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

Qihang Lin QIHANG-LIN@UIOWA.EDU
The University of Iowa, Iowa City, IA 52245 USA

Lin Xiao LIN.XIAO@MICROSOFT.EDU
Microsoft Research, Redmond, WA 98052 USA

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 31^{st} 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 significantly 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."

To use this algorithm, we need to first choose an initial optimistic estimate $L_{\min}$ for the Lipschitz constant L_f: $0 < L_{\min} \le L_f$, and two adjustment parameters $\gamma_{\text{dec}} \ge 1$ 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 $\phi(x^{(k+1)}) \le \psi_{M_k}(y^{(k)}; x^{(k+1)})$, 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

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

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

parameter: $L_{\min} \ge \mu > 0$, $\gamma_{\text{dec}} \ge 1$
$x^{(-1)} \leftarrow x^{(0)}$
$\alpha_{-1} = 1$
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 $\omega(x^{(k+1)}) \le \hat{\epsilon}$
$\hat{x} \leftarrow x^{(k+1)}$
$\hat{M} \leftarrow M_k$

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 $\phi(x^{(k+1)}) \le \psi_L(y^{(k)}; x^{(k+1)})$
$M_k \leftarrow L$
$g^{(k)} \leftarrow M_k(y^{(k)} - x^{(k+1)})$
$S_k \leftarrow S_L(y^{(k)})$

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

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

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*

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

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

where

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

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*

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

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*

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

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 $\theta_{\text{sc}} \in (0,1)$ be a desired shrinking factor. We check the following two conditions at iteration k of Algorithm 1:

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

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

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 $\omega(x^{(k+1)}) \leq \hat{\epsilon}$
$\hat{x} \leftarrow x^{(k+1)},\ \hat{M} \leftarrow M_k,\ \hat{\mu} \leftarrow \mu$

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 problem with a large regularization parameter λ_0 first, and then gradually decreases the value of λ until the target regularization is reached. In Xiao & Zhang (2013), the PG method is employed to solve the ℓ_1-LS problem for a fixed λ up to an adequate precision, then the solution is used to warm start the next stage. It was shown that under a restricted eigenvalue condition on A, such a homotopy scheme guarantees that all iterates generated by the method 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 $\widetilde{O}(\kappa_s \log(1/\epsilon))$ for certain sparsity level s, where κ_s is the restricted condition number defined in (7), and the notation $\widetilde{O}(\cdot)$ hides additional $\log(\kappa_s)$ factors.

In this section, we show that, by combining the adaptive APG method (Algorithm 3) with the same homotopy continuation scheme, the iteration complexity for solving the ℓ_1-LS problem can be improved to $\widetilde{O}\big(\sqrt{\kappa_{s'}}\log(1/\epsilon)\big)$, with s' slightly larger than s.

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

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**
  $\lambda_{K+1} \leftarrow \eta\lambda_K$
  $\hat{\epsilon}_{K+1} \leftarrow \delta\lambda_{K+1}$
  $\{\hat{x}^{(K+1)}, \hat{M}_{K+1}, \hat{\mu}_{K+1}\}$
    $\leftarrow \mathtt{AdapAPG}\big(\hat{x}^{(K)}, \hat{M}_K, \hat{\mu}_K, \hat{\epsilon}_{K+1}, \lambda_{K+1}\big)$
end for
$\{\hat{x}^{(\mathrm{tgt})}, \hat{M}_{\mathrm{tgt}}\} \leftarrow \mathtt{AdapAPG}\big(\hat{x}^{(N)}, \hat{M}_N, \hat{\mu}_N, \epsilon, \lambda_{\mathrm{tgt}}\big)$
return: $\hat{x}^{(\mathrm{tgt})}$

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

According to Zhang & Huang (2008), the above assumption implies $\|x^\star(\lambda)_{\bar{S}^c}\|_0 \leq \tilde{s}$ whenever $\lambda \geq \lambda_{\mathrm{tgt}}$ (here $\bar{S}^c$ denotes the complement of the support set $\bar{S}$). 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 $\kappa(A, s) < (1+\nu)/(1-\nu)$. See Xiao & Zhang (2013, Section 3) for an example of sufficient RIP conditions. Another sufficient condition is $\kappa(A, \bar{s}+3\tilde{s}) \leq C\tilde{s}/\bar{s}$ with $C = 1/(24(1+\gamma)(3+\gamma_{\mathrm{inc}}))$, which is more accessible but can be very conservative.
- 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 $\widetilde{O}(\sqrt{\kappa_{s'}}\ln(1/\epsilon))$.

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

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

Our $\widetilde{O}\left(\sqrt{\kappa_{s'}}\ln(1/\epsilon)\right)$ complexity of the APG homotopy method improves the $\widetilde{O}\left(\kappa_s\ln(1/\epsilon)\right)$ complexity of PGH in the dependence on restricted condition number. We note that this 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 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.

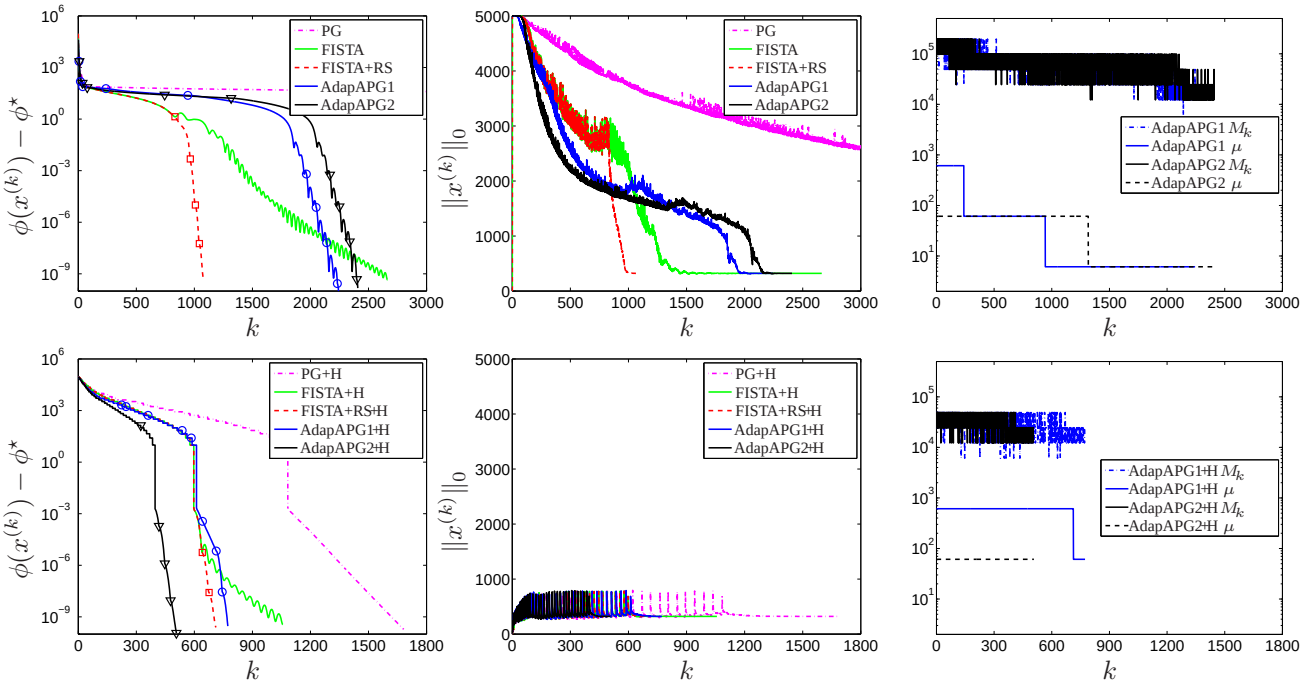

Figure 1. Solving an ill-conditioned ℓ_1-LS problem. AdapAPG1 starts with $\mu_0 = L_0/10$, and AdapAPG2 uses $\mu_0 = L_0/100$.

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 $\gamma_{\rm sc} = 10$, 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 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.

Zhang, C.-H. and Huang, J. The sparsity and bias of the lasso selection in high–dimensional linear regression. *Annals of Statistics*, 36:1567–1594, 2008.