin{aligned} v^{(k+1)} &= (1-\alpha_k)v^{(k)} + \alpha_k y^{(k)} - \frac{1}{\alpha_k M_k} g_{M_k}(y^{(k)}), & (27)\\ V_{k+1}^\star &= (1-\alpha_k)V_k^\star + \alpha_k \phi\big(T_{M_k}(y^{(k)})\big) - \frac{1-\alpha_k}{2M_k}\big\|g_{M_k}(y^{(k)})\big\|_2^2 & (28)\\ &\quad +\alpha_k(1-\alpha_k)\left(\frac{\mu}{2}\|y^{(k)} - v^{(k)}\|_2^2 + \left\langle g_{M_k}(y^{(k)}), v^{(k)} - y^{(k)}\right\rangle\right). \end{aligned}$$

Proof. Follow similar algebraic derivations as in (Nesterov, 2004, Lemma 2.2.3), omitted here. □

In order to prove Theorem 1, we first notice that the three sequences generated by the scAPG method (Algorithms 2 and 1), $\{y^{(k)}\}$, $\{M_k\}$ and $\{\alpha_k\}$, satisfy the assumptions in Lemma 5. More specifically, Lemma 5 does not have any restriction on $\{y^{(k)}\}$, the condition on $\{M_k\}$ is exactly the stopping criterion in Algorithm 2, and also

$$\alpha_k = \sqrt{\frac{\mu}{M_k}} \geq \sqrt{\frac{\mu}{\gamma_{\text{inc}} L_f}} \quad \Longrightarrow \quad \alpha_k \in (0,1), \quad \sum_{k=0}^{\infty}\alpha_k = \infty.$$

Therefore, we can use them to construct an estimate sequence as in (24) and (25). Next we need to show that the choice of $x^{(k+1)} = T_{M_k}(y^{(k)})$ guarantees the condition (22), so that we can invoke Lemma 4 to prove the convergence rate.

To proceed, we split the update of $y^{(k)}$, i.e.,

$$y^{(k)} = x^{(k)} + \frac{\alpha_k(1-\alpha_{k-1})}{\alpha_{k-1}(1+\alpha_k)}(x^{(k)} - x^{(k-1)}), \tag{29}$$

into the following two steps:

$$\begin{aligned} v^{(k)} &= x^{(k)} + \frac{1-\alpha_{k-1}}{\alpha_{k-1}}(x^{(k)} - x^{(k-1)}), & (30)\\ y^{(k)} &= \frac{\alpha_k v^{(k)} + x^{(k)}}{\alpha_k + 1}. & (31) \end{aligned}$$

It is straightforward to check that substituting the expression of $v^{(k)}$ in (30) into (31) yields (29). Also it is no coincidence that we used the same notation $v^{(k)}$ as the minimizer of $V_k(x)$: together with (31), the update of $v^{(k)}$ in (27) is equivalent to (30). To see this, we first check that with the choice of $\alpha_{-1} = 1$ and $x^{(-1)} = x^{(0)}$ in Algorithm 1, it holds that $y^{(0)} = v^{(0)} = x^{(0)}$. Then, with the choice of $x^{(k+1)} = T_{M_k}(y^{(k)})$ for $k \geq 0$, the expression of $v^{(k+1)}$ in (27) becomes

$$\begin{aligned} v^{(k+1)} &= (1-\alpha_k)v^{(k)} + \alpha_k y^{(k)} - \frac{1}{\alpha_k M_k} g_{M_k}(y^{(k)})\\ &= (1-\alpha_k)v^{(k)} + \alpha_k y^{(k)} - \frac{1}{\alpha_k M_k} M_k(y^{(k)} - x^{(k+1)})\\ &= (1-\alpha_k)v^{(k)} + \left(\frac{\alpha_k^2 - 1}{\alpha_k}\right) y^{(k)} + \frac{1}{\alpha_k} x^{(k+1)}. \end{aligned}$$

Now replacing $y^{(k)}$ in the above expression with the right-hand side of (31) yields

$$\begin{aligned} v^{(k+1)} &= (1-\alpha_k)v^{(k)} + \left(\frac{\alpha_k^2-1}{\alpha_k}\right)\frac{\alpha_k v^{(k)} + x^{(k)}}{\alpha_k+1} + \frac{1}{\alpha_k}x^{(k+1)} \\ &= x^{(k+1)} + \frac{1-\alpha_k}{\alpha_k}(x^{(k+1)} - x^{(k)}), \end{aligned}$$

which is the same as (30). Therefore, the sequence y_k generated in Algorithm 2 is a convex combination of the current iterate $x^{(k)}$ and $v^{(k)}$, which is the minimizer of the function $V_k(x)$,

Finally, we are ready to prove that (22) holds for all $k \geq 0$. It holds for $k = 0$ simply by the definition of $V_0^\star$. Given that it holds for some k, i.e., $V_k^\star \geq \phi(x^{(k)})$, the expression of $V_{k+1}^\star$ in (28) implies

$$\begin{aligned} V_{k+1}^\star &\geq (1-\alpha_k)\phi(x^{(k)}) + \alpha_k\phi(x^{(k+1)}) - \frac{1-\alpha_k}{2M_k}\|g_{M_k}(y^{(k)})\|_2^2 \\ &\quad + \alpha_k(1-\alpha_k)\left\langle g_{M_k}(y^{(k)}), v^{(k)} - y^{(k)}\right\rangle. \end{aligned} \tag{32}$$

According to Lemma 3, we have

$$\begin{aligned} \phi(x^{(k)}) &\geq \phi(x^{(k+1)}) + \left\langle g_{M_k}(y^{(k)}), x^{(k)} - y^{(k)}\right\rangle + \frac{1}{2M_k}\|g_{M_k}(y^{(k)})\|_2^2 + \frac{\mu}{2}\|x^{(k)} - y^{(k)}\|_2^2 \\ &\geq \phi(x^{(k+1)}) + \left\langle g_{M_k}(y^{(k)}), x^{(k)} - y^{(k)}\right\rangle + \frac{1}{2M_k}\|g_{M_k}(y^{(k)})\|_2^2. \end{aligned}$$

Applying this to $\phi(x^{(k)})$ in (32) yields

$$\begin{aligned} V_{k+1}^\star &\geq \phi(x^{(k+1)}) + (1-\alpha_k)\left\langle g_{M_k}(y^{(k)}), \alpha_k(v^{(k)} - y^{(k)}) + x^{(k)} - y^{(k)}\right\rangle \\ &= \phi(x^{(k+1)}) + (1-\alpha_k)\left\langle g_{M_k}(y^{(k)}), \left(\alpha_k v^{(k)} + x^{(k)}\right) - (\alpha_k+1)y^{(k)}\right\rangle \\ &= \phi(x^{(k+1)}), \end{aligned}$$

where the last equality is due to (31). We have shown that (22) holds for all $k \geq 0$. Therefore, the result (12) of Theorem 1 follows from Lemma 4 and the definition of $V_0(x)$.

It remains to prove (13). Using strong convexity of ϕ and (12), we have

$$\frac{\mu}{2}\|x^{(k)} - x^\star\|_2^2 \leq \phi(x^{(k)}) - \phi(x^\star) \leq \tau_k\left[\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\|x^{(0)} - x^\star\|_2^2\right] = \tau_k\left(V_0(x^\star) - \phi(x^\star)\right).$$

According to (26),

$$\frac{\mu}{2}\|v^{(k)} - x^\star\|_2^2 = V_k(x^\star) - V_k^\star.$$

Since the relationship $\phi(x^{(k)}) \leq V_k^\star$ implies $\phi(x^\star) \leq V_k^\star$, we have

$$\begin{aligned} \frac{\mu}{2}\|v^{(k)} - x^\star\|_2^2 &\leq V_k(x^\star) - \phi(x^\star) \\ &\leq (1-\tau_k)\phi(x^\star) + \tau_k V_0(x^\star) - \phi(x^\star) \\ &= \tau_k\left(V_0(x^\star) - \phi(x^\star)\right), \end{aligned}$$

where in the second inequality we used the fact that $\{V_k(x)\}$ and $\{\tau_k\}$ is an estimate sequence of $\phi(x)$. Finally, by convexity of the function $\frac{\mu}{2}\|\cdot - x^\star\|_2^2$ and (31),

$$\begin{aligned} \frac{\mu}{2}\|y^{(k)} - x^\star\|_2^2 &\leq \frac{\alpha_k}{\alpha_k+1}\cdot\frac{\mu}{2}\|v^{(k)} - x^\star\|_2^2 + \frac{1}{\alpha_k+1}\cdot\frac{\mu}{2}\|x^{(k)} - x^\star\|_2^2 \\ &\leq \frac{\alpha_k}{\alpha_k+1}\tau_k\left(V_0(x^\star) - \phi(x^\star)\right) + \frac{1}{\alpha_k+1}\tau_k\left(V_0(x^\star) - \phi(x^\star)\right) \\ &= \tau_k\left(V_0(x^\star) - \phi(x^\star)\right) \\ &= \tau_k\left[\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\|x^{(0)} - x^\star\|_2^2\right]. \end{aligned}$$

This finishes the proof of Theorem 1.

6.2. Proof of Lemma 1

We first need to prove the following lemma.

Lemma 7. *Suppose* $0 < \mu \le L_{\min}$. *Then Algorithm 1 guarantees that*

$$\phi(x^{(k+1)}) \le \phi(x^{(k)}) + \frac{M_{k-1}}{2}\|x^{(k)} - x^{(k-1)}\|_2^2 - \frac{M_k}{2}\|x^{(k+1)} - x^{(k)}\|_2^2. \tag{33}$$

Proof. According to the optimality of $x^{(k+1)} = T_{M_k}(y^{(k)})$ in minimizing the function $\psi(y^{(k)}, \cdot)$, there exists a $\xi \in \partial\Psi(x^{(k+1)})$ such that

$$\nabla f(y^{(k)}) + \xi + M_k(x^{(k+1)} - y^{(k)}) = 0. \tag{34}$$

Let $\beta_k = \frac{\alpha_k(1-\alpha_{k-1})}{\alpha_{k-1}(1+\alpha_k)}$. Using the assumed property of $f(x)$, we have

$$\begin{aligned}
\phi(x^{(k+1)}) &\le f(y^{(k)}) + \langle \nabla f(y^{(k)}), x^{(k+1)} - y^{(k)}\rangle + \frac{M_k}{2}\|x^{(k+1)} - y^{(k)}\|_2^2 + \Psi(x^{(k+1)}) \\
&= f(y^{(k)}) + \langle \nabla f(y^{(k)}), x^{(k+1)} - x^{(k)}\rangle \\
&\quad + \langle \nabla f(y^{(k)}), x^{(k)} - y^{(k)}\rangle + \frac{M_k}{2}\|x^{(k+1)} - y^{(k)}\|_2^2 + \Psi(x^{(k+1)}) \\
&= f(y^{(k)}) - \langle \xi + M_k(x^{(k+1)} - y^{(k)}), x^{(k+1)} - x^{(k)}\rangle \\
&\quad + \langle \nabla f(y^{(k)}), x^{(k)} - y^{(k)}\rangle + \frac{M_k}{2}\|x^{(k+1)} - y^{(k)}\|_2^2 + \Psi(x^{(k+1)}) \\
&= f(y^{(k)}) + \langle \nabla f(y^{(k)}), x^{(k)} - y^{(k)}\rangle + \Psi(x^{(k+1)}) + \langle \xi, x^{(k)} - x^{(k+1)}\rangle \\
&\quad + \frac{M_k}{2}\|x^{(k+1)} - y^{(k)}\|_2^2 - M_k\langle x^{(k+1)} - y^{(k)}, x^{(k+1)} - x^{(k)}\rangle \\
&\le f(x^{(k)}) - \frac{\mu_f}{2}\|x^{(k)} - y^{(k)}\|_2^2 + \Psi(x^{(k)}) \\
&\quad + \frac{M_k}{2}\|x^{(k+1)} - y^{(k)}\|_2^2 - M_k\langle x^{(k+1)} - y^{(k)}, x^{(k+1)} - x^{(k)}\rangle.
\end{aligned}$$

Here, the first inequality is due to the stopping condition for searching M_k in algorithm 1. The first and third equalities are just reorganizing terms while the second one is due to (34). The last inequality are guaranteed by the strong convexity of $f(x)$ and the convexity of $\Psi(x)$. Given that $y^{(k)} = x^{(k)} + \beta_k(x^{(k)} - x^{(k-1)})$, the inequality above implies

$$\begin{aligned}
\phi(x^{(k+1)}) &\le \phi(x^{(k)}) - \frac{\mu_f\beta_k^2}{2}\|x^{(k)} - x^{(k-1)}\|_2^2 + \frac{M_k}{2}\|x^{(k+1)} - x^{(k)} - \beta_k(x^{(k)} - x^{(k-1)})\|_2^2 \\
&\quad - M_k\langle x^{(k+1)} - x^{(k)} - \beta_k(x^{(k)} - x^{(k-1)}), x^{(k+1)} - x^{(k)}\rangle \\
&= \phi(x^{(k)}) + \frac{(M_k - \mu_f)\beta_k^2}{2}\|x^{(k)} - x^{(k-1)}\|_2^2 - \frac{M_k}{2}\|x^{(k+1)} - x^{(k)}\|_2^2.
\end{aligned}$$

Using the fact $\alpha_k^2 M_k = \mu$, we can show that

$$\begin{aligned}
(M_k - \mu_f)\,\beta_k^2 &= (M_k - \mu_f)\,\frac{(1-\alpha_{k-1})^2\alpha_k^2}{(1+\alpha_k)^2\alpha_{k-1}^2} = (M_k - \mu_f)\,\frac{(1-\alpha_{k-1})^2 M_{k-1}}{(1+\alpha_k)^2 M_k} \\
&= \left(1 - \frac{\mu_f}{M_k}\right)\frac{(1-\alpha_{k-1})^2}{(1+\alpha_k)^2}M_{k-1} \le M_{k-1},
\end{aligned}$$

which implies our conclusion. □

Then, Lemma 1 can be easily proved by applying inequality (33) recursively to obtain

$$\begin{aligned}
\phi(x^{(k+1)}) &\le \phi(x^{(0)}) + \frac{M_{-1}}{2}\|x^{(0)} - x^{(-1)}\|_2^2 - \frac{M_k}{2}\|x^{(k+1)} - x^{(k)}\|_2^2 \\
&= \phi_\lambda(x^{(0)}) - \frac{M_k}{2}\|x^{(k+1)} - x^{(k)}\|_2^2.
\end{aligned}$$

Here the last equality holds because $x^{(0)} = x^{(-1)}$.

6.3. Proof of Lemma 2

We will need the following properties of composite gradient mapping shown in (Nesterov, 2007):

Lemma 8. *(Part of (Nesterov, 2007, Theorem 2)) For any $y \in \operatorname{dom}\Psi$ and any $L > 0$,*

$$\psi_L(y; T_L(y)) \le \phi(y) - \frac{1}{2L}\|g_L(y)\|_2^2.$$

Lemma 9. *(Part of (Nesterov, 2007, Theorem 1)) For any $x, y \in \operatorname{dom}\Psi$ and any $L > 0$, we have*

$$\langle \phi'(T_L(y)), x - T_L(y)\rangle \ge -\left(1 + \frac{1}{L}S_L(y)\right)\cdot \|g_L(y)\|_2 \cdot \|T_L(y) - x\|_2.$$

Lemma 10. *(Nesterov, 2007, Lemma 2) Suppose ϕ is strongly convex with convexity parameter $\mu > 0$, and let $x^\star$ be the unique minimizer of ϕ. Then for any $y \in \operatorname{dom}\Psi$ and any $L > 0$, we have*

$$\|T_L(y) - x^\star\|_2 \le \frac{1}{\mu}\left(1 + \frac{1}{L}S_L(y)\right)\|g_L(y)\|_2.$$

By definition of the gradient mapping,

$$\left\|g_{M_k}(y^{(k)})\right\|_2 = \left\|M_k\big(y^{(k)} - x^{(k+1)}\big)\right\|_2 \le M_k\Big(\left\|y^{(k)} - x^\star\right\|_2 + \left\|x^{(k+1)} - x^\star\right\|_2\Big),$$

where $x^\star$ is the unique minimizer of ϕ. By strong convexity of ϕ, we have

$$\frac{\mu}{2}\left\|x^{(k+1)} - x^\star\right\|_2 \le \phi(x^{(k+1)}) - \phi(x^\star).$$

Then using Theorem 1, we obtain

$$\begin{aligned}\|g_{M_k}(y^{(k)})\|_2 &\le M_k\left(\sqrt{2\tau_k} + \sqrt{2\tau_{k+1}}\right)\sqrt{\frac{1}{\mu}\left(\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\|x^{(0)} - x^\star\|_2^2\right)}\\ &\le 2M_k\sqrt{2\tau_k}\sqrt{\frac{1}{\mu}\left(\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\|x^{(0)} - x^\star\|_2^2\right)}. \end{aligned} \tag{35}$$

On the other hand, also by strong convexity of ϕ, we have

$$\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\left\|x^\star - x^{(0)}\right\|_2^2 \le -\left\langle \phi'(x^{(0)}), x^\star - x^{(0)}\right\rangle,$$

where $\phi'(x^{(0)})$ is a subgradient of ϕ at $x^{(0)}$. According to the updating schemes within the subroutine $\{x^{(0)}, M_{-1}, \alpha_{-1}, g^{(-1)}, S_{-1}\} \leftarrow$ `AccelLineSearch`$(x^{\text{ini}}, x^{\text{ini}}, L_{\text{ini}}, \mu, 1)$, we have $x^{(0)} = T_{M_{-1}}(x^{\text{ini}})$, $S_{-1} = S_{M_{-1}}(x^{\text{ini}})$ and $g^{(-1)} = g_{M_{-1}}(x^{\text{ini}})$. According to Lemma 9, we have

$$\left\langle \phi'(x^{(0)}), x^\star - x^{(0)}\right\rangle \ge -\left(1 + \frac{S_{-1}}{M_{-1}}\right)\left\|g^{(-1)}\right\|_2 \cdot \left\|x^{(0)} - x^\star\right\|_2.$$

Therefore,

$$\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\left\|x^{(0)} - x^\star\right\|_2^2 \le \left(1 + \frac{S_{-1}}{M_{-1}}\right)\left\|g^{(-1)}\right\|_2 \cdot \left\|x^{(0)} - x^\star\right\|_2.$$

Moreover, by Lemma 10,

$$\left\|x^{(0)} - x^\star\right\|_2 \le \frac{1}{\mu}\left(1 + \frac{S_{-1}}{M_{-1}}\right)\|g^{(-1)}\|_2.$$

The above two inequalities imply

$$\phi(x^{(0)}) - \phi(x^\star) + \frac{\mu}{2}\left\|x^{(0)} - x^\star\right\|_2^2 \le \frac{1}{\mu}\left(1 + \frac{S_{-1}}{M_{-1}}\right)^2\left\|g^{(-1)}\right\|_2^2.$$

Combining this with the inequality (35) gives the desired result.

6.4. Proof of Theorem 2

To prove Theorem 2, we need the following lemma which shows that we can measure how close $T_L(y)$ is from satisfying the optimality condition by using the norm of the composite gradient mapping at y.

Lemma 11. *(Xiao & Zhang, 2013, Lemma 2) If f has Lipschitz continuous gradients with Lipschitz constant L_f, then*

$$\omega(T_L(y)) \le \left(1 + \frac{S_L(y)}{L}\right) \|g_L(y)\|_2 \le \left(1 + \frac{L_f}{L}\right) \|g_L(y)\|_2.$$

By Lemma 11 and the facts that $M_k \ge L_{\min}$ and $S_{M_k}(y^{(k)}) \le L_f$, we have

$$\omega(x^{(k+1)}) \;\le\; \left(1 + \frac{L_f}{L_{\min}}\right) \left\|g_{M_k}(y^{(k)})\right\|_2.$$

According to the stopping criterion $\omega(x^{(k+1)}) \le \hat{\epsilon}$, the algorithm stops after the condition A is satisfied N_A times if

$$\left(1 + \frac{L_f}{L_{\min}}\right) \left\|g_{M_k}(y^{(k)})\right\|_2 \le \left(1 + \frac{L_f}{L_{\min}}\right) \theta_{sc}^{N_{\rm A}} \|g^{\rm ini}\|_2 \le \hat{\epsilon}.$$

Therefore, N_A is at most $\left\lceil \log_{1/\theta_{sc}}\left(\left(1 + \frac{L_f}{L_{\min}}\right) \frac{\|g^{\rm ini}\|_2}{\hat{\epsilon}}\right)\right\rceil$.

Note that condition B can be satisfied only when $\mu > \mu_f$. Once $\mu = \mu_0/\gamma_{sc}^{N_{\rm B}} \le \mu_f$, it will no longer be satisfied. Therefore, $N_{\rm B}$ is at most $\left\lceil \log_{\gamma_{sc}}\left(\frac{\mu_0}{\mu_f}\right)\right\rceil$ and we always have $\mu \ge \mu_f/\gamma_{\rm sc}$.

Next we bound the number of iterations before either condition A or B must be satisfied. It suffices to find the bound for condition B. For this purpose, we first upper bound the squared left-hand side of condition B:

$$\begin{aligned} 8\tau_k \left(\frac{M_k}{\mu}\right)^2 \left(1 + \frac{S_{-1}}{M_{-1}}\right)^2 &\le 8\left(1 - \sqrt{\frac{\mu}{L_f\gamma_{\rm inc}}}\right)^k \left(\frac{L_f\gamma_{\rm inc}}{\mu}\right)^2 \left(1 + \frac{L_f}{L_{\min}}\right)^2 \\ &\le 8\left(1 - \sqrt{\frac{\mu_f/\gamma_{\rm sc}}{L_f\gamma_{\rm inc}}}\right)^k \left(\frac{L_f\gamma_{\rm inc}}{\mu_f/\gamma_{\rm sc}}\right)^2 \left(1 + \frac{L_f}{L_{\min}}\right)^2. \end{aligned}$$

Setting the above upper bound be less than $\theta_{\rm sc}^2$, we find that either condition A or B must be satisfied after the following number of iterations:

$$\begin{aligned} &\ln\left(8\left(\frac{L_f\gamma_{\rm inc}\gamma_{\rm sc}}{\mu_f\theta_{sc}}\right)^2 \left(1 + \frac{L_f}{L_{\min}}\right)^2\right) \Big/ \ln\left(1\Big/\left(1 - \sqrt{\frac{\mu_f}{L_f\gamma_{\rm inc}\gamma_{\rm sc}}}\right)\right) \\ \le\; &\sqrt{\frac{L_f\gamma_{\rm inc}\gamma_{sc}}{\mu_f}} \ln\left(8\left(\frac{L_f\gamma_{\rm inc}\gamma_{\rm sc}}{\mu_f\theta_{sc}}\right)^2 \left(1 + \frac{L_f}{L_{\min}}\right)^2\right). \end{aligned}$$

Hence, the total number iterations of Algorithm 3 is bounded by the above upper bound multiplied by $(N_{\rm A} + N_{\rm B})$.

6.5. Proof of Theorem 3

Since the ℓ_1-LS problem (5) depends on the parameter λ, some of the notations we introduced before can be further parametrized by λ. More specifically, we define

$$\begin{aligned} \psi_{\lambda,L}(y;x) &= f(y) + \nabla f(y)^T(x-y) + \frac{L}{2}\|x-y\|_2^2 + \lambda\|x\|_1 \\ T_{\lambda,L}(y) &= \arg\min_x \; \psi_{\lambda,L}(y;x) \\ g_{\lambda,L}(y) &= L\big(y - T_{\lambda,L}(y)\big) \\ \omega_\lambda(x) &= \min_{\xi\in\partial\|x\|_1} \|\nabla f(x) + \lambda\xi\|_\infty \\ S_{\lambda,L}(y) &= \frac{\|\nabla f(T_{\lambda,L}(y)) - \nabla f(y)\|_2}{\|T_{\lambda,L}(y) - y\|_2}. \end{aligned}$$

Similarly, we use $\texttt{AdapAPG}(x^{\text{ini}}, L_{\text{ini}}, \mu_0, \hat{\epsilon}, \lambda)$ to represent applying Algorithm 3 to (5) whose regularization parameter is λ. Given the gradient $\nabla f(x)$, the optimality residue $\omega_\lambda(x)$ can be easily computed with $O(n)$ flops. For the ℓ_1-LS problem, the proximal gradient step, $T_{\lambda,L}(x)$, has the closed-form solution given as

$$T_{\lambda,L}(x) = \text{shrink}\left(x - \frac{1}{L}\nabla f(x),\ \frac{\lambda}{L}\right), \tag{36}$$

where $\text{shrink} : \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}^n$ is the well-known *shrinkage* or *soft-thresholding* operator, defined as

$$(\text{shrink}(x,\alpha))_i = \text{sgn}(x_i)\max\{|x_i| - \alpha,\ 0\}, \quad i = 1,\dots,n. \tag{37}$$

One of the key steps to prove Theorem 3 is showing the path of solutions in Algorithm 3 will be sparse if Assumption 1 and (19) holds. As a preparation for showing this property, in the next subsection, we present some technical lemmas regarding the sparsity of $T_{\lambda,L}(y)$, given that y is sparse and close enough to the optimality.

6.5.1. Sparsity along the solution path

First, we list some useful inequalities that are direct consequences of (17) and $\delta' \in (0, 0.2]$:

$$(1-\delta')\lambda - 4\|A^T z\|_\infty > 0 \tag{38}$$

$$(1+\delta')\lambda + \|A^T z\|_\infty \le 1.4\lambda \tag{39}$$

$$\lambda + \|A^T z\|_\infty \le (1.4-\delta')\lambda \tag{40}$$

$$\frac{(1+\delta')\lambda + \|A^T z\|_\infty}{(1-\delta')\lambda - \|A^T z\|_\infty} \le \gamma. \tag{41}$$

The following result means that if x is sparse, and it satisfies an approximate optimality condition for minimizing ϕ_λ, then $\phi_\lambda(x)$ is not much larger than $\phi_\lambda(\bar{x})$.

Lemma 12 (Lemma 4 in (Xiao & Zhang, 2013))**.** *Suppose Assumption 1 holds for some $\bar{x}$, δ', γ and $\tilde{s}$, and $\lambda \ge \lambda_{\text{tgt}}$. If x is sparse, i.e., $\|x_{\bar{S}^c}\|_0 \le \tilde{s}$, and it satisfies the approximate optimality condition*

$$\min_{\xi \in \partial\|x\|_1} \left\|A^T(Ax-b) + \lambda\xi\right\|_\infty \le \delta'\lambda, \tag{42}$$

then we have

$$\|(x-\bar{x})_{\bar{S}^c}\|_1 \le \gamma\|(x-\bar{x})_{\bar{S}}\|_1 \tag{43}$$

and

$$\|x-\bar{x}\|_2 \le \frac{1.4\,\lambda\sqrt{\bar{s}}}{\rho_-(A, \bar{s}+\tilde{s})} \tag{44}$$

and

$$\phi_\lambda(x) \le \phi_\lambda(\bar{x}) + \frac{1.4\,\delta'(1+\gamma)\lambda^2\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})}. \tag{45}$$

The next lemma means that if x is sparse, and $\phi_\lambda(x)$ is not much larger than $\phi_\lambda(\bar{x})$, then both $\|x-\bar{x}\|_2$ and $\|x-\bar{x}\|_1$ are small.

Lemma 13 (Lemma 5 in (Xiao & Zhang, 2013))**.** *Suppose Assumption 1 holds for some $\bar{x}$, δ', γ and $\tilde{s}$, and $\lambda \ge \lambda_{\text{tgt}}$. Consider x such that*

$$\|x_{\bar{S}^c}\|_0 \le \tilde{s}, \qquad \phi_\lambda(x) \le \phi_\lambda(\bar{x}) + \frac{1.4\,\delta'(1+\gamma)\lambda^2\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})},$$

then

$$\max\left\{\frac{1}{2.8\,\lambda}\|A(x-\bar{x})\|_2^2,\ \|x-\bar{x}\|_1\right\} \le \frac{1.4\,(1+\gamma)\lambda\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})}.$$

The next lemma implies that if both $x^{(k)}$ and $x^{(k-1)}$ are sparse and their objective values are not much larger than $\phi_\lambda(\bar{x})$, then the next iterate $x^{(k+1)}$ generated by the accelerated line search procedure (Algorithm 2) is also sparse. Its proof uses similar arguments as in (Xiao & Zhang, 2013, Lemma 6).

Lemma 14. *Suppose Assumption 1 holds for some $\bar{x}$, δ', γ and $\tilde{s}$, and $\lambda \geq \lambda_{\rm tgt}$. Suppose x and x' satisfies*

$$\begin{aligned} \|x_{\bar{S}^c}\|_0 \leq \tilde{s}, \qquad & \phi_\lambda(x) \leq \phi_\lambda(\bar{x}) + \tfrac{2\delta'(1+\gamma)\lambda^2\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})}, \\ \|x'_{\bar{S}^c}\|_0 \leq \tilde{s}, \qquad & \phi_\lambda(x') \leq \phi_\lambda(\bar{x}) + \tfrac{2\delta'(1+\gamma)\lambda^2\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})}, \end{aligned} \tag{46}$$

and $y = x + \beta(x - x')$ with $0 \leq \beta \leq 1$. Then for any $L < \gamma_{\rm inc}\rho_+(A, \bar{s} + 3\tilde{s})$, we have

$$\left\| \left(T_{\lambda,L}(y)\right)_{\bar{S}^c} \right\|_0 < \tilde{s}.$$

Proof. Recall that $T_{\lambda,L}$ can be computed by the soft-thresholding operator as in (36). That is,

$$(T_L(y))_i = \mathrm{sgn}(\tilde{y}_i) \max\left\{ |\tilde{y}_i| - \frac{\lambda}{L},\ 0 \right\}, \qquad i = 1, \ldots, n,$$

where

$$\tilde{y} = y - \frac{1}{L}A^T(Ay - b) = y - \frac{1}{L}A^TA(y - \bar{x}) + \frac{1}{L}A^Tz.$$

In order to upper bound the number of nonzero elements in $(T_L(y))_{\bar{S}^c}$, we split the truncation threshold λ/L on elements of $\tilde{y}_{\bar{S}^c}$ into three parts:

- $0.175\,\lambda/L$ on elements of $y_{\bar{S}^c}$,
- $0.125\,\lambda/L$ on elements of $(1/L)A^Tz$, and
- $0.7\,\lambda/L$ on elements of $(1/L)A^TA(y - \bar{x})$.

Since by assumption $\|A^Tz\|_\infty \leq \lambda/8$, we have $\left|\{j : ((1/L)A^Tz)_j > 0.125\,\lambda/L\}\right| = 0$. Therefore,

$$\left\| \left(T_{\lambda,L}(y)\right)_{\bar{S}^c} \right\|_0 \ \leq\ \left|\{j \in \bar{S}^c : |y_j| > 0.175\,\lambda/L\}\right| + \left|\{j : \left|\left(A^TA(y - \bar{x})\right)_j\right| \geq 0.7\,\lambda\}\right|.$$

Note that

$$\begin{aligned} \left|\{j \in \bar{S}^c : |y_j| \geq 0.175\,\lambda/L\}\right| &= \left|\{j \in \bar{S}^c : |(y - \bar{x})_j| \geq 0.175\,\lambda/L\}\right| \\ &\leq \left|\{j : |(y - \bar{x})_j| \geq 0.175\,\lambda/L\}\right| \\ &\leq L(0.175\,\lambda)^{-1}\|y - \bar{x}\|_1 \\ &\leq L(0.175\,\lambda)^{-1}((1+\beta)\|x - \bar{x}\|_1 + \beta\|x' - \bar{x}\|_1) \\ &\leq \frac{1.4\,L(1+2\beta)(1+\gamma)\lambda\bar{s}}{0.175\,\lambda\rho_-(A, \bar{s}+\tilde{s})} \qquad (47) \\ &\leq \frac{24\,L(1+\gamma)\bar{s}}{\rho_-(A, \bar{s}+\tilde{s})}, \qquad (48) \end{aligned}$$

where the second-to-the-last inequality follows from Lemma 13, and the last one used $\beta \in [0, 1]$.

For the last part, consider S with maximum size $s = |S| \leq \tilde{s}$ such that

$$S \subset \{j : |(A^TA(y - \bar{x}))_j| \geq 0.7\,\lambda\}.$$

Then there exists u such that $\|u\|_\infty = 1$ and $\|u\|_0 = s$, and $0.7\,s\lambda \leq u^TA^TA(y - \bar{x})$. Moreover,

$$0.7\,s\lambda \ \leq\ u^TA^TA(y - \bar{x}) \ \leq\ \|Au\|_2\|A(y - \bar{x})\|_2 \ \leq\ \sqrt{\rho_+(A, s)}\sqrt{s}(1+2\beta)\sqrt{\frac{2 \cdot 1.4^2\,(1+\gamma)\lambda^2\bar{s}}{\rho_-(A, \bar{s}+\tilde{s})}},$$

where the last inequality again follows from Lemma 13. Taking squares of both sides of the above inequality gives

$$s \le \frac{8\,\rho_+(A,s)(1+\gamma)\bar{s}(1+2\beta)^2}{\rho_-(A,\bar{s}+\tilde{s})} \le \frac{72\,\rho_+(A,\tilde{s})(1+\gamma)\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})} < \tilde{s},$$

where the last inequality is due to (18). Since $s = |S|$ achieves the maximum possible value such that $s \le \tilde{s}$ for any subset S of $\{j : |(A^TA(y-\bar{x}))_j| \ge 0.7\,\lambda\}$, and the above inequality shows that $s < \tilde{s}$, we must have

$$S = \{j : |(A^TA(y-\bar{x}))_j| \ge 0.7\,\lambda\},$$

and thus

$$s = \big|\{j : |(A^TA(y-\bar{x}))_j| \ge 0.7\,\lambda\}\big| \le \left\lfloor \frac{72\,\rho_+(A,\tilde{s})(1+\gamma)\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})} \right\rfloor.$$

Finally, combining the above bound with the bound in (48) gives

$$\big\|\big(T_{\lambda,L}(x)\big)_{\bar{S}^c}\big\|_0 \;\le\; \frac{24\,(L+3\rho_+(A,\tilde{s}))}{\rho_-(A,\bar{s}+\tilde{s})}(1+\gamma)\bar{s}.$$

Under the assumption $L < \gamma_{\rm inc}\rho_+(A,\bar{s}+3\tilde{s})$ and (18), the right-hand side of the above inequality is less than $\tilde{s}$. This proves the desired result. □

Finally, we are ready to prove Theorem 3 in the next subsection.

6.5.2. Proof of Theorem 3

It can be shown that (Nesterov, 2007), the PG method keeps the value of objective function decreasing monotonically. This is the key property for the PGH method in (Xiao & Zhang, 2013) to enforce all the iterates along the solution path to be sufficiently sparse. Unfortunately, the scAPG and AdapAPG methods do not have such a monotone decreasing property. As an alternative, we proved that they have a non-blowout property (Lemma 1); that is, the objective value at any intermediate step will not exceed the initial objective value. This is the key in showing that all the iterates along the solution path are sufficiently sparse for the AdapAPG method, provided that the initial point is sparse and not far from optimality.

Lemma 15. *Suppose Assumption 1 holds for some $\bar{x}$, δ', γ and $\tilde{s}$. In addition, assume $\lambda \ge \lambda_{\rm tgt}$ and $\mu \le L_{\min}$. If the initial point $x^{\rm ini}$ in Algorithm 3 satisfies*

$$\big\|x^{\rm ini}_{\bar{S}^c}\big\|_0 \le \tilde{s}, \qquad \omega_\lambda(x^{\rm ini}) \le \delta'\lambda,$$

then for all $k \ge 0$, we have

$$\big\|x^{(k)}_{\bar{S}^c}\big\|_0 \le \tilde{s}, \qquad \big\|y^{(k)}_{\bar{S}^c}\big\|_0 \le 2\tilde{s}.$$

Proof. According to Lemma 12, the assumptions on $x^{\rm ini}$ implies

$$\phi_\lambda(x^{\rm ini}) \le \phi_\lambda(\bar{x}) + \frac{1.4\,\delta'(1+\gamma)\lambda^2\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})}.$$

Because $x^{(0)} = T_{\lambda,M}(x^{\rm ini})$, we have $\phi_\lambda(x^{(0)}) \le \phi_\lambda(x^{\rm ini})$ so that

$$\phi_\lambda(x^{(0)}) \le \phi_\lambda(\bar{x}) + \frac{1.4\,\delta'(1+\gamma)\lambda^2\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})}.$$

Although Algorithm 3 is not monotone decreasing, the non-blowout property in Lemma 1 guarantees that, for all $k \ge 0$,

$$\phi_\lambda(x^{(k+1)}) \le \phi_\lambda(\bar{x}) + \frac{1.4\,\delta'(1+\gamma)\lambda^2\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})}.$$

Because $\big\|x^{\rm ini}_{\bar{S}^c}\big\|_0 \le \tilde{s}$, we have $\big\|x^{(-1)}_{\bar{S}^c}\big\|_0 = \big\|x^{(0)}_{\bar{S}^c}\big\|_0 \le \tilde{s}$ according to Lemma 14. Suppose $\big\|x^{(k)}_{\bar{S}^c}\big\|_0 \le \tilde{s}$ and $\big\|x^{(k-1)}_{\bar{S}^c}\big\|_0 \le \tilde{s}$. Since $y^{(k+1)} = x^{(k)} + \beta_k(x^{(k)} - x^{(k-1)})$ and $x^{(k+1)} = T_{M_k}(y^{(k)})$, Lemma 14 again implies $\big\|x^{(k+1)}_{\bar{S}^c}\big\|_0 \le \tilde{s}$. By induction, we have $\big\|x^{(k)}_{\bar{S}^c}\big\|_0 \le \tilde{s}$ holds for all k, which further implies $\big\|y^{(k)}_{\bar{S}^c}\big\|_0 \le 2\tilde{s}$ for all k. □

According to Lemma 12, under the condition (19), Algorithm 3 essentially operates only on vectors with at most either $\tilde{s}$ or $2\tilde{s}$ nonzero components. Therefore, we are solving the ℓ_1-LS problem restricted in a sparse subspace, where the restricted smoothness and restricted strong convexity are available, that is,

$$f(x) \;\geq\; f(y) + \langle \nabla f(y), x-y\rangle + \frac{\rho_-(A, \bar{s}+3\tilde{s})}{2}\|x-y\|_2^2,$$

$$f(x) \;\leq\; f(y) + \langle \nabla f(y), x-y\rangle + \frac{\rho_+(A, \bar{s}+3\tilde{s})}{2}\|x-y\|_2^2.$$

Here, the effective sparse level is $s' = \bar{s}+3\tilde{s}$ because when the above two inequalities are used in Section 2 and Section 3, they are always applied to x and y with $\left\|x_{\bar{S}^c}\right\|_0 \leq \tilde{s}$ and $\left\|y_{\bar{S}^c}\right\|_0 \leq 2\tilde{s}$. To show Theorem 3, we just need to repeat the proof of Theorem 2 by replacing L_f and μ_f with $\rho_+(A, \bar{s}+3\tilde{s})$ and $\rho_-(A, \bar{s}+3\tilde{s})$, respectively.

6.6. Proof of Theorem 4

In Algorithm 4, $\hat{x}^{(K)}$ denotes an approximate solution for minimizing the function ϕ_{λ_K}. A key idea of the APG homotopy method is to use $\hat{x}^{(K)}$ as the starting point in the AdapAPG method for minimizing the next function $\phi_{\lambda_{K+1}}$. The following lemma shows that if we choose the parameters δ and η appropriately, then $\hat{x}^{(K)}$ satisfies the approximate optimality condition for λ_{K+1} that guarantees local geometric convergence.

Lemma 16 (Lemma 7 in (Xiao & Zhang, 2013)). *Suppose $\hat{x}^{(K)}$ satisfies the approximate optimality condition*

$$\omega_{\lambda_K}(\hat{x}^{(K)}) \leq \delta\lambda_K$$

for some $\delta < \delta'$. Let $\lambda_{K+1} = \eta\lambda_K$ for some η that satisfies

$$\frac{1+\delta}{1+\delta'} \leq \eta < 1. \tag{49}$$

Then we have

$$\omega_{\lambda_{K+1}}(\hat{x}^{(K)}) \leq \delta'\lambda_{K+1}.$$

Lemma 17 (Lemma 8 in (Xiao & Zhang, 2013)). *Suppose Assumption 1 holds for some $\bar{x}$, δ', γ and $\tilde{s}$, and $\lambda \geq \lambda_{\rm tgt}$. If x satisfies*

$$\omega_\lambda(x) \leq \delta'\lambda,$$

then for all $\lambda' \in [\lambda_{\rm tgt}, \lambda]$, we have

$$\phi_{\lambda'}(x) - \phi_{\lambda'}(x^\star(\lambda')) \leq \frac{2(1+\gamma)(\lambda+\lambda')(\omega_\lambda(x)+\lambda-\lambda')\bar{s}}{\rho_-(A, \bar{s}+\tilde{s})}.$$

Now we are ready to give an estimate of the overall complexity of the APG homotopy method (Algorithm 4). First, we need to bound the number of iterations within each call of Algorithm 3. According to Theorem 3 and Theorem 2, the total number of iterations in each call of $\texttt{AdapAPG}\big(\hat{x}^{(K)}, \hat{M}_K, \hat{\mu}_K, \hat{\epsilon}_{K+1}, \lambda_{K+1}\big)$ is no more than

$$(N_A+N_B)\sqrt{\kappa_{s'}\gamma_{\rm inc}\gamma_{sc}}\;\ln\left(8\left(\frac{\kappa_{s'}\gamma_{\rm inc}\gamma_{sc}}{\theta_{sc}}\right)^2\left(1+\frac{\rho_+(s')}{L_{\min}}\right)^2\right), \tag{50}$$

where N_A is the number of times that condition A is satisfied first, which is bounded as

$$N_A \leq \left\lceil \log_{\frac{1}{\theta_{sc}}}\left(\left(1+\frac{\rho_+(s')}{L_{\min}}\right)\frac{\|g_{\lambda_{K+1},M}(\hat{x}^{(K)})\|_2}{\hat{\epsilon}}\right)\right\rceil$$

with M generated from $\{x^{(0)}, M, \alpha_{-1}, g^{(-1)}, S_{-1}\} \leftarrow \texttt{AceelLineSearch}(\hat{x}^{(K)}, \hat{x}^{(K)}, \hat{M}_K, \hat{\mu}_K, 1)$, and N_B is the number of times that condition B is satisfied first, which is bounded as

$$N_B \;\leq\; \left\lceil \log_{\gamma_{sc}}\left(\frac{\hat{\mu}_K}{\rho_-(s')}\right)\right\rceil \;\leq\; \left\lceil \log_{\gamma_{sc}}\left(\frac{\hat{\mu}_0}{\rho_-(s')}\right)\right\rceil.$$

The bound on N_A depends on $\|g_{\lambda_{K+1},M}(\hat{x}^{(K)})\|_2$, which we can further bound using Lemma 8 to obtain

$$\begin{aligned}\left\|g_{\lambda_{K+1},M}(\hat{x}^{(K)})\right\|_2^2 &\leq 2M\left(\phi_{\lambda_{K+1}}(\hat{x}^{(K)})-\phi^\star_{\lambda_{K+1}}\right)\\ &\leq 2\gamma_{\rm inc}\rho_+(s')\left(\phi_{\lambda_{K+1}}(\hat{x}^{(K)})-\phi^\star_{\lambda_{K+1}}\right),\end{aligned}$$

where $\phi^\star_{\lambda_{K+1}}=\min_x \phi_{\lambda_{K+1}}(x)$. We still need to bound the gap $\phi_{\lambda_{K+1}}(\hat{x}^{(K)})-\phi^\star_{\lambda_{K+1}}$. Since Lemma 16 implies that $\omega_{\lambda_{K+1}}(\hat{x}^{(K)})\leq\delta'\lambda_{K+1}$, we can obtain directly from Lemma 17 the following inequality by setting $\lambda'=\lambda=\lambda_{K+1}$ and $x=\hat{x}^{(K)}$:

$$\phi_{\lambda_{K+1}}(\hat{x}^{(K)})-\phi^\star_{\lambda_{K+1}} \;\leq\; \frac{4(1+\gamma)\lambda_{K+1}^2\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})}\leq\frac{4(1+\gamma)\lambda_{K+1}^2\bar{s}}{\rho_-(s')}.$$

Therefore, the bound on N_A can be relaxed as

$$\begin{aligned}N_A &\leq \left\lceil \log_{\frac{1}{\theta_{sc}}}\left(\left(1+\frac{\rho_+(s')}{L_{\min}}\right)\frac{\|g_{\lambda_{K+1},M}(\hat{x}^{(K)})\|_2}{\delta\lambda_{K+1}}\right)\right\rceil\\ &\leq \left\lceil \log_{\frac{1}{\theta_{sc}}}\left(\left(1+\frac{\rho_+(s')}{L_{\min}}\right)\frac{\sqrt{2\gamma_{\rm inc}\rho_+(s')\left(\phi_{\lambda_{K+1}}(\hat{x}^{(K)})-\phi^\star_{\lambda_{K+1}}\right)}}{\delta\lambda_{K+1}}\right)\right\rceil\\ &\leq \left\lceil \log_{\frac{1}{\theta_{sc}}}\left(\left(1+\frac{\rho_+(s')}{L_{\min}}\right)\frac{\sqrt{8\gamma_{\rm inc}\rho_+(s')(1+\gamma)\lambda_{K+1}^2\bar{s}}}{\delta\lambda_{K+1}\sqrt{\rho_-(s')}}\right)\right\rceil\\ &= \left\lceil \log_{\frac{1}{\theta_{sc}}}\left(\left(1+\frac{\rho_+(s')}{L_{\min}}\right)\frac{\sqrt{8\gamma_{\rm inc}\kappa_{s'}(1+\gamma)\bar{s}}}{\delta}\right)\right\rceil.\end{aligned}$$

Combining the above bounds on N_A and N_B with (50) yields Part 1 of Theorem 4. We note that this bound is independent of λ_{K+1}.

In the homotopy method (Algorithm 4), after K outer iterations for $K\leq N-1$, we have from Lemma 16 that $\omega_{\lambda_{K+1}}(\hat{x}^{(K)})\leq\delta'\lambda_{K+1}$. The sparse recovery performance bound

$$\|\hat{x}^{(K)}-\bar{x}\|_2\leq 2\eta^{K+1}\lambda_0\sqrt{\bar{s}}/\rho_-(A,\bar{s}+\tilde{s})$$

follows directly from Lemma 12 and $\lambda_{K+1}=\eta^{K+1}\lambda_0$. Moreover, from Lemma 17 with $\lambda'=\lambda_{\rm tgt}$, $\lambda=\lambda_{K+1}$, and $x=\hat{x}^{(K)}$, we obtain

$$\phi_{\lambda_{\rm tgt}}(\hat{x}^{(K)})-\phi^\star_{\lambda_{\rm tgt}}\leq\frac{4.5(1+\gamma)\lambda_{K+1}^2\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})}=\eta^{2(K+1)}\frac{4.5(1+\gamma)\lambda_0^2\bar{s}}{\rho_-(A,\bar{s}+\tilde{s})}.$$

This proves Part 2 of Theorem 4.

In Algorithm 4, the number of homotopy stages, excluding the last one for $\lambda_{\rm tgt}$, is

$$N=\left\lfloor\frac{\ln(\lambda_0/\lambda_{\rm tgt})}{\ln(1/\eta)}\right\rfloor.$$

The last iteration for $\lambda_{\rm tgt}$ uses an absolute precision ϵ instead of the relative precision $\delta\lambda_{\rm tgt}$. Therefore, the overall complexity is bounded by

$$\left(\frac{\ln(\lambda_0/\lambda_{\rm tgt})}{\ln(1/\eta)}\left(\log_{\frac{1}{\theta_{sc}}}\left(\frac{C}{\delta}\right)+D\right)+\log_{\gamma_{sc}}\max\left(1,\frac{\lambda_{\rm tgt}C}{\epsilon}\right)+D\right)\sqrt{\kappa_{s'}\gamma_{\rm inc}\gamma_{sc}}\;\ln\left(8\left(\frac{\kappa_{s'}\gamma_{\rm inc}\gamma_{sc}}{\theta_{sc}}\right)^2\left(1+\frac{\rho_+(s')}{L_{\min}}\right)^2\right),$$

which is $\widetilde{O}\left(\sqrt{\kappa_{s'}}\ln(1/\epsilon)\right)$. Finally, when Algorithm 4 terminates, we have $\omega_{\lambda_{\rm tgt}}(\hat{x}^{(\rm tgt)})\leq\epsilon$. Therefore we can apply Lemma 17 with $\lambda=\lambda'=\lambda_{\rm tgt}$ and $x=\hat{x}^{(\rm tgt)}$ to obtain the last desired bound in Part 3 of Theorem 4.

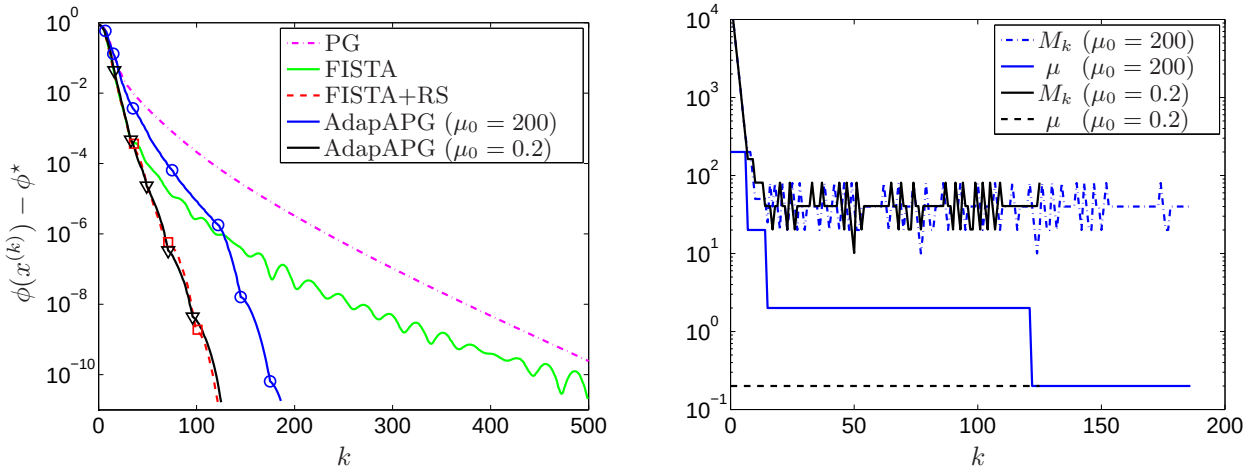

Figure 2. Minimizing a random instance of the log-sum-exp function.

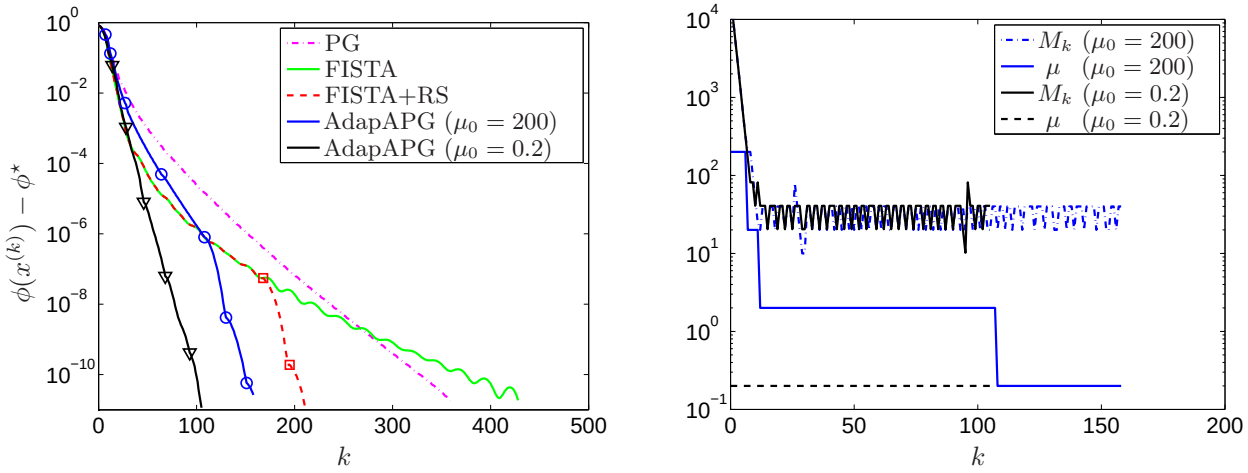

Figure 3. Minimizing another random instance of the log-sum-exp function.

6.7. Experiments on the AdapAPG method

In this subsection, we present extra numerical examples to show that the FISTA+RS method can have less desirable performance compared to the AdaptAPG method when applied to non-quadratic minimizations. We consider the problem of minimizing the *log-sum-exp* function, i.e.,

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) \triangleq \rho \log\left(\sum_{i=1}^{m} \exp\left(\frac{1}{\rho}(a_i^T x - b_i)\right)\right)$$

where all $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$, for $i = 1, \ldots, m$. This corresponds to problem (1) with $\Psi(x) = 0$. In our experiments we took $n = 200$ and $m = 10000$, and generated the a_i's and b_i's randomly with independent, standard normal distribution. Note that this is not really a strongly convex function, since it grows linearly asymptotically. However it is smooth and the region around the optimum may be well approximated by a strongly convex quadratic function. The parameter ρ controls the smoothness of f and is set to $\rho = 0.1$.

Figure 2 shows the convergence characteristics of five different methods on a random instance, and the AdapAPG method was initialized with two different values of μ_0. All methods are equipped with a line search procedure on the Lipschitz constant with the initial value $L_0 = 10000$. We see that the PG method converged with a slow linear rate. FISTA was much faster than PG in the beginning but slowed down eventually due to its lack

of capability of exploiting strong convexity; it also demonstrated nonmotone ripples or bumps in the objective value. FISTA+RS converged fast with a linear rate. For the first run of the AdapAPG method, we intentionally chose a large initial value $\mu_0 = 200$ to test its automatic tuning capability. In fact this initial value is even larger than the restricted Lipschitz constant M_k in later iterations found by the line search procedure; see the right plot in Figure 2. For the second run, we set μ_0 to the final estimate of μ by the first run.

In the left plot in Figure 2, each marker on the curves indicates a restart of the corresponding algorithm. We see that FISTA+RS had three restarts, which was activated by the condition $g_{L_k}(y^{(k-1)})^T(x^{(k)} - x^{(k-1)}) > 0$. Out of the seven restarts of the AdapAPG method with $\mu_0 = 200$, four of them was due to condition A, and three of them was due to condition B (see Algorithm 3). Correspondingly, the right plot in Figure 2 shows that the estimate of the convexity parameter μ was reduced three times, each by a factor of 10, and the final estimate was 0.2. After the last reduction of μ (around $k = 120$), AdapAPG converged fast with a linear rate that is similar to FISTA+RS. For the second run of the AdapAPG method, we used the initial estimate $\mu_0 = 0.2$ directly. As a consequence, all of the five restarts in this case was due to condition A, and the value of μ stayed at the constant 0.2. Without the need for tuning μ, the second run of the AdapAPG converged as fast as FISTA+RS.

From the above comparison, it looks that FISTA+RS is the best method for this particular problem instance, since it demonstrated the fastest convergence without explicit tuning of the convexity parameter. AdapAPG may achieve the same convergence speed, but needs to be initialized with a good estimate of μ to avoid the extra effort involved in tuning it. In general, the procedure of tuning μ costs extra number of iterations, but with a quite modest degradation of performance. For example, Figure 2 showed that AdapAPG with $\mu_0 = 200$ needed an extra 50% iterations while reducing μ by three orders of magnitude.

However, the performance of FISTA+RS vary substantially even on the same class of log-sum-exp functions. Figure 3 illustrates the situation with another random instance in this problem class, in which we simply changed the random seed for generating the problem with the same size. For this instance, the non-monotone behaviour of FISTA appeared quite late, so the first restart of FISTA+RS occurred after $k = 170$. By that time both runs of the AdapAPG method had already finished with high precision (even for the first run which needed to reduce μ three times by a total factor of 1000). Therefore, the AdapAPG method often has a more robust performance guarantee, which is backed by our convergence analysis for general convex functions. In contrast, the FISTA+RS scheme is motivated by the analysis on the quadratic functions, and its behavior on non-quadratics can be hard to predict.