Numer Algor (2013) 64:321–347
DOI 10.1007/s11075-012-9668-5

ORIGINAL PAPER

ParNes: a rapidly convergent algorithm for accurate recovery of sparse and approximately sparse signals

Ming Gu · Lek-Heng Lim · Cinna Julie Wu

Received: 28 June 2011 / Accepted: 5 November 2012 / Published online: 1 December 2012

Abstract In this article, we propose an algorithm, NESTA-LASSO, for the LASSO problem, i.e., an underdetermined linear least-squares problem with a 1-norm constraint on the solution. We prove under the assumption of the *restricted isometry property* (RIP) and a sparsity condition on the solution, that NESTA-LASSO is guaranteed to be almost always locally linearly convergent. As in the case of the algorithm NESTA, proposed by Becker, Bobin, and Candès, we rely on Nesterov's accelerated proximal gradient method, which takes $O(\sqrt{1/\varepsilon})$ iterations to come within $\varepsilon > 0$ of the optimal value. We introduce a modification to Nesterov's method that regularly updates the prox-center in a provably optimal manner. The aforementioned linear convergence is in part due to this modification. In the second part of this article, we attempt to solve the basis pursuit denoising (BPDN) problem (i.e., approximating the minimum 1-norm solution to an underdetermined least squares problem) by using NESTA-LASSO in conjunction with the Pareto root-finding method employed by van den Berg and Friedlander in their SPGL1 solver. The resulting algorithm is called PARNES. We provide numerical evidence to show that it is comparable to currently available solvers.

Keywords Basis pursuit · Newton's method · Pareto curve · Nesterov's method · Compressed sensing · Convex minimization · Duality · LASSO

M. Gu · C. J. Wu (✉)
Department of Mathematics, University of California at Berkeley, Berkeley, CA 94720-3840, USA
e-mail: cinnawu@math.berkeley.edu

M. Gu
e-mail: mgu@math.berkeley.edu

L.-H. Lim
Department of Statistics, University of Chicago, Chicago, IL 60637-1514, USA
e-mail: lekheng@galton.uchicago.edu

1 Introduction

We would like to find a solution to the sparsest recovery problem with noise

$$\min \|x\|_0 \quad \text{s.t.} \quad \|Ax - b\|_2 \leq \sigma. \tag{1}$$

Here, σ specifies the noise level, A is an m-by-n matrix with $m \ll n$, and $\|x\|_0$ is the number of nonzero entries of x. This problem comes up in fields such as image processing [33], seismics [24, 25], astronomy [8], and model selection in regression [16]. Since (1) is known to be ill-posed and NP-hard [21, 26], various convex, l_1-relaxed formulations are often used.

Relaxing the 0-norm in (1) gives the basis pursuit denoising (BPDN) problem

$$\text{BP}(\sigma) \quad \min \|x\|_1 \quad \text{s.t.} \quad \|Ax - b\|_2 \leq \sigma. \tag{2}$$

The special case of $\sigma = 0$ is the basis pursuit problem [12]. Two other commonly used l_1-relaxations are the LASSO problem [34]

$$\text{LS}(\tau) \quad \min \|Ax - b\|_2 \quad \text{s.t.} \quad \|x\|_1 \leq \tau \tag{3}$$

and the penalized least-squares problem

$$\text{QP}(\lambda) \quad \min \|Ax - b\|_2^2 + \lambda \|x\|_1 \tag{4}$$

proposed by Chen et al. [12]. A large amount of work has been done to show that these formulations give an effective approximation of the solution to (1); see [11, 14, 35]. In fact, under certain conditions on the sparsity of the solution to (1), these formulations can exactly recover the solution whenever A satisfies the *restricted isometry property* (RIP).

There is a wide variety of algorithms which solve the BP(σ), QP(λ), and LS(τ) problems. Refer to Section 5 for descriptions of some of the current algorithms. Our work has been motivated by the accuracy and speed of the recent solvers NESTA and SPGL1. In [27], Nesterov presents an algorithm to minimize a smooth convex function over a convex set with an optimal convergence rate. An extension to the nonsmooth case is presented in [28]. NESTA solves the BP(σ) problem using the nonsmooth version of Nesterov's work.

For appropriate parameter choices of σ, λ, and τ, the solutions of BP(σ), QP(λ), and LS(τ) coincide [37]. Although the exact dependence is usually hard to compute [37], there are solution methods which exploit these relationships. The MATLAB solver SPGL1 is based on the Pareto root-finding method [37] which solves BP(σ) by approximately solving a sequence of LS(τ) problems. In SPGL1, the LS(τ) problems are solved using a spectral projected-gradient (SPG) method.

While we are ultimately interested in solving the BPDN problem in (2), our main result is an algorithm for solving the LASSO problem (3). Our algorithm, NESTA-LASSO (cf. Algorithm 3), essentially uses Nesterov's work to solve the LASSO

problem. We introduce one improvement to Nesterov's original method, namely, we update the prox-center every K steps instead of fixing it throughout the algorithm. With this modification, we prove in Theorem 3 that NESTA-LASSO is guaranteed to be almost always locally linearly convergent for sufficiently large K, as long as the solution is s-sparse and A satisfies the restricted isometry property of order $2s$. In fact, Theorem 3 also provides the choice for the optimal K.

Finally, we show that replacing the SPG method in the Pareto root-finding procedure, used in SPGL1, with our NESTA-LASSO method leads to an effective method for solving BP(σ). We call this modification PARNES and compare its efficacy with the state-of-the-art solvers presented in Section 5.

1.1 Notation, terminology, and assumptions

In this paper, a vector is s-sparse if it has exactly s nonzero elements. We say that a vector is *at least* s-sparse if it has at most s nonzero elements. For a nonzero, s-sparse vector $x \in \mathbb{R}^n$, let I_x be the set of indices of the nonzero coefficients of x, i.e. the support of x; $\overline{x}$ is the vector containing the nonzero elements of x. For an $I \subseteq \{1, \ldots, n\}$, I^c is the complement of I. Given a matrix $A \in \mathbb{R}^{m \times n}$ and $I \subseteq \{1, \ldots, n\}$, A_I is the submatrix of A containing the j-th columns of A where $j \in I$. Throughout the paper, we use MATLAB terminology to describe vectors and matrices. Thus, $x[s : r]$ represents the subvector of x containing elements s to r. For a set S, let $\text{int}(S)$ be the interior of S and ∂S be the boundary of S.

Throughout the paper, we make the blanket assumption that $b \in \text{range}(A)$. That is, $Ax - b = 0$ is always possible. In many applications, A has full rank and therefore automatically satisfies this assumption; see [37].

1.2 Organization of the paper

In Section 2, we present and describe the background of NESTA-LASSO. We show in Section 3 that, under some reasonable assumptions, NESTA-LASSO is almost always locally linearly convergent. In Section 4, we describe the Pareto root-finding procedure behind the BPDN solver SPGL1 and show how NESTA-LASSO can be used to solve a subproblem. Section 5 describes some of the available algorithms for solving BPDN and the equivalent QP(λ) problem. Lastly, in Section 6, we show in a series of numerical experiments that using NESTA-LASSO in SPGL1 to solve BPDN is comparable with current competitive solvers.

2 NESTA-LASSO

We present the main parts of our method to solve the LASSO problem. Our algorithm, NESTA-LASSO (cf. Algorithm 3), is an application of the accelerated proximal gradient algorithm of Nesterov [27] outlined in Section 2.1. Additionally, we have a prox-center update improving convergence which we describe in Section 3. In each iteration, we use the fast l_1-projector of Duchi et al. [15] given in Section 2.3.

2.1 Nesterov's algorithm

Let $Q \subseteq \mathbb{R}^n$ be a convex closed set. Let $f : Q \to \mathbb{R}$ be smooth, convex and, Lipschitz differentiable with L as the Lipschitz constant of its gradient, i.e.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \quad \text{for all } x, y \in Q.$$

Nesterov's accelerated proximal gradient algorithm iteratively defines a sequence x_k as a judiciously chosen convex combination of two other sequences y_k and z_k, which are in turn solutions to two quadratic optimization problems on Q. The sequence z_k involves a strongly convex *prox-function*, $d(x)$, which satisfies

$$d(x) \geq \frac{\alpha}{2}\|x - c\|_2^2. \tag{5}$$

For simplicity, we have chosen the right-hand side of (5) with $\alpha = 1$ as our prox-function throughout this paper. The c in the prox-function is called the *prox-center*. With this prox-function, we have:

$$\begin{aligned}
y_k &= \operatorname*{argmin}_{y \in Q} \nabla f(x_k)^\top (y - x_k) + \frac{L}{2}\|y - x_k\|_2^2, \\
z_k &= \operatorname*{argmin}_{z \in Q} \sum_{i=0}^{k} \frac{i+1}{2}\left[f(x_i) + \nabla f(x_i)^\top (z - x_i)\right] + \frac{L}{2}\|z - c\|_2^2, \\
x_k &= \frac{2}{k+3} z_k + \frac{k+1}{k+3} y_k.
\end{aligned}$$

Nesterov showed that if x^* is the optimal solution to

$$\min_{x \in Q} f(x),$$

then the iterates defined above satisfy

$$f(y_k) - f(x^*) \leq \frac{L}{k(k+1)}\|x^* - c\|_2^2 = O\left(\frac{L}{k^2}\right).$$

An implication is that the algorithm requires $O(\sqrt{L/\varepsilon})$ iterations to bring $f(y_k)$ to within $\varepsilon > 0$ of the optimal value.

Algorithm 1 Accelerated proximal gradient method for convex minimization

Input: function f, gradient ∇f, Lipschitz constant L, prox-center c.
Output: $x^* = \operatorname{argmin}_{x \in Q} f(x)$

1: initialize x_0;
2: **for** $k = 0, 1, 2, \ldots,$ **do**
3: compute $f(x_k)$ and $\nabla f(x_k)$;
4: $y_k = \operatorname{argmin}_{y \in Q} \nabla f(x_k)^\top (y - x_k) + \frac{L}{2}\|y - x_k\|_2^2$;
5: $z_k = \operatorname{argmin}_{z \in Q} \sum_{i=0}^{k} \frac{i+1}{2}[f(x_i) + \nabla f(x_i)^\top (z - x_i)] + \frac{L}{2}\|z - c\|_2^2$;
6: $x_k = \frac{2}{k+3} z_k + \frac{k+1}{k+3} y_k$;
7: **end for**

In [28], Nesterov extends his work to minimize nonsmooth convex functions f. Nesterov shows that one can obtain the minimum by applying his algorithm for smooth minimization to a smooth approximation f_μ of f. Since ∇f_μ is shown to have Lipschitz constant $L_\mu = 1/\mu$, if μ is chosen to be proportional to ε, it takes $O\left(\frac{1}{\varepsilon}\right)$ iterations to bring $f(x_k)$ within ε of the optimal value.

The recent algorithm NESTA solves BP(σ) using Nesterov's algorithm for nonsmooth minimization. Our algorithm, NESTA-LASSO, solves LS(τ) using Nesterov's smooth minimization algorithm. In [29], Nesterov suggests an algorithm for minimizing composite functions which has a complexity of $O(\frac{1}{\epsilon^{1/2}})$. We are motivated by the accuracy and speed of NESTA, and the fact that the smooth version of Nesterov's algorithm has a faster convergence rate than the nonsmooth version.

2.2 NESTA-LASSO-K: an accelerated proximal gradient algorithm for LASSO

We apply Nesterov's accelerated proximal gradient method, Algorithm 1, to the LASSO problem LS(τ). We make one slight improvement to Algorithm 1. Namely, we update our prox-centers every K steps (cf. Algorithm 2); that is, Algorithm 1 is restarted every K iterations with a new prox-center. We will see that this leads to local linear convergence under a suitable application of RIP (see Corollary 1 for details). In fact, we show in Section 3 that the prox-centers may be updated in an optimal fashion (cf. Algorithm 3).

In our case, $f = \frac{1}{2}\|b - Ax\|_2^2$, $\nabla f = A^\top(Ax - b)$, and Q is the 1-norm ball $\|x\|_1 \le \tau$. The initial point x_0 is used as the prox-center c. To compute the iterate y_k, we have

$$\begin{aligned} y_k &= \operatorname*{argmin}_{\|y\|_1 \le \tau} \nabla f(x_k)^\top (y - x_k) + \frac{L}{2}\|y - x_k\|_2^2 \\ &= \operatorname*{argmin}_{\|y\|_1 \le \tau} y^\top y - 2(x_k - \nabla f(x_k)/L)^\top y \\ &= \operatorname*{argmin}_{\|y\|_1 \le \tau} \|y - (x_k - \nabla f(x_k)/L)\|_2 \\ &= \operatorname{proj}_1 (x_k - \nabla f(x_k)/L, \tau) \end{aligned}$$

where $\operatorname{proj}_1(v, \tau)$ returns the projection of the vector v onto the 1-norm ball of radius τ. By similar reasoning, computing z_k can be shown to be equivalent to computing

$$z_k = \operatorname{proj}_1 \left(c - \frac{1}{L} \sum_{i=0}^{k} \frac{i+1}{2} \nabla f(x_i), \tau \right).$$

In each iteration, we use the fast l_1-projector proj_1 described in the next section.

In NESTA-LASSO-K, Nesterov's method is restarted every K steps with the new prox-center $\operatorname{proj}_1(y_{iK} - \nabla f(y_{iK})/L, \tau)$. Here, y_{iK-1} is the K-th iterate of Nesterov's method after the i-th prox-center change; see Algorithm 2. In NESTA-LASSO, Nesterov's method is restarted in the same manner, except K is chosen in an optimal way. Algorithms 2 and 3 are stopped when the duality gap η_k is sufficiently small.

Algorithm 2 NESTA-LASSO-K algorithm with prox-center updates every K steps

Input: initial point x_0, LASSO parameter τ, tolerance η, steps to update K
Output: $x_\tau = \operatorname{argmin}\{\|b - Ax\|_2 : \|x\|_1 \le \tau\}$.

for $j = 0, \ldots, j_{\max}$, **do**
  $c_j = x_0$, $h_0 = 0$, $r_0 = b - Ax_0$, $g_0 = -A^\top r_0$,
  $\eta_0 = \|r_0\|_2 - (b^\top r_0 - \tau\|g_0\|_\infty)/\|r_0\|_2$;
  for $k = 0, \ldots, K-1$ **do**
    $y_k = \text{proj}_1(x_k - g_k/L, \tau)$;
    $h_k = h_k + \frac{k+1}{2} g_k$;
    $z_k = \text{proj}_1(c_j - h_k/L, \tau)$;
    $x_k = \frac{2}{k+3} z_k + \frac{k+1}{k+3} y_k$;
    $r_k = b - Ax_k$;
    $g_k = -A^\top r_k$;
    $\eta_k = \|r_k\|_2 - (b^\top r_k - \tau\|g_k\|_\infty)/\|r_k\|_2$;
  end for
  $x_0 = \text{proj}_1(y_k + A^\top(b - Ay_k)/L, \tau)$;
  if $\eta_k \le \eta$ **then**
    return $x_\tau = y_k$;
  end if
end for

Algorithm 3 NESTA-LASSO algorithm with optimal prox-center updates

Input: initial point x_0, LASSO parameter τ, tolerance η.
Output: $x_\tau = \operatorname{argmin}\{\|b - Ax\|_2 : \|x\|_1 \le \tau\}$.

for $j = 0, \ldots, j_{\max}$, **do**
  $c_j = x_0$, $h_0 = 0$, $r_0 = b - Ax_0$, $g_0 = -A^\top r_0$,
  $\eta_0 = \|r_0\|_2 - (b^\top r_0 - \tau\|g_0\|_\infty)/\|r_0\|_2$;
  for $k = 0, \ldots, k_{\text{opt}} - 1$, **do**
    if $\eta_k \le e^{-2}\eta_0$ **then**
      return y_k, η_k
    end if
    $y_k = \text{proj}_1(x_k - g_k/L, \tau)$;
    $h_k = h_k + \frac{k+1}{2} g_k$;
    $z_k = \text{proj}_1(c_j - h_k/L, \tau)$;
    $x_k = \frac{2}{k+3} z_k + \frac{k+1}{k+3} y_k$;
    $r_k = b - Ax_k$;
    $g_k = -A^\top r_k$;
    $\eta_k = \|r_k\|_2 - (b^\top r_k - \tau\|g_k\|_\infty)/\|r_k\|_2$;
  end for
  $x_0 = \text{proj}_1(y_k + A^\top(b - Ay_k)/L, \tau)$;
  if $\eta_k \le \eta$ **then**
    return $x_\tau = y_k$;
  end if
end for

2.3 l_1-projector

The projection of an n-vector, d, onto the 1-norm ball, $\|x\|_1 \leq \tau$, is the solution to the minimization problem

$$\text{proj}_1(d, \tau) := \underset{x}{\text{argmin}} \|d - x\|_2 \quad \text{s.t.} \quad \|x\|_1 \leq \tau.$$

Let $\hat{d}$ be a reordering of d with $|\hat{d}_1| \geq ... \geq |\hat{d}_n|$. Then $a = \text{proj}_1(d, \tau)$, is given by

$$a_i = \text{sgn}(d_i) \cdot \max\{0, |d_i| - \eta\} \quad \text{with} \quad \eta = \frac{\left(|\hat{d}_1| + \cdots + |\hat{d}_k|\right) - \tau}{k}, \tag{6}$$

where k is the largest index such that $\eta \leq |\hat{d}_k|$.

See [15], by Duchi et al., and [37] for fast algorithms to compute a. Such algorithms cost $O(n \log n)$ in the worst case but have been shown experimentally to cost much less [37]. The results in [36] imply the two calls to proj_1 in the inner loop of NESTA-LASSO can be reduced to one call, but due to the low cost of proj_1, we do not make this modification.

3 Local linear convergence and optimality

Under reasonable assumptions on the matrix A and the solution x^* of the LASSO problem, we prove that NESTA-LASSO-K almost always has a local linear convergence rate for large enough K. We also show that we can update the prox-centers c in a provably optimal way (NESTA-LASSO). Let y_k be the k-th iterate of Nesterov's accelerated proximal gradient method when minimizing a function f. Recall,

$$f(y_k) - f\left(x^*\right) \leq \frac{L}{k(k+1)} \|x^* - c\|_2^2 \tag{7}$$

where L is the Lipschitz constant for ∇f and c is the prox-center [27, 28].

In our case, $f(x) = \frac{1}{2}\|Ax - b\|_2^2$, where $A \in \mathbb{R}^{m \times n}$ with $m < n$. We will assume that A satisfies the *restricted isometry property* (RIP) of order $2s$ as described in [9, 10]. Namely, there exists a constant $\delta_{2s} \in (0, 1)$ such that

$$(1 - \delta_{2s}) \|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_{2s}) \|x\|_2^2 \tag{8}$$

whenever $\|x\|_0 \leq 2s$. Since the RIP helps ensure that the solution to (1) is closely approximated by the solution to (2) [9], and we are ultimately interested in solving (2), this is a reasonable assumption. Moreover, since we hope to recover the sparse solution to the solution to (1), we assume that the solution x^* to the LASSO problem is s-sparse. We plan to analyze the approximately sparse case for future work.

Under these assumptions, the LASSO problem has a unique solution (see Theorem 5 in [30]). Since the 1-norm ball is compact, the sequence of y_k's converges to the solution x^*.

Lemma 1 *If A satisfies the restricted isometry property* (RIP) *of order $2s$, and the optimal solution x^* is s-sparse, then the sequence of y_k's converges to x^*.*

3.1 Almost sure sparsity of Nesterov's method

We first state and prove the following results before proving our main results, i.e. the local linear convergence of NESTA-LASSO-K and the optimality of NESTA-LASSO. In particular, we show that under certain assumptions on the LASSO problem, the solution is almost always non-degenerate (see Proposition 11), and the iterates of Algorithm 1 are almost always eventually s-sparse. Our first lemma describes when the image of proj_1 is s-sparse.

For $d \in \mathbb{R}^n$, recall from Section 2.3 that if $\hat{d}$ is a reordering of d with $|\hat{d}_1| \geq ... \geq |\hat{d}_n|$, then $a = \text{proj}_1(d, \tau)$ is given by

$$a_i = \text{sgn}(d_i) \cdot \max\left\{0, |d_i| - \eta\right\} \quad \text{with} \quad \eta = \frac{\left(|\hat{d}_1| + \cdots + |\hat{d}_k|\right) - \tau}{k}, \tag{9}$$

where k is the largest index such that $\eta \leq |\hat{d}_k|$. For each $i \in \{1, \ldots, n\}$, define

$$\eta_i := \frac{|\hat{d}_1| + \cdots + |\hat{d}_i| - \tau}{i}.$$

The η_i's satisfy the following property which is used in the proof of our next lemma.

Claim $\eta = \max\left\{\eta_i : i = 1, \ldots, n\right\}$.

Proof Assume, without loss of generality, that $d \geq 0$ and $d = \hat{d}$ so that $d_1 \geq ... \geq d_n \geq 0$. A simple algebraic manipulation shows that $\eta_i - \eta_{i-1} = \frac{1}{i-1}(d_i - \eta_i)$ for $i \in \{2, \ldots, n\}$. Thus, $\text{sgn}(\eta_i - \eta_{i-1}) = \text{sgn}(d_i - \eta_i)$. Suppose $\eta = \eta_k$ for some k. Then $\eta_k \leq d_k$. Since $\text{sgn}(\eta_i - \eta_{i-1}) = \text{sgn}(d_i - \eta_i)$, it follows that $\eta_{k-1} \leq \eta_k$ and so $\eta_{k-1} \leq d_{k-1}$; thus, we can repeatedly apply this argument to show that $\eta_i \leq \eta_k$ for any $i < k$. A similar argument shows that $\eta_i \leq \eta_k$ for any $i > k$. □

Given a nonempty $I \subseteq \{1, \ldots, n\}$ with $|I| = s$ and $\tau > 0$, if $s < n$, define the set

$$C_{I,\tau} := \left\{x \in \mathbb{R}^n : \sum\nolimits_{i \in I} |x_i| - \tau \geq s \cdot |x_j| \text{ for} j \notin I\right\}.$$

If $I = \{1, \ldots, n\}$, let $C_{I,\tau} := \{x \in \mathbb{R}^n : \|x\|_1 \geq \tau\}$. The following lemma shows that proj_1 sends vectors in $C_{I,\tau}$ to vectors that are *at least* s-sparse.

Lemma 2 *Suppose $I \subseteq \{1, \ldots, n\}$ with $|I| = s$. If $d \in C_{I,\tau}$ then $I_{\text{proj}_1(d,\tau)} \subseteq I$. Namely,* $\text{proj}_1(d, \tau)$ *is at least s-sparse.*

Proof Suppose $d \in C_{I,\tau}$. Assume, without loss of generality, that $d \geq 0$ and $d = \hat{d}$ so that $d_1 \geq ... \geq d_n \geq 0$. For simplicity, let $[1], \ldots, [n]$ be a labeling of the indices of d so that $I = \{[1], \ldots, [s]\}$, $d_{[1]} \geq \ldots \geq d_{[s]}$, and $d_{[s+1]} \geq \ldots \geq d_{[n]}$.

By (9), $a_{[s+1]} \geq \ldots \geq a_{[n]}$, so it is enough to show that $a_{[s+1]} = 0$. Since $d \in C_{I,\tau}$,

$$s \cdot d_{[s+1]} \leq d_{[1]} + \ldots + d_{[s]} - \tau. \tag{10}$$

Let $r \leq s$ be the largest index such that $d_{[r]} \geq d_{[s+1]}$. Such an r exists since $s \cdot d_{[1]} \geq d_{[1]} + \cdots + d_{[s]} - \tau \geq s \cdot d_{[s+1]}$. By (10),

$$\begin{aligned} r \cdot d_{[s+1]} &\leq d_{[1]} + \cdots + d_{[r]} + \left(d_{[r+1]} - d_{[s+1]}\right) + \cdots + \left(d_{[s]} - d_{[s+1]}\right) - \tau \\ &\leq d_{[1]} + \cdots + d_{[r]} - \tau, \end{aligned}$$

which implies,

$$d_{[s+1]} \leq \frac{d_{[1]} + \cdots + d_{[r]} + d_{[s+1]} - \tau}{r+1} = \eta_{r+1}.$$

The last equality holds since we assumed that r is the largest index such that $r \leq s$ and $d_{[r]} \geq d_{[s+1]}$. Thus, $d_{[1]} + \cdots + d_{[r]} + d_{[s+1]} = d_1 + \cdots + d_r + d_{r+1}$. By the above claim, $d_{[s+1]} \leq \eta$, and so $a_{[s+1]} = 0$ by definition of $a_{[s+1]}$. □

The next few lemmas involve the LASSO problem. First note the following LASSO optimality conditions (see e.g. [19] and [20]).

Proposition 1 (LASSO optimality conditions) *For an $x^* \in \mathbb{R}^n$, let $I = I_{x^*}$. Then x^* is the optimal solution to the* LASSO *problem if and only if the gradient, $-\nabla f(x^*) = A^\top(b - Ax^*)$, at x^* satisfies*

$$A_I^\top \left(b - A_I \overline{x}^*\right) = \gamma \cdot \operatorname{sgn}\left(\overline{x}^*\right), \tag{11}$$

$$\|A_{I^c}^\top \left(b - A_I \overline{x}^*\right)\|_\infty \leq \gamma. \tag{12}$$

for some $\gamma \geq 0$. Moreover, there is a one-to-one correspondence between the γ and τ. Following the typical convention, if (12) is a strict inequality, we say that x^* is a non-degenerate solution. Otherwise, we say that x^* is a degenerate solution.

The following lemma relates non-denerate LASSO solutions x^* to the previously defined set $\operatorname{int}(C_{I_{x^*},\tau})$.

Lemma 3 *If x^* is a non-degenerate solution with $I_{x^*} = I$, then $x^* - \nabla f(x^*)/L \in \operatorname{int}(C_{I,\tau})$.*

Proof By (11) and (12), for any $j \notin I$, we have

$$\begin{aligned}\sum_{i\in I}\left|x_i^* + \frac{a_i^\top\left(b - A_I\overline{x}^*\right)}{L}\right| - \tau &= \sum_{i\in I}\left|x_i^* + \frac{\gamma\cdot\operatorname{sgn}\left(x_i^*\right)}{L}\right| - \tau\\ &= \sum_{i\in I}\left|x_i^*\right| + |I|\cdot\frac{\gamma}{L} - \tau\\ &\geq |I|\cdot\left|a_j^\top\left(b - A_I\overline{x}^*\right)\right|/L\\ &= |I|\cdot\left|x_j^* + a_j^\top\left(b - A_I\overline{x}^*\right)\right|/L.\end{aligned}$$

The third equation on the right holds since we must have $\|x^*\|_1 = \tau$. If not, then we must have $Ax^* - b = 0$ which is only possible when x^* is a degenerate solution. □

We now prove that under our assumptions on the LASSO problem, the gradient at the optimal solution will almost always lie in a desirable direction. In other words, we have the following result.

Theorem 1 *Suppose $A \in \mathbb{R}^{m\times n}$ satisfies the restricted isometry property* (RIP) *of order $2s$, and the optimal solution x^* is s-sparse. The solution x^* will almost always be non-degenerate.*

Proof Fix positive integers m, n, and $I \subseteq \{1,\ldots,n\}$ with $|I| = s \leq m$. Define $\text{LS}(m, n, I)$ to be the set of LASSO problems

$$\min \|Ax - b\|_2 \quad \text{s.t.} \quad \|x\|_1 \leq \tau$$

with s-sparse solutions x^* such that $I_{x^*} = I$ and $A \in \mathbb{R}^{m\times n}$ satisfying the RIP of order $2s$. As seen in the proof of Lemma 1, x^* is unique.

The LASSO optimality conditions above say that x^* is the solution to a LASSO problem if and only if $A_I^\top(b - A_I\overline{x}^*) = \gamma\cdot\operatorname{sgn}(\overline{x}^*)$ and $\|A_{I^c}^\top(b - A_I\overline{x}^*)\|_\infty \leq \gamma$ for some $\gamma \geq 0$. Since there is a one-to-one correspondence between τ and γ, we represent each LASSO problem in $\text{LS}(m, n, I)$ with the quadruple $(A_I, A_{I^c}, b, \gamma)$. Following this notation,

$$\text{LS}(m, n, I) = T_1 \cup T_2$$

where

$$\begin{aligned}T_1 &:= \left\{(A_I, A_{I^c}, b, \gamma) \in \text{LS}(m, n, I) : \|A_{I^c}^\top\left(b - A_I\overline{x}^*\right)\|_\infty = \gamma\right\},\\ T_2 &:= \left\{(A_I, A_{I^c}, b, \gamma) \in \text{LS}(m, n, I) : \|A_{I^c}^\top\left(b - A_I\overline{x}^*\right)\|_\infty < \gamma\right\}.\end{aligned}$$

We show that T_1 has Lebesgue measure zero and T_2 has nonzero Lebesgue measure.

By the RIP, A_I has full rank since

$$0 < (1 - \delta_{2s})\|x\|_2^2 \leq \|A_I x\|_2^2 \leq (1 + \delta_{2s})\|x\|_2^2$$

for all nonzero $x \in \mathbb{R}^s$. Thus, $A_I^\top A_I$ is invertible, and if x^* is the solution to $(A_I, A_{I^c}, b, \gamma) \in \text{LS}(m, n, I)$ then

$$\overline{x}^* = \left(A_I^\top A_I\right)^{-1}\left(A_I^\top b - \gamma \cdot \text{sgn}\left(\overline{x}^*\right)\right).$$

Let $U := \{(A_I, A_{I^c}, b, \gamma) \in \mathbb{R}^{m\times s} \times \mathbb{R}^{m\times(n-s)} \times \mathbb{R}^m \times \mathbb{R}^+ : A_I \text{ nonsingular}\}$. For each $w \in \{-1, 1\}^s$, define the function $g_w : U \to \mathbb{R}^{n-s}$ by

$$g_w\left(A_I, A_{I^c}, b, \gamma\right) = \frac{A_{I^c}^\top\left(b - A_I\left(A_I^\top A_I\right)^{-1}\left(A_I^\top b - \gamma \cdot w\right)\right)}{\gamma},$$

If $S := \{x \in \mathbb{R}^{(n-s)} : |x| \le 1\}$ with boundary ∂S and interior $\text{int}(S)$, then

$$T_1 \subseteq \bigcup_w g_w^{-1}(\partial S) \bigcup \mathbb{R}^{m\times s} \times \mathbb{R}^{m\times(n-s)} \times \mathbb{R}^m \times \{0\}.$$

Each component function of g_w involves exactly one row of the variables in $A_{I^c}^\top$, and g_w is the composition of matrix inversion and basic matrix operations. Thus, g_w is a smooth map of constant rank $(n - s)$ on the open set $U \setminus g_w^{-1}(0)$. An application of Theorem 1 of [32] shows that $g_w^{-1}(\partial S)$ has measure zero. Hence, T_1 has Lebesgue measure zero.

To see that T_2 has nonzero measure, note that T_2 is the set of $(A_I, A_{I^c}, b, \gamma) \in U$ such that A satisfies the RIP of order $2s$ intersected with

$$\bigcup_w g_w^{-1}(\text{int}(S)) \cap \left\{(A_I, A_{I^c}, b, \gamma) \in U : \text{sgn}\left(\left(A_I^\top A_I\right)^{-1}\left(A_I^\top b - \gamma \cdot w\right)\right) = w\right\}.$$

Using the triangle inequality, it is easy to see that the former set is open since

$$(1 - \delta_{2s})\,\|x\|_2^2 \le \|Ax\|_2^2 \le (1 + \delta_{2s})\,\|x\|_2^2$$

holds under small perturbations of A. The latter set is open since g_w and $(A_I, A_{I^c}, b, \gamma) \mapsto (A_I^\top A_I)^{-1}(A_I^\top b - \gamma \cdot w)$ are continuous functions for each w. Thus, T_2 is open. Moreover, it is easy to see that if $(A_I, A_{I^c}, b, \gamma) \in T_1$ then there exists a small perturbation E such that $(A_I, A_{I^c} + E, b, \gamma) \in T_2$. If $\text{LS}(m, n, I)$ is nonempty, it must be that T_2 is nonempty and therefore, has nonzero measure.

This argument is easily extended for any $I \subseteq \{1, \ldots, n\}$. Since there are a finite number of I's and a finite union of measure zero sets has measure zero, our lemma holds. □

Let y_k be the k-th iterate of Nesterov's accelerated proximal gradient method applied to the LASSO problem. The previous results allow us to make the following conclusion regarding the sparsity of y_k.

Theorem 2 *Suppose A satisfies the restricted isometry property* (RIP) *of order $2s$, and the optimal solution x^* is s-sparse. The iterates y_k are almost always eventually s-sparse.*

Proof By Lemma 1, the sequence $\{y_k\}$ converges to the optimal solution x^*. Since $x_k = \frac{2}{k+3}z_k + \frac{k+1}{k+3}y_k$ and $\nabla f(x) = A^\top(Ax - b)$ is continuous, the sequence $\{x_k - \nabla f(x_k)/L\}$ converges to $x^* - \nabla f(x^*)/L$.

Theorem 1 says that x^* is almost always non-degenerate, in which case, by Lemma 3, $x^* - \nabla f(x^*)/L \in \text{int}(C_{Ix^*,\tau})$, where $\text{int}(C_{Ix^*,\tau})$ is the interior of $C_{Ix^*,\tau}$. Thus, if x^* is non-degenerate, there exists an N such that for $k \geq N$, $x_k - \nabla f(x_k)/L \in \text{int}(C_{I_{x^*}})$. By Lemma 2, for such k, $y_k = \text{proj}_1(x_k - \nabla f(x_k)/L, \tau)$ is s-sparse. □

3.2 Local linear convergence of NESTA-LASSO

We now show that NESTA-LASSO-K, Algorithm 2, is almost always locally linearly convergent under certain assumptions. First we give some motivation for why we update the prox-centers in NESTA-LASSO-K.

Consider applying Nesterov's accelerated proximal gradient method, Algorithm 1, to the LASSO problem. Suppose A satisfies the *restricted isometry property* (RIP) of order $2s$ and the optimal solution x^* is s-sparse. As seen in Theorem 2, the iterates y_k are almost always eventually s-sparse. Thus, it is reasonable to assume that y_k is s-sparse.

Let $\delta = 1 - \delta_{2s}$ where δ_{2s} is the RIP constant of A. We have

$$\begin{aligned} &\|A\left(x^* - y_k\right)\|_2^2 + 2\left(y_k - x^*\right)^\top A^\top\left(Ax^* - b\right) \\ &\quad = f\left(y_k\right) - f\left(x^*\right) \geq \|A\left(y_k - x^*\right)\|_2^2 \delta \|y_k - x^*\|_2^2. \end{aligned} \tag{13}$$

To see the first inequality, let $y = x^* + \tau(y_k - x^*)$ for $\tau \in [0, 1]$. Due to the convexity of the 1-norm ball, y is feasible. Since x^* is the minimum, for any $\tau \in [0, 1]$,

$$f(y) - f\left(x^*\right) = \tau^2 \|A\left(x^* - y_k\right)\|_2^2 + 2\tau\left(y_k - x^*\right)^\top A^\top(Ax^* - b) \geq 0.$$

Thus, $(y_k - x^*)^\top A^\top(Ax^* - b) \geq 0$. The second inequality follows from (8) since the vector $y_k - x^*$ has at most $2s$ nonzeros. Then from (7), we have

$$\delta\|y_k - x^*\|_2^2 \leq \frac{L}{k(k+1)}\|x^* - c\|_2^2.$$

Putting everything together gives

$$\|y_k - x^*\|_2 \leq \sqrt{\frac{L}{k(k+1)\delta}}\|x^* - c\|_2 \leq \frac{1}{k}\sqrt{\frac{L}{\delta}}\|c - x^*\|_2. \tag{14}$$

The above relation and (7) suggest that when solving the LASSO problem, we can speed up Algorithm 1 by updating the prox-center, c, every K steps. With our assumptions, we prove in the first part of following theorem that for every $K > \sqrt{\frac{L}{\delta}}$, restarting Algorithm 1 every K steps with the new prox-center, $\text{proj}_1(y_k - \nabla f(y_k)/L, \tau)$, is locally linearly convergent. In the second part of Theorem 3, we prove that there is an optimal number of such steps.

In the following, allow the iterates to be represented by y_{jk} where j is the number of times the prox-center has been changed (the outer iteration) and k is number of iterations after the last prox-center change (the inner iteration).

Theorem 3 *Suppose A satisfies the restricted isometry property of order $2s$ and the solution x^* is s-sparse. The following holds if x^* is non-degenerate and the initial point $p_0 := x_0$ in Algorithm 2 is chosen to be sufficiently close to x^*.*

(i) *Algorithm 2 is locally linearly convergent for any $K > \sqrt{\frac{L}{\delta}}$.*

(ii) *In Algorithm 2, let j_{tot} be the total number of prox-center changes. The total number of iterations, $j_{\text{tot}} \cdot K$, to get $\|p_j - x^*\|_2 \leq \varepsilon$ is minimized if K is equal to*

$$k_{\text{opt}} := e\sqrt{\frac{L}{\delta}} \tag{15}$$

where e is the base of the natural logarithm. Moreover, for each j,

$$\|p_j - x^*\|_2 \leq \frac{1}{e^j}\|p_0 - x^*\|_2.$$

Proof

(i) By Lemma 3, $x^* - \nabla f(x^*)/L \in \text{int}(C_{I_{x^*},\tau})$, where $\text{int}(C_{I_{x^*},\tau})$ is the interior of $C_{I_{x^*},\tau}$. Let U_α be a ball of radius $\alpha > 0$, centered at $x^* - \nabla f(x^*)/L$, such that $U_\alpha \subseteq \text{int}(C_{I_{x^*},\tau})$. By continuity, we may choose an $\epsilon > 0$ such that $\|x - x^*\|_2 < \epsilon$ implies $x - \nabla f(x)/L \in U_\alpha$.
Now choose $\beta > 0$ such that for all $\|x\|_1 \leq \tau$, $f(x) - f(x^*) < \beta$ implies $\|x - x^*\|_2 < \epsilon$. To see that $\beta > 0$ exists, suppose for a contradiction that $\forall\, n$, $\exists\, x_n$ with $\|x_n\|_1 \leq \tau$ where $f(x_n) - f(x^*) < 1/n$ but $\|x_n - x^*\|_2 \geq \epsilon$. Since the 1-norm ball is compact, there is a subsequence $\{x_{n_k}\}$ of $\{x_n\}$ converging to some x'. By continuity, $f(x_{n_k})$ converges to $f(x')$. As mentioned right before the statement of Lemma 1, x^* is a unique minimum. Thus, $f(x') \neq f(x^*)$ contradicting the assumption that $f(x_n)$ converges to $f(x^*)$.
We now show that Algorithm 2 is linearly convergent if the initial prox-center p_0 is close enough to x^*. Suppose $\|p_0 - x^*\|_2 < \beta/L$. Then (7) implies

$$f(y_{1K}) - f\left(x^*\right) \leq \frac{L}{K(K+1)}\|p_0 - x^*\|_2^2 < \beta,$$

and so $\|y_{1K} - x^*\| < \epsilon$. By Lemma 2, $p_1 = \text{proj}_1(y_{1K} - \nabla f(y_{1K})/L, \tau)$ is s-sparse, and by (13),

$$\delta\|p_1 - x^*\|_2^2 \leq f(p_1) - f\left(x^*\right). \tag{16}$$

Note that p_1 is the result of a step of the *projected gradient method*, i.e. $x_{k+1} = \text{proj}_1(x_k - \nabla f(x_k)/L, \tau)$. Since this method is monotonically decreasing (see [41] for a proof),

$$f(p_1) - f\left(x^*\right) \leq f(y_{1K}) - f\left(x^*\right). \tag{17}$$

Combining (16) and (17) with (7), gives

$$\|p_1 - x^*\|_2 \leq \frac{1}{K}\sqrt{\frac{L}{\delta}}\|p_0 - x^*\|_2.$$

Since we assume that $K > \sqrt{\frac{L}{\delta}}$, we have $\|p_1 - x^*\|_2 < \beta/L$. Thus, the above arguments can be repeatedly applied to show that for any j,

$$\|p_j - x^*\|_2 \leq \left(\frac{1}{K}\sqrt{\frac{L}{\delta}}\right)^j \|p_0 - x^*\|_2. \tag{18}$$

(ii) First observe that (18) implies

$$\|p_j - x^*\|_2 \leq \left(\frac{1}{K}\sqrt{\frac{L}{\delta}}\right)^j \|p_0 - x^*\|_2 \leq \varepsilon\|p_0 - x^*\|_2$$

when

$$j \log\left(\frac{1}{K}\sqrt{\frac{L}{\delta}}\right) = \log\varepsilon.$$

This relation allows us to choose K to minimize the product $j \cdot K$. Since

$$j \cdot K = \frac{K \log\varepsilon}{\log\sqrt{L/\delta} - \log K},$$

taking derivative of the expression on the right shows that $j \cdot K$ is minimized when

$$K = e\sqrt{\frac{L}{\delta}},$$

where e is the base of the natural logarithm. The total number of iterations will then be

$$j_{\text{tot}} \cdot K = -e\sqrt{\frac{L}{\delta}}\log\varepsilon.$$

□

Theorem 1 implies that we almost always have local linear convergence:

Corollary 1 *If A satisfies the restricted isometry property of order $2s$ and the solution x^* is s-sparse, Algorithm 2 is almost always locally linearly convergent for any $K > \sqrt{\frac{L}{\delta}}$.*

Table 1 Number of products with A and $A^\top$ for NESTA-LASSO without prox-center updates (cf. Algorithm 1) and NESTA-LASSO with prox-center updates (cf. Algorithm 3)

Number of rows of A	Number of columns of A	τ	N_A	N_A^{update}
100	256	6.28	69	37
200	512	12.6	77	47
400	1,024	25.1	157	45

These values are given by N_A and N_A^{update} respectively

In our experiments, there are some cases where updating the prox-center will eventually cause the duality gap to jump to a higher value than the previous iteration. This can cause the algorithm to run for more iterations than necessary. A check is added to prevent the prox-center from being updated if it no longer helps.

In Table 1, we give some results showing that updating the prox-center is effective when using NESTA-LASSO to solve the LASSO problem.

4 PARNES

In applications where the noise level of the problem is approximately known, it is preferable to solve BP(σ). The Pareto root-finding method used by van den Berg and Friedlander [37] interprets BP(σ) as finding the root of a single-variable nonlinear equation whose graph is called the Pareto curve. Their implementation of this approach is called SPGL1. In SPGL1, an inexact version of Newton's method is used to find the root, and at each iteration, an approximate solution to the LASSO problem, LS(τ), is found using an SPG approach. Refer to [13] for more information on the inexact Newton method. In Section 6, we show experimentally that using NESTA-LASSO in place of the SPG approach for solving the LS(τ) subproblems can lead to improved results. We call this version of the Pareto root-finding method, PARNES. The pseudocode of PARNES is given in Algorithm 4.

4.1 Pareto curve

Suppose A and b are given, with $0 \neq b \in \text{range}(A)$. The points on the Pareto curve are given by $(\tau, \varphi(\tau))$ where $\varphi(\tau) = \|Ax_\tau - b\|_2$, $\tau = \|x_\tau\|_1$, and x_τ solves LS(τ). The Pareto curve gives the optimal trade-off between the 2-norm of the residual and 1-norm of the solution to LS(τ). It can also be shown that the Pareto curve also characterizes the optimal trade-off between the 2-norm of the residual and 1-norm of the solution to BP(σ). Refer to [37] for a more detailed explanation of these properties of the Pareto curve. An example of a Pareto curve is shown in Fig. 1.

Let τ_{BP} be the optimal objective value of BP(0). The Pareto curve is restricted to the interval $\tau \in [0, \tau_{\text{BP}}]$ with $\varphi(0) = \|b\|_2 > 0$ and $\varphi(\tau_{\text{BP}}) = 0$. The following theorem, proven by van den Berg and Friedlander, shows that the Pareto curve is convex,

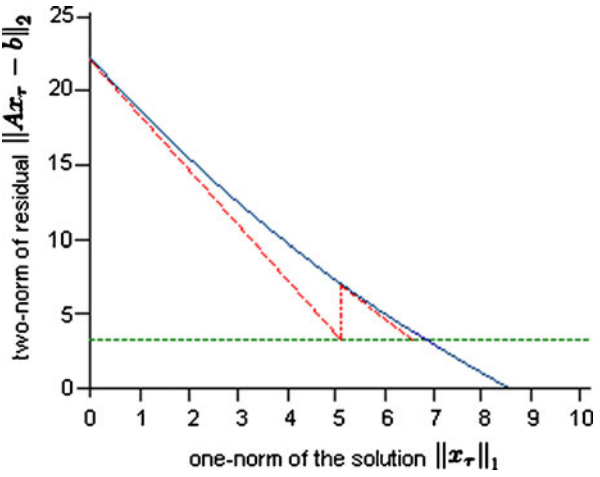

Fig. 1 An example of a Pareto curve. The *solid line* is the Pareto curve; the *dotted red lines* give two iterations of Newton's method

strictly decreasing over the interval $\tau \in [0, \tau_{\text{BP}}]$, and continuously differentiable for $\tau \in (0, \tau_{\text{BP}})$.

Proposition 2 [37] *The function φ is*

(i) *convex and nonincreasing;*
(ii) *continuously differentiable for $\tau \in (0, \tau_{\text{BP}})$ with $\varphi'(\tau) = -\lambda_\tau$ where $\lambda_\tau = \|A^T y_\tau\|_\infty$ is the optimal dual variable to $\text{LS}(\tau)$ and $y_\tau = r_\tau / \|r_\tau\|_2$ with $r_\tau = Ax_\tau - b$;*
(iii) *strictly decreasing and $\|x_\tau\|_1 = \tau$ for $\tau \in [0, \tau_{\text{BP}}]$.*

4.2 Root finding

Since the Pareto curve characterizes the optimal trade-off for both $\text{BP}(\sigma)$ and $\text{LS}(\tau)$, solving $\text{BP}(\sigma)$ for a fixed σ can be interpreted as finding a root of the non-linear equation $\varphi(\tau) = \sigma$. The iterations consist of finding the solution to $\text{LS}(\tau)$ for a sequence of parameters $\tau_k \to \tau_\sigma$ where τ_σ is the optimal objective value of $\text{BP}(\sigma)$.

Applying Newton's method to φ gives

$$\tau_{k+1} = \tau_k + \big(\sigma - \varphi(\tau_k)\big)/\varphi'(\tau_k).$$

Since φ is convex, strictly decreasing and continuously differentiable, $\tau_k \to \tau_\sigma$ superlinearly for all initial values $\tau_0 \in (0, \tau_{\text{BP}})$ (see Proposition 1.4.1 in [6]). By Proposition 2, $\varphi(\tau_k)$ is the optimal value to $\text{LS}(\tau_k)$ and $\varphi'(\tau_k)$ is the dual solution to $\text{LS}(\tau_k)$. Since evaluating $\varphi(\tau_k)$ involves solving a potentially large optimization problem, an inexact Newton method is carried out with approximations of $\varphi(\tau_k)$ and $\varphi'(\tau_k)$.

Let $\overline{y}_\tau$ and $\overline{\lambda}_\tau$ be the approximations of the y_τ and λ_τ defined in Proposition 2. The duality gap at each iteration is given by

$$\eta_\tau = \|\overline{r}_\tau\|_2 - \left(b^T \overline{y}_\tau - \tau \overline{\lambda}_\tau\right).$$

The following convergence result has been proven by van den Berg and Friedlander.

Theorem 4 [37] *Suppose A has full rank, $\sigma \in (0, \|b\|_2)$, and the inexact Newton method generates a sequence $\tau_k \to \tau_\sigma$. If $\eta_k := \eta_{\tau_k} \to 0$ and τ_0 is close enough to τ_σ, we have*

$$|\tau_{k+1} - \tau_\sigma| = \gamma_1 \eta_k + \zeta_k |\tau_k - \tau_\sigma|,$$

where $\zeta_k \to 0$ and γ_1 is a positive constant.

4.3 Solving the LASSO problem

Approximating $\varphi(\tau_k)$ and $\varphi'(\tau_k)$ require approximately minimizing LS(τ). The solver SPGL1 uses a spectral projected-gradient (SPG) algorithm. T he method follows the algorithm by Birgin et al. [7] and is shown to be globally convergent. The costs include evaluating Ax, $A^\top r$, and a projection onto the 1-norm ball $\|x\|_1 \le \tau$. In PARNES, we replace this SPG algorithm with our algorithm, NESTA-LASSO (cf. Algorithm 3).

Algorithm 4 PARNES: Pareto curve method with NESTA-LASSO

Input: initial point x_0, BPDN parameter σ, tolerance η.
Output: $x_\sigma = \operatorname{argmin}\{\|x\|_1 : \|Ax - b\|_2 \le \sigma\}$

$\tau_0 = 0, \varphi_0 = \|b\|_2, \varphi_0' = \|A^\top b\|_\infty$;
for $k = 0, \ldots, k_{\max}$, **do**
  $\tau_{k+1} = \tau_k + (\sigma - \varphi_k)/\varphi_k'$;
  $x_{k+1} = \text{NESTA-LASSO}(x_k, \tau_{k+1}, \eta)$;
  $r_{k+1} = b - Ax_{k+1}$;
  $\varphi_{k+1} = \|r_{k+1}\|_2$;
  $\varphi_{k+1}' = -\|A^\top r_{k+1}\|_\infty / \|r_{k+1}\|_2$;
  if $\|r_{k+1}\|_2 - \sigma \le \eta \cdot \max\{1, \|r_{k+1}\|_2\}$ **then**
    return $x_\sigma = x_{k+1}$;
  end if
end for

5 Other solution techniques and tools

In the our numerical experiments, we compare PARNES with other state-of-the-art methods. The algorithms we test and their experimental details are described below. Note that the algorithms either solve BP(σ) or QP(λ).

5.1 NESTA [5]

NESTA is used to solve BP(σ). Its code is available at http://www.acm.caltech.edu/~nestanesta. The parameters for NESTA are set to be

$$x_0 = A^\top b, \quad \mu = 0.02,$$

where x_0 is the initial guess and μ is the smoothing parameter for the 1-norm function in BP(σ).

Continuation techniques are used to speed up NESTA in [5]. Such techniques are useful when it is observed that a problem involving some parameter λ is faster for large λ, [22, 31]. Thus, the idea of continuation is to solve a sequence of problems for decreasing values of λ. In the case of NESTA, it is observed that convergence is faster for larger values of μ. When continuation is used in the experiments, there are four continuation steps with $\mu_0 = \|x_0\|_\infty$ and $\mu_t = (\mu/\mu_0)^{t/4}\mu_0$ for $t = 1, 2, 3, 4$.

5.2 GPSR: gradient projection for sparse reconstruction [17]

GPSR is used to solve the penalized least-squares problem QP(λ). The code is available at http://www.lx.it.pt/~mtf/GPSR. The problem is first recast as a bound-constrained quadratic program (BCQP) by using a standard change of variables on x. Here, $x = u_1 - u_2$, and the variables are now given by $[u_1, u_2]$ where the entries are positive. The new problem is then solved using a gradient projection (GP) algorithm. The parameters are set to the default values in the following experiments.

A version of GPSR with continuation is also tested. The number of continuation steps is 40, the variable TOLERANCEA is set to 10^{-3}, and the variable MINITERA is set to 1. All other parameters are set to their default values.

5.3 SPARSA: sparse reconstruction by separable approximation [18]

SPARSA is used to minimize functions of the form $\phi(x) = f(x) + \lambda c(x)$ where f is smooth and c is non-smooth and non-convex. The QP(λ) problem is a special case of functions of this form. The code for SPARSA is available at http://www.lx.it.pt/~mtf/SpaRSA.

In a sense, SPARSA is an iterative shrinkage/thresholding algorithm. Utilizing continuation and a Brazilai-Borwein heuristic [3] to find step sizes, the speed of the algorithm can be increased. The number of continuation steps is set to 40 and the variable MINITERA is set to 1. All remaining variables are set to their default values.

5.4 SPGL1 [37] and SPARCO [38]

SPGL1 is available at http://www.cs.ubc.ca/labs/scl/spgl1. The parameters for our numerical experiments are set to their default values.

Due to the vast number of available and upcoming algorithms for sparse reconstruction, the authors of SPGL1 and others have created SPARCO [38]. In SPARCO, they provide a much needed testing framework for benchmarking algorithms. It

consists of a large collection of imaging, compressed sensing, and geophysics problems. Moreover, it includes a library of standard operators which can be used to create new test problems. SPARCO is implemented in MATLAB and was originally created to test SPGL1. The toolbox is available at http://www.cs.ubc.ca/labs/scl/sparco.

5.5 FISTA: fast iterative soft-thresholding algorithm [4]

FISTA solves QP(λ). It can be thought of as a simplified version of the Nesterov algorithm in Section 2.1 since it involves two sequences of iterates instead of three. In Section 4.2 of [5], FISTA is shown to give very accurate solutions provided enough iterations are taken. Due to its ease of use and accuracy, FISTA is used to compute reference solutions in [5] and in this paper. The code for FISTA can be found in the NESTA experiments code at http://www.acm.caltech.edu/~nesta.

5.6 FPC: fixed point continuation [22, 23]

FPC solves the general problem $\min_x \|x\|_1 + \lambda f(x)$ where $f(x)$ is differentiable and convex. The special case with $f(x) = \frac{1}{2}\|Ax - b\|_2^2$ is the QP(λ) problem. The algorithm is available at http://www.caam.rice.edu/~optimization/L1/fpc.

FPC is equivalent to iterative soft-thresholding. The approach is based on the observation that the solution solves a fixed-point equation $x = F(x)$ where the operator F is a composition of a gradient descent-like operator and a shrinkage operator. It can be shown that the algorithm has q-linear convergence and also, finite-convergence for some components of the solution. Since the parameter λ affects the speed of convergence, continuation techniques are used to slowly decrease λ for faster convergence. A more recent version of FPC, FPC-BB, uses Brazilai-Borwein steps to speed up convergence. Both versions of FPC are tested with their default parameters.

5.7 FPC-AS: fixed-point continuation and active set [39]

FPC-AS is an extension of FPC into a two-stage algorithm which solves QP(λ). The code can be found at http://www.caam.rice.edu/~optimization/L1/fpc. It has been shown in [22] that applying the shrinkage operator a finite number of times yields the support and signs of the optimal solution. Thus, the first stage of FPC-AS involves applying the shrinkage operator until an active set is determined. In the second stage, the objective function is restricted to the active set and $\|x\|_1$ is replaced by $c^T x$ where c is the vector of signs of the active set. The constraint $c_i \cdot x_i > 0$ is also added. Since the objective function is now smooth, many available methods can now be used to solve the problem. In the following tests, the solvers L-BFGS and conjugate gradients, CG (referred to as FPC-AS (CG)), are used. Continuation methods are used to decrease λ to increase speed. For experiments involving approximately sparse signals, the parameter controlling the estimated number of nonzeros is set to n, and the maximum number of subspace iterations is set to 10. The other parameters are set to their default values. All other experiments were tested with the default parameters.

5.8 Bregman iteration [40]

The Bregman Iterative algorithm consists of solving a sequence of QP(λ) problems for a fixed λ and updated observation vectors b. Each QP(λ) is solved using the Brazilai-Borwein version of FPC. Typically, very few (around four) outer iterations are needed. Code for the Bregman algorithm can be found at http://www.caam.rice.edu/~optimization/L1/2006/10/bregman-iterative-algorithms-for.html. All parameters are set to their default values.

5.9 C-SALSA [1, 2]

This state-of-the-art method solves BP(σ) and has been shown to be competitive with SPGL1 and NESTA. The method solves the general constrained optimization problem

$$\min_x \phi(x) \text{ s.t.} \|Ax - b\|_2 \leq \epsilon.$$

First, the method transforms the problem into an unconstrained problem which is then transformed into a different constrained problem and then solved with an augmented Lagrangian scheme.

The algorithm requires a method to compute the inverse of $(A^\top A + \alpha I)$ with $\alpha > 0$ and an efficient method for computing the denoising operator associated with ϕ. We have hand-tuned the parameters μ_1 and μ_2 for optimal performance. The code for C-SALSA can be found at http://cascais.lx.it.pt/~mafonso/salsa.html.

6 Numerical results

In the NESTA paper [5] extensive experiments are carried out, comparing the effectiveness of the state-of-the-art sparse reconstruction algorithms described in Section 5. The code used to run these experiments is available at http://www.acm.caltech.edu/~nesta. We have modified this NESTA experiment infrastructure to include PARNES and C-SALSA, and we repeat some of the tests in [5] using the same experimental standards and parameters. Refer to the [5] for a detailed description of the experiments.

One difficulty that arises in carrying out such broad experiments is that some of the algorithms solve QP(λ) whereas others solve BP(σ). Comparing the algorithms thus requires a way of finding a (σ, λ) pair for which the solutions of QP(λ) and BP(σ) coincide. The NESTA experiments utilize a two-step procedure. Given the noise level ϵ, the authors choose $\sigma_0 := \sqrt{m + 2\sqrt{2m}\epsilon}$, and then use SPGL1 to solve the corresponding BP(σ_0) problem. The SPGL1 dual solution then provides an estimate of the corresponding λ. In practice, the computation of λ is not very stable, and so a second step is performed in which FISTA is used to compute a σ corresponding to λ using a very high accuracy of around 10^{-14}.

The highly accurate solution computed by FISTA is used to determine the accuracy of the solutions computed by the other solvers. Section 4.2 of [5] shows that this is reasonable since FISTA gives very accurate solutions provided that enough iterations

are taken. For each test, FISTA is ran twice. In the first run, FISTA is ran with no limit on the number of iterations until the relative change in the function value is less than 10^{-14}. This solution is used to determine the accuracy of the computed solutions. The results recorded for FISTA are from running FISTA a second time with either stopping criterion (19) or (20).

Since the different algorithms utilize different stopping criteria, to maintain fairness, the codes have been modified to allow for two new stopping criteria. Intuitively, the algorithms are run until they achieve a solution at least as accurate as the one obtained by NESTA. In [5], NESTA (with continuation) is used to compute a solution x_{NES}. Let $\hat{x}_k$ be the k-th iteration in the algorithm being tested. The stopping criteria used are:

$$\|\hat{x}_k\|_{\ell_1} \leq \|x_{\mathrm{NES}}\|_{\ell_1} \quad \text{and} \quad \|b - A\hat{x}_k\|_{\ell_2} \leq 1.05\,\|b - Ax_{\mathrm{NES}}\|_{\ell_2}, \tag{19}$$

and

$$\lambda\|\hat{x}_k\|_{\ell_1} + \frac{1}{2}\|A\hat{x}_k - b\|_{\ell_2}^2 \leq \lambda\|x_{\mathrm{NES}}\|_{\ell_1} + \frac{1}{2}\|Ax_{\mathrm{NES}} - b\|_{\ell_2}^2. \tag{20}$$

The rationale for having two stopping criteria is to reduce any potential bias arising from the fact that some algorithms solve QP(λ), for which (20) is the most natural, while others solve BP(σ), for which (19) is the most natural. It is evident from the tables below that there is not a significant difference between using (19) and (20). For each test, the number of calls to A and $A^\top$ (N_A) is recorded, and the algorithms are said to have not converged (DNC) if the number of calls exceeds 20,000.

In Tables 3 and 4, we repeat the experiments done in Tables 5.1 and 5.2 of [5]. These experiments involve recovering an unknown, exactly s-sparse signal with $n = 262{,}144$, $m = n/8$, and $s = m/5$. For each run, the measurement operator A is a randomly subsampled discrete cosine transform, and the noise level is set to 0.1. The experiments are performed with increasing values of the dynamic range d where $d = 20, 40, 60, 80, 100$ dB.

The dynamic range d is a measure of the ratio between the largest and smallest magnitudes of the non-zero coefficients of the unknown signal. Problems with a high dynamic range occur often in applications. In these cases, high accuracy becomes important since one must be able to detect and recover low-power signals with small amplitudes which may be obscured by high-power signals with large amplitudes.

Table 2 compares the accuracy of the different solvers when used to calculate the results in the last column of Table 3. As this corresponds to a very high dynamic range (100 dB), one hopes to obtain very accurate results. Although FISTA produces the most accurate results ($\|x - x^*\|_1/\|x^*\|_1 = 3.63 \cdot 10^{-4}$), with at least twice the accuracy of the other solvers, it requires the over 10,000 calls to A and $A^\top$. In contrast, PARNES only requires 632 function calls to reach a relative accuracy of $\|x - x^*\|_1/\|x^*\|_1 = 6.93 \cdot 10^{-4}$. The solvers FPC-AS and FPC-AS (CG) do well and only require around 300 iterations to reach a relative accuracy of around $6.93 \cdot 10^{-4}$. The remaining algorithms reach relative accuracies of around $8 \cdot 10^{-4}$ or more, and GSPR does not converge. Without continuation, NESTA only achieves a relative accuracy of $4.12 \cdot 10^{-3}$ after 15,227 function calls. However, NESTA with continuation does much better and reaches a relative accuracy of $8.12 \cdot 10^{-4}$ after 787 function calls.

Table 2 Comparison of accuracy using experiments from Table 3

Methods	N_A	$\|x\|_1$	$\|Ax-b\|_2$	$\frac{\|x-x^*\|_1}{\|x^*\|_1}$	$\|x-x^*\|_\infty$	$\|x-x^*\|_2$
PARNES	632	942,197.606	2.692	0.000693	8.312	46.623
NESTA	15,227	942,402.960	2.661	0.004124	45.753	255.778
NESTA + CT	787	942,211.581	2.661	0.000812	9.317	52.729
GPSR	DNC	DNC	DNC	DNC	DNC	DNC
GPSR + CT	11,737	942,211.377	2.725	0.001420	15.646	90.493
SPARSA	693	942,197.785	2.728	0.000783	9.094	51.839
SPGL1	504	942,211.520	2.628	0.001326	14.806	84.560
FISTA	12,462	942,211.540	2.654	0.000363	4.358	26.014
FPC-AS	287	942,210.925	2.498	0.000672	9.374	45.071
FPC-AS (CG)	361	942,210.512	2.508	0.000671	9.361	45.010
FPC	9,614	942,211.540	2.719	0.001422	15.752	90.665
FPC-BB	1,082	942,209.854	2.726	0.001378	15.271	87.963
BREGMAN-BB	1,408	942,286.656	1.326	0.000891	9.303	52.449
C-SALSA	1,338	942,219.455	2.317	0.000851	9.541	55.14

Dynamic range 100 dB, $\sigma = 0.100$, $\mu = 0.020$, sparsity level $s = m/5$. Stopping rule is (19)

In Tables 3 and 4, the same experiment is ran for the two stopping criteria. Since there is no notable difference between the two sets of results, we only analyze Table 3. Here, FPC-AS and FPC-AS (CG) perform the best for large values of d, and the number of function calls ranges from 200 to 375 for all values of the dynamic range. In these cases, we see a relatively small increase in N_A as d increases from 20 dB to 100 dB. Our method, PARNES, and SPGL1 generally perform well and do particularly well for

Table 3 Number of function calls where the sparsity level is $s = m/5$ and the stopping rule is (19)

Method	20 dB	40 dB	60 dB	80 dB	100 dB
PARNES	122	172	214	470	632
NESTA	383	809	1,639	4,341	15,227
NESTA + CT	483	513	583	685	787
GPSR	64	622	5,030	DNC	DNC
GPSR + CT	271	219	357	1,219	11,737
SPARSA	323	387	465	541	693
SPGL1	58	102	191	374	504
FISTA	69	267	1,020	3,465	12,462
FPC-AS	209	231	299	371	287
FPC-AS (CG)	253	289	375	481	361
FPC	474	386	478	1,068	9,614
FPC-BB	164	168	206	278	1,082
BREGMAN-BB	211	223	309	455	1,408
C-SALSA	242	602	702	970	1,338

Table 4 Number of function calls where the sparsity level is $s = m/5$ and the stopping rule is (20)

Method	20 dB	40 dB	60 dB	80 dB	100 dB
PARNES	74	116	166	364	562
NESTA	383	809	1,639	4,341	15,227
NESTA + CT	483	513	583	685	787
GPSR	62	618	5,026	DNC	DNC
GPSR + CT	271	219	369	1,237	11,775
SPARSA	323	387	463	541	689
SPGL1	43	99	185	365	488
FISTA	72	261	1,002	3,477	12,462
FPC-AS	115	167	159	371	281
FPC-AS (CG)	142	210	198	481	355
FPC	472	386	466	1,144	9,734
FPC-BB	164	164	202	276	1,092
BREGMAN-BB	211	223	309	455	1,408
C-SALSA	202	550	650	898	1,230

small values of d. However, both exhibit a larger increase in N_A with d, with PARNES increasing from 122 to 632 function calls and SPGL1 ranging between 58 and 504. The solvers NESTA + CT and SPARSA perform relatively well for large values of d with N_A ranging between 500 and 800.

In applications, the signal to be recovered is often approximately sparse rather than exactly sparse. Again, high accuracy is important when solving these problems. The last two tables, Tables 5 and 6, replicate Tables 5.3 and 5.4 of [5]. Each run involves an approximately sparse signal obtained from a permutation of the Haar

Table 5 Recovery results of an approximately sparse signal (with Gaussian noise of variance 1 added) and with (20) as a stopping rule

Method	Run 1	Run 2	Run 3	Run 4	Run 5
PARNES	838	810	1,038	1,098	654
NESTA	8,817	10,867	9,887	9,093	11,211
NESTA + CT	3,807	3,045	3,047	3,225	2,735
GPSR	DNC	DNC	DNC	DNC	DNC
GPSR + CT	DNC	DNC	DNC	DNC	DNC
SPARSA	2,143	2,353	1,977	1,613	DNC
SPGL1	916	892	1,115	1,437	938
FISTA	3,375	2,940	2,748	2,538	3,855
FPC-AS	DNC	DNC	DNC	DNC	DNC
FPC-AS (CG)	DNC	DNC	DNC	DNC	DNC
FPC	DNC	DNC	DNC	DNC	DNC
FPC-BB	5,614	7,906	5,986	4,652	6,906
BREGMAN-BB	3,288	1,281	1,507	2,892	3,104
C-SALSA	742	626	630	1,226	826

Table 6 Recovery results of an approximately sparse signal (with Gaussian noise of variance 0.1 added) and with (20) as a stopping rule

Method	Run 1	Run 2	Run 3	Run 4	Run 5
PARNES	1,420	1,772	1,246	1,008	978
NESTA	11,573	10,457	10,705	8,807	13,795
NESTA + CT	7,543	13,655	11,515	3,123	2,777
GPSR	DNC	DNC	DNC	DNC	DNC
GPSR + CT	DNC	DNC	DNC	DNC	DNC
SPARSA	12,509	DNC	DNC	3,117	DNC
SPGL1	1,652	1,955	2,151	1,311	2,365
FISTA	10,845	12,165	10,050	7,647	11,997
FPC-AS	DNC	DNC	DNC	DNC	DNC
FPC-AS (CG)	DNC	DNC	DNC	DNC	DNC
FPC	DNC	DNC	DNC	DNC	DNC
FPC-BB	DNC	DNC	DNC	DNC	DNC
BREGMAN-BB	3,900	3,684	2,045	3,292	3,486
C-SALSA	1,886	1,926	1,770	1,754	1,854

wavelet coefficients of a 512×512 image. The measurement vector b consists of $m = n/8 = 512^2/8 = 32{,}768$ random discrete cosine measurements, and the noise level is set to have a variance of 1 in Table 5 and 0.1 in Table 6. For more specific details, refer to [5].

We have seen that NESTA + CT, SPARSA, SPGL1, PARNES, and both versions of FPC-AS perform well in the case of exactly sparse signals for all values of the dynamic range. However, in the case of approximately sparse signals, SPARSA and all versions of FPC no longer converge in under 20,000 function calls. In Table 5, PARNES, SPGL1, and C-SALSA perform well, with PARNES and C-SALSA taking around 650 function calls for some runs (compare to NESTA + CT which takes at least 3,000 iterations). These algorithms also perform the best in Table 6, and most other algorithms no longer converge in under 10,000 function calls.

6.1 Choice of parameters

As Tseng observed, accelerated proximal gradient algorithms will converge so long as the condition given as (45) in [36] is satisfied. In our case this translates into

$$\min_{x\in\mathbb{R}^n} \left\{ \nabla f(y_k)^\top x + \frac{L}{2}\|x - x_k\|_2^2 + P(x) \right\} \geq \nabla f(y_k)^\top y_k + P(y_k), \tag{21}$$

upon setting $\gamma_k = 1$ and

$$P(x) = \begin{cases} 0 & \text{if } \|x\|_1 \leq \tau, \\ \infty & \text{otherwise,} \end{cases}$$

in (45) in [36]. In other words, the value of L need not necessarily be fixed at the Lipschitz constant of ∇f but may be decreased, and decreasing L has the same effect

as increasing the stepsize. Tseng suggests to decrease L adaptively by a constant factor until (45) is violated, then backtrack and repeat the iteration (cf. Note 6 in [36]). For simplicity, and very likely at the expense of speed, we do not change our L adaptively in PARNES and NESTA-LASSO. Instead, we choose a small fixed L by trying a few different values so that (21) is satisfied for all k and likewise for the tolerance η in Algorithm 3. However, even with this crude way of selecting L and η, the results obtained are still rather encouraging.

7 Conclusions

As seen in the numerical results, SPGL1 and NESTA are among some of the top performing solvers available for basis pursuit denoising problems. We have therefore made use of Nesterov's accelerated proximal gradient method in our algorithm NESTA-LASSO and shown that updating the prox-center leads to improved results. Through our experiments, we have shown that using NESTA-LASSO in the Pareto root-finding method leads to results comparable to those of currently available state-of-the-art methods. Moreover, PARNES performs consistently well in all our experiments.

Acknowledgements We would like to give special thanks to Emmanuel Candès for helpful discussions and ideas. The numerical experiments in this paper rely on the shell scripts and MATLAB codes[1] of Jérôme Bobin. We have also benefited from Michael Friedlander and Ewout van den Berg's MATLAB codes[2] for SPGL1. We are grateful to them for generously making their codes available on the web. The work of Lek-Heng Lim is partially supported by NSF Grants DMS 1209136 and DMS 1057064.

References

1. Afonso, M., Bioucas-Dias, J., Figueiredo, M.: Fast image recovery using variable splitting and constrained optimization. IEEE Trans. Image Process. **19**(9), 2345–2356 (2010)
2. Afonso, M., Bioucas-Dias, J., Figueiredo, M.: Fast frame-based image deconvolution using variable splitting and constrained optimization. In: IEEE/SP 15th Workshop on Statistical Signal Processing, 2009. SSP '09, pp. 109–112 (2009)
3. Barzilai, J., Borwein, J.: Two point step size gradient method. IMA J. Numer. Anal. **8**(1), 141–148 (1988)
4. Beck, A., Teboulle, M.: Fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imag. Sci. **2**(1), 183–202 (2009)
5. Becker, S., Bobin, J., Candès, E.J.: NESTA: a fast and accurate first-order method for sparse recovery. SIAM J. Imag. Sci. **4**(1), 1–39 (2011)
6. Bertsekas, D.P.: Nonlinear Programming. Belmont, MA (1999)
7. Birgin, E., Martínez, J., Raydan, M.: Nonmonotone spectral projected-gradient methods on convex sets. SIAM J. Optim. **10**(4), 1196–1211 (2000)
8. Bobin, J., Stark, J.-L., Ottensamer, R.: Compressed sensing in astronomy. IEEE J. Sel. Top. Signal Process. **2**(5), 718–726 (2008)
9. Candès, E.J.: The restricted isometry property and its implications for compressed sensing. C. R. Math. Acad. Sci. Paris **346**(9–10), 589–592 (2008)

[1] http://www.acm.caltech.edu/~nesta/NESTA_ExperimentPackage.zip
[2] http://www.cs.ubc.ca/labs/scl/spgl1

10. Candès, E.J., Tao, T.: The Dantzig selector: statistical estimation when p is much larger than n. Ann. Stat. **35**(6), 2313–2351 (2007)
11. Candès, E.J., Romberg, J., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Commun. Pure Appl. Math. **59**(8), 1207–1223 (2006)
12. Chen, S., Donoho, D.L., Saunders, M.: Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. **20**(1), 33–61 (1998)
13. Dembo, R.S., Eisenstat, S.C., Steihaug, T.: Inexact newton methods. SIAM J. Numer. Anal. **19**(2), 400–408 (1982)
14. Donoho, D.L.: For most large underdetermined systems of linear equations the ℓ_1-norm solution is also the sparsest solution. Commun. Pure Appl. Math. **59**(6), 797–829 (2006)
15. Duchi, J., Shalev-Shwartz, S., Singer, Y., Chandra, T.: Efficient projections onto the ℓ_1-ball for learning. Proc. Int. Conf. Mach. Learn. (ICML '08) **25**(307), 272–279 (2008)
16. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Ann. Stat. **32**(2), 407–499 (2004)
17. Figueiredo, M., Nowak, R., Wright, S.: Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. IEEE J. Sel. Top. Signal Process. **1**(4), 586–597 (2007)
18. Figueiredo, M., Nowak, R., Wright, S.: Sparse reconstruction by separable approximation. IEEE Trans. Signal Process. **577**, 2479–2493 (2009)
19. Friedman, J., Hastie, T., Höfling, H., Tibshirani, R.: Pathwise coordinate optimization. Ann. Appl. Stat. **1**(2), 302–332 (2007)
20. Fuchs, J.J.: On sparse representations in arbitrary redundant bases. IEEE Trans. Inf. Theory **50**(6), 1344 (2004)
21. Garey, M.R., Johnson, D.S.: Computers and Intractability. A Guide to the Theory of NP-Completeness. W.H. Freeman, New York (1979)
22. Hale, E.T., Yin, W., Zhang, Y.: A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing. Rice University Technical Report (2007)
23. Hale, E.T., Yin, W., Zhang, Y.: Fixed-point continuation for ℓ_1-minimization: methodology and convergence. SIAM J. Optim. **19**(3), 1107–1130 (2008)
24. Hennenfent, G., Herrmann, F.J.: Sparseness-constrained data continuation with frames: applications to missing traces and aliased signals in 2/3-D. SEG Tech. Program Expanded Abstracts **24**(1), 2162–2165 (2005)
25. Hennenfent, G., Herrmann, F.J.: Simply denoise: wavefield reconstruction via jittered undersampling. Geophysics **73**(3), V19–V28 (2008)
26. Natarajan, B.K.: Sparse approximate solutions to linear systems. SIAM J. Comput. **24**(2), 227–234 (1995)
27. Nesterov, Y.: A method for solving the convex programming problem with convergence rate $O(1/k^2)$. Dokl. Akad. Nauk SSSR **269**(3), 543–547 (1983)
28. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. **103**(1), 127–152 (2005)
29. Nesterov, Y.: Gradient methods for minimizing composite objective function. CORE Discussion Paper 2007/76 (2007)
30. Osborne, M.R., Presnell, B., Turlach, B.A.: On the LASSO and its dual. J. Comput. Graph. Stat. **9**(2), 319–337 (2000)
31. Osborne, M.R., Presnell, B., Turlach, B.A.: A new approach to variable selection in least squares problems. IMA J. Numer. Anal. **20**(3), 389–403 (2000)
32. Ponomarev, S.P.: Submersions and preimages of sets of measure zero. Sib. Math. J. **28**(1), 153–163 (1987)
33. Romberg, J.: Imaging via compressive sensing. IEEE Trans. Signal Process. **25**(2), 14–20 (2008)
34. Tibshirani, R.: Regression shrinkage and selection via the LASSO. J. R. Stat. Soc., Ser. B Stat. Methodol. **58**(1), 267–288 (1996)
35. Tropp, J.A.: Just relax: convex programming methods for identifying sparse signals in noise. IEEE Trans. Inf. Theory **52**(3), 1030–1051 (2006)
36. Tseng, P.: On accelerated proximal gradient methods for convex-concave optimization. Preprint (2008)
37. van den Berg, E., Friedlander, M.P.: Probing the Pareto frontier for basis pursuit solutions. SIAM J. Sci. Comput. **31**(2), 890–912 (2008/09)

38. van den Berg, E., Friedlander, M.P., Hennenfent, G., Herrmann, F.J., Saab, R., Yilmaz, Ö.: Algorithm 890: SPARCO: a testing framework for sparse reconstruction. ACM Trans. Math. Softw. **35**(4), 16 (2009)
39. Wen, Z., Yin, W., Goldfarb, D., Zhang, Y.: A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization and continuation. SIAM J. Sci. Comput. **32**(4), 1832 (2010)
40. Yin, W., Osher, S., Goldfarb, D., Darbon, J.: Bregman iterative algorithms for l_1 minimization with applications to compressed sensing. SIAM J. Imag. Sci. **1**(1), 143–168 (2008)
41. Yu, Y.L.: Nesterov's optimal gradient method. LLL, Jul. 30 2009 (2009). http://webdocs.cs.ualberta.ca/~yaoliang/Non-smooth%20Optimization.pdf