Preprint manuscript No.
(will be inserted by the editor)

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

Ming Gu · Lek-Heng Lim · Cinna Julie Wu

Received: date / Accepted: date

Abstract In this article we propose an algorithm, NESTA-LASSO, for the LASSO problem (i.e., an underdetermined linear least-squares problem with a one-norm constraint on the solution) that exhibits linear convergence under the *restricted isometry property* (RIP) and some other reasonable assumptions. Inspired by the state-of-the-art sparse recovery method, NESTA, we rely on an accelerated proximal gradient method proposed by Nesterov in 1983 that 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, resulting in the linear convergence of NESTA-LASSO under reasonable assumptions.

Our work is motivated by recent advances in solving the basis pursuit denoising (BPDN) problem (i.e., approximating the minimum one-norm solution to an underdetermined least squares problem). Thus, one of our goals is to show that NESTA-LASSO can be used to solve the BPDN problem. We use NESTA-LASSO to solve a subproblem within the Pareto root-finding method used by the state-of-the-art BPDN solver SPGL1. The resulting algorithm is called PARNES, and we show, experimentally, that it is comparable to currently available solvers.

Keywords basis pursuit · Newton's method · Pareto curve · Nesterov's method · compressed sensing · convex minimization · duality

1 Introduction

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

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

M. Gu
Department of Mathematics, University of California at Berkeley, Berkeley, CA 94720-3840, USA
Tel.: 510-642-3145
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

C. J. Wu
Department of Mathematics, University of California at Berkeley, Berkeley, CA 94720-3840, USA
E-mail: cinnawu@math.berkeley.edu

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 [29], seismics [22, 21], astronomy [6], and model selection in regression [15]. Since (1) is known to be ill-posed and NP-hard [18, 23], various convex, l_1-relaxed formulations are often used.

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

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

The special case of $\sigma = 0$ is the basis pursuit problem [12]. Two other commonly used l_1-relaxations are the LASSO problem [30]

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

and the penalized least-squares problem

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

proposed by Chen, Donoho, and Saunders [12]. A large amount of work has been done to show that these formulations give an effective approximation of the solution to (1); see [13, 31, 10]. In particular, under certain conditions on the sparsity of the solution x, x can be recovered exactly provided that A satisfies the *restricted isometry property* (RIP).

There are 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 [24], 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 [25]. 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 [33]. Although the exact dependence is usually hard to compute [33], there are solution methods which exploit these relationships. The Matlab solver SPGL1 is based on the Pareto root-finding method [33] 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. Our algorithm, NESTA-LASSO (cf. Algorithm 2), effectively uses Nesterov's work to solve the LASSO problem. Additionally, we propose a modification of Nesterov's work which results in the local linear convergence of NESTA-LASSO under reasonable assumptions. 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 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 can exhibit linear convergence. 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 2), is an application of the accelerated proximal gradient algorithm of Nesterov [24] 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, Shalev-Shwartz, Singer, and Chandra [14] 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.

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

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:

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

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

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

then the iterates defined above satisfy

$$f(y_k) - f(x^*) \leq \frac{L}{k(k+1)}\|x^* - c\|_2^2 = O\left(\frac{L}{k^2}\right). \tag{6}$$

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.

In [25], 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. We are motivated by the accuracy and speed of NESTA, and the fact that smooth version of Nesterov's algorithm has a faster convergence rate than the nonsmooth version.

Algorithm 1 Accelerated proximal gradient method for convex minimization

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

```
initialize x_0;
for k = 0, 1, 2, ..., do
    compute f(x_k) and ∇f(x_k);
    y_k = argmin_{y∈Q} ∇f(x_k)^T (y − x_k) + (L/2)‖y − x_k‖_2^2;
    z_k = argmin_{z∈Q} Σ_{i=0}^{k} ((i+1)/2)[f(x_i) + ∇f(x_i)^T (z − x_i)] + (L/2)‖z − c‖_2^2;
    x_k = (2/(k+3)) z_k + ((k+1)/(k+3)) y_k;
end for
```

2.2 NESTA-LASSO: An accelerated proximal gradient algorithm for LASSO

We apply Nesterov's accelerated proximal gradient method in Algorithm 1 to the LASSO problem LS(τ), and in each iteration, we use the fast l^1-projector proj_1 described in the next section. The pseudocode for this is given in Algorithm 2. We make one slight improvement to Algorithm 1, namely, we update our prox-centers from time to time. In fact, we will see in Section 3 that the prox-centers may be updated in an optimal fashion and that this leads to linear convergence under a suitable application of RIP (see Corollary 1 for details).

In our case, $f = \frac{1}{2}\|b - Ax\|_2^2$, $\nabla f = A^\top(Ax - b)$, and Q is the one-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

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

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

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

2.3 One Projector

The one-projector, proj_1, is used twice in each iteration of Algorithm 2. We briefly describe the algorithm of Duchi, Shalev-Schwartz, Singer, and Chandra [14] for fast projection to an l^1-ball in high-dimension. A similar algorithm is presented in [33]. The algorithm costs $O(n \log n)$ in the worst case, but it has been shown to cost much less experimentally [33]. Note that the two calls to one-projector in each iteration can be reduced to one call with results in Tseng's paper [32]. However, we make no change to our algorithm due to the low cost of the one-projector.

Consider the projection of an n-vector c onto the one-norm ball $\|x\|_1 \le \tau$. This is given by the minimization problem

$$\text{proj}_1(c, \tau) := \operatorname*{argmin}_{x} \|c - x\|_2 \quad \text{s.t.} \quad \|x\|_1 \le \tau. \tag{7}$$

Algorithm 2 NESTA-LASSO algorithm with prox-center updates

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

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

Without loss of generality, the symmetry of the one-norm ball allows us to assume that c has all nonnegative entries. Assuming the coefficients of the vector c are ordered from largest to smallest, the solution $x^* = \operatorname{proj}_1(c, \tau)$ is given by

$$x_i^* = \max\{0, c_i - \eta\} \quad \text{with} \quad \eta = \frac{\tau - (c_1 + \cdots + c_k)}{k} \tag{8}$$

where k is the largest index such that $\eta \le c_k$. Please refer to [33,14] for more details on the implementation.

3 Optimality

In NESTA-LASSO, we update the prox-center, c, in a provably optimal way which yields a local linear convergence rate. We analyze the case where the solution x^* is s-sparse. The case where x^* is approximately sparse has not been analyzed; we plan to do this for future work. Nesterov has shown that when his accelerated proximal gradient algorithm is applied to minimize the objective function f, the kth iterate y_k and the minimizer x^* satisfy

$$f(y_k) - f(x^*) \le \frac{L}{k(k+1)} \|x^* - c\|_2^2 \tag{9}$$

where L is the Lipschitz constant for ∇f and c is the prox-center [24,25].

In our case, $f(x) = \|Ax - b\|_2^2$ where the underdetermined matrix A is understood to represent a dictionary of atoms, usually sampled via some measurements. We will assume that A satisfies the *restricted isometry property* (RIP) of order $2s$ as described in [8,9]. Namely, there exists a constant $\delta_{2s} \in (0,1)$ such that

$$(1 - \delta_{2s})\|x\|_2^2 \le \|Ax\|_2^2 \le (1 + \delta_{2s})\|x\|_2^2 \tag{10}$$

whenever $\|x\|_0 \le 2s$. 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.

Additionally, we assume the iterates y_k are s-sparse when sufficiently close to x^*. This is reasonable since each iteration of NESTA-LASSO involves projecting the current iteration onto a 1-norm ball. Due to the geometry of the projection, there is a high likelihood that our assumption will hold. Under the RIP and the assumption that x^* is s-sparse, it follows by Theorem 3.2 in [26] that the LASSO problem has a unique solution. Since the one-norm ball is compact, this implies that y_k converges to x^*.

Let $\delta = 1 - \delta_{2s}$. We have

$$\|A(x^* - y_k)\|_2^2 + 2(y_k - x^*)^\top A^\top (Ax^* - b) = f(y_k) - f(x^*) \geq \|A(y_k - x^*)\|_2^2 \geq \delta \|y_k - x^*\|_2^2.$$

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

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

Thus, $(y_k - x^*)^\top A^\top (Ax^* - b) \geq 0$. The second inequality follows from (10). Then from (9), we have

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

Putting everything together gives

$$\|y_k - x^*\|_2 \leq \sqrt{\frac{L}{k(k+1)\delta}} \|x^* - c\|_2 \leq \frac{1}{k} \sqrt{\frac{L}{\delta}} \|c - x^*\|_2. \tag{11}$$

The above relation and (9) suggest that we can speed up Algorithm 1 by updating the prox-center, c, with the current iterate y_k every k steps. With our assumptions, we prove in the following theorem that there is an optimal number of such steps. 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 1 *Suppose A satisfies the restricted isometry property of order $2s$, the solution x^* is s-sparse, and the iterates y_{jk} are eventually s-sparse. Then for each j,*

$$\|y_{jk_{\text{opt}}} - x^*\| \leq 1/e \|y_{j1} - x^*\|$$

where

$$k_{\text{opt}} = e\sqrt{\frac{L}{\delta}} \tag{12}$$

and e is the base of the natural logarithm. Moreover, the total number of iterations, $j_{\text{tot}} k_{\text{opt}}$, to get $\|y_{jk} - x^*\|_2 \leq \varepsilon$ is minimized with this choice of k_{opt}.

Proof First observe that (11) implies

$$\|y_{jk} - x^*\|_2 \leq \frac{1}{k}\sqrt{\frac{L}{\delta}} \|y_{j1} - x^*\|_2 \leq \left(\frac{1}{k}\sqrt{\frac{L}{\delta}}\right)^j \|y_{11} - x^*\|_2 \leq \varepsilon \|y_{11} - x^*\|_2$$

when

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

Table 1 Number of products, N_A, with A and A^* for NESTA-LASSO without prox-center updates

Number of Rows of A	Number of Columns of A	τ	N_A
100	256	6.28	69
200	512	12.6	77
400	1024	25.1	157

Table 2 Number of products, N_A, with A and A^* for NESTA-LASSO with prox-center updates

Number of Rows of A	Number of Columns of A	τ	N_A
100	256	6.28	37
200	512	12.6	47
400	1024	25.1	45

This relation allows us to choose k to minimize the product jk. Since

$$jk = \frac{k \log \varepsilon}{\log \sqrt{L/\delta} - \log k},$$

taking derivative of the expression on the right shows that jk is minimized when

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

where e is the base of the natural logarithm. Thus,

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

The assumption that the iterates y_{jk} will eventually be s-sparse is reasonable precisely because we expect the LASSO relaxation (3) to be a good proxy for (1). In other words, we expect our solutions to (3) to be sparse. Of course this argument is merely meant to be a heuristic; a more rigorous justification along the lines of [13] may be possible and is in our future plans.

Let k_{opt} be as in (12). For each prox-center change, j, perform k_{opt} inner iterations, and let $p_j = y_{jk_{\text{opt}}}$ be the output to the jth outer iteration. An immediate consequence of Theorem 1 is that the relative decrease after k_{opt} steps of the inner iteration in Algorithm 2 is e^{-2}, i.e.

$$\|p_j - x^*\|_2^2 \leq e^{-2} \|p_{j-1} - x^*\|_2^2$$

and in general, this is the best possible.

Corollary 1 *If A satisfies the restricted isometry property of order $2s$, the solution x^* is s-sparse, and the iterates p_j are eventually s-sparse, then Algorithm 2 is linearly convergent.*

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 Tables 1 and 2, 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 [33] 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 [11] for more information on the inexact Newton method. In Section 6, we show experimentally that using NESTA-LASSO in place of the SPG approach can lead to improved results. We call this version of the Pareto root-finding method, PARNES. The pseudocode of PARNES is given in Algorithm 3.

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 [33] for a more detailed explanation of these properties of the Pareto curve.

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, proved in [33], shows that the Pareto curve is convex, strictly decreasing over the interval $\tau \in [0, \tau_{\text{BP}}]$, and continuously differentiable for $\tau \in (0, \tau_{\text{BP}})$.

Theorem 2 *The function φ is*

1. *convex and nonincreasing.*
2. *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 LS(τ) and $y_\tau = r_\tau/\|r_\tau\|_2$ with $r_\tau = Ax_\tau - b$.*
3. *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 BP(σ) and LS(τ), solving BP(σ) 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 LS(τ) for a sequence of parameters $\tau_k \to \tau_\sigma$ where τ_σ is the optimal objective value of BP(σ).

Applying Newton's method to φ gives

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

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 [4]). By Theorem 2, $\varphi(\tau_k)$ is the optimal value to LS(τ_k) and $\varphi'(\tau_k)$ is the dual solution to LS(τ_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 $\varphi'(\tau_k)$.

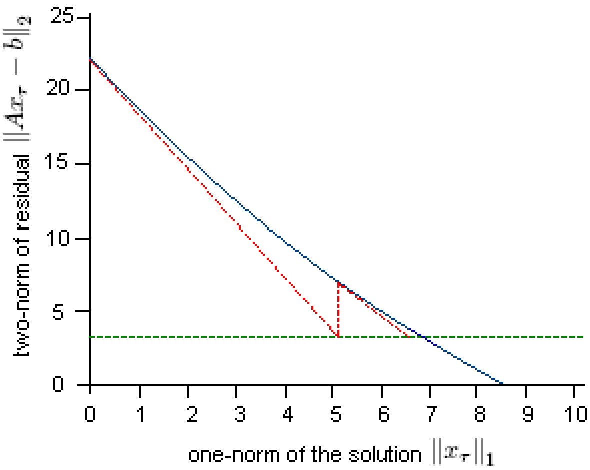

Fig. 1 An example of a Pareto curve. The solid line is the Pareto curve; the dotted red lines give two iterations of Newton's method.

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

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

The authors of [33] have proved the following convergence result.

Theorem 3 *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. The method follows the algorithm by Birgin, Martínez, and Raydan [5] and is shown to be globally convergent. The costs include evaluating Ax, $A^\top r$, and a projection onto the one-norm ball $\|x\|_1 \leq \tau$. In PARNES, we replace this SPG algorithm with our algorithm, NESTA-LASSO (cf. Algorithm 2).

5 Other solution techniques and tools

In the NESTA paper, [3], extensive experiments are carried out, comparing the effectiveness of state-of-the-art sparse reconstruction algorithms. 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 in its tests and repeated some of the tests in [3] with the same experimental standards and parameters. Refer to the [3] for a detailed description of the experiments. The algorithms tested and their experimental details are described below. Note that the algorithms either solve BP(σ) or QP(λ).

Algorithm 3 PARNES: Pareto curve method with NESTA-LASSO

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

$\tau_0 = 0$, $\varphi_0 = \|b\|_2$, $\varphi'_0 = \|A^\top b\|_\infty$;
for $k = 0, \dots, k_{\max}$, **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

5.1 NESTA [3]

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

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

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

Continuation techniques are used to speed up NESTA in [3]. Such techniques are useful when it is observed that a problem involving some parameter λ is faster for large λ, [27, 19]. 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 [16]

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 [17]

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 [1] 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 [33] and SPARCO [34]

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 [34]. 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 [2]

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 [3], 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 [3] 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 [19,20]

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 [35]

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 [19] 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 [36]

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 SALSA [38,37]

There is a new state-of-the-art method solving BP(σ) called SALSA which has been shown to be competitive with SPGL1 and NESTA. The experimental results in our paper do not include SALSA, but we hope to make comparisons in future work. The code for SALSA can be found at `http://cascais.lx.it.pt/~mafonso/salsa.html`.

6 Numerical results

As mentioned above, some of the algorithms we test solve QP(λ) and some solve BP(σ). Comparing the algorithms thus requires a way of finding a (σ, λ) pair which for which the solutions coincide. The tests in [3] use a two-step procedure. From the noise level, σ is chosen, and then SPGL1 is used to solve BP(σ). The SPGL1 dual solution gives an estimate of the corresponding λ and then FISTA is used to compute a second σ corresponding to this λ with high accuracy.

In the NESTA experiments, FISTA is used to determine the accuracy of the computed solutions while NESTA is used to compute a solution that is used in the stopping criteria. Please refer to [3] for a complete description of the experimental details. Section 4.2 of [3], shows that FISTA gives very accurate solutions provided enough iterations are taken. For each test, FISTA is run twice. FISTA is first run, 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. FISTA is ran a second time with one of the following stopping criterion; the results of this run are recorded in the tables.

In [3], NESTA (with continuation), is used to compute a solution x_{NES}. In the tests, the following stopping criteria are used where $\hat{x}_k$ is the kth-iteration in the algorithm being tested.

$$\|\hat{x}_k\|_{\ell_1} \leq \|x_{\text{NES}}\|_{\ell_1} \quad \text{and} \quad \|b - A\hat{x}_k\|_{\ell_2} \leq 1.05\,\|b - Ax_{\text{NES}}\|_{\ell_2}, \tag{13}$$

or

$$\lambda\|\hat{x}_k\|_{\ell_1} + \frac{1}{2}\|A\hat{x}_k - b\|_{\ell_2}^2 \leq \lambda\|x_{\text{NES}}\|_{\ell_1} + \frac{1}{2}\|Ax_{\text{NES}} - b\|_{\ell_2}^2. \tag{14}$$

In other words, the algorithms are run until they achieve a solution at least as accurate as NESTA.

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 (14) is the most natural, while others solve

Table 3 Comparison of accuracy using experiments from Table 4. Dynamic range 100 dB, $\sigma = 0.100$, $\mu = 0.020$, sparsity level $s = m/5$. Stopping rule is (13).

Methods	N_A	$\|x\|_1$	$\|Ax - b\|_2$	$\frac{\|x-x^*\|_1}{\|x^*\|_1}$	$\|x - x^*\|_\infty$	$\|x - x^*\|_2$
PARNES	632	942197.606	2.692	0.000693	8.312	46.623
NESTA	15227	942402.960	2.661	0.004124	45.753	255.778
NESTA + CT	787	942211.581	2.661	0.000812	9.317	52.729
GPSR	DNC	DNC	DNC	DNC	DNC	DNC
GPSR + CT	11737	942211.377	2.725	0.001420	15.646	90.493
SPARSA	693	942197.785	2.728	0.000783	9.094	51.839
SPGL1	504	942211.520	2.628	0.001326	14.806	84.560
FISTA	12462	942211.540	2.654	0.000363	4.358	26.014
FPC-AS	287	942210.925	2.498	0.000672	9.374	45.071
FPC-AS (CG)	361	942210.512	2.508	0.000671	9.361	45.010
FPC	9614	942211.540	2.719	0.001422	15.752	90.665
FPC-BB	1082	942209.854	2.726	0.001378	15.271	87.963
BREGMAN-BB	1408	942286.656	1.326	0.000891	9.303	52.449

BP(σ), for which (13) is the most natural. It is evident from the tables below that there is not a significant difference whether (13) or (14) is used. The algorithms are recorded to not have converged (DNC) if the number of calls to A or A^* exceeds 20,000.

In Tables 4 and 5, we repeat the experiments done in Tables 5.1 and 5.2 of [3]. These experiments involve recovering an unknown signal that is exactly s-sparse with $n = 262144$, $m = n/8$, and $s = m/5$. The experiments are performed with increasing values of the dynamic range d where $d = 20, 40, 60, 80, 100$ dB. For each run, the measurement operator is a randomly subsampled discrete cosine transform, and the noise level is set to 0.1.

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 lower-power signals with small amplitudes which may be obscured by high-power signals with large amplitudes.

The last two tables, Tables 6 and 7, replicate Tables 5.3 and 5.4 of [3]. There are five runs of each experiment. Each run involves an approximately sparse signals obtained from a permutation of the Haar 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 0.1. For more specific details, refer to [3]. In applications, the signal to be recovered is often approximately sparse rather than exactly sparse. Again, high accuracy is important when solving these problems.

It can be 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, all versions of FPC and SPARSA no longer converge in 20,000 function calls. PARNES still performs well, converging in under 2000 iterations for all runs. The accuracy of the various algorithms is compared in Table 3.

6.1 Choice of parameters

As Tseng observed, accelerated proximal gradient algorithms will converge so long as the condition given as equation (45) in [32] is satisfied. In our case this translates into

$$\min_{x \in \mathbb{R}^n} \left\{ \nabla f(y_k)^\top x + \frac{L}{2}\|x - x_k\|_2^2 + P(x) \right\} \geq \nabla f(y_k)^\top y_k + P(y_k), \tag{15}$$

Table 4 Number of function calls where the sparsity level is $s = m/5$ and the stopping rule is (13).

Method	20 dB	40 dB	60 dB	80 dB	100 dB
PARNES	122	172	214	470	632
NESTA	383	809	1639	4341	15227
NESTA + CT	483	513	583	685	787
GPSR	64	622	5030	DNC	DNC
GPSR + CT	271	219	357	1219	11737
SPARSA	323	387	465	541	693
SPGL1	58	102	191	374	504
FISTA	69	267	1020	3465	12462
FPC-AS	209	231	299	371	287
FPC-AS (CG)	253	289	375	481	361
FPC	474	386	478	1068	9614
FPC-BB	164	168	206	278	1082
BREGMAN-BB	211	223	309	455	1408

Table 5 Number of function calls where the sparsity level is $s = m/5$ and the stopping rule is (14).

Method	20 dB	40 dB	60 dB	80 dB	100 dB
PARNES	74	116	166	364	562
NESTA	383	809	1639	4341	15227
NESTA + CT	483	513	583	685	787
GPSR	62	618	5026	DNC	DNC
GPSR + CT	271	219	369	1237	11775
SPARSA	323	387	463	541	689
SPGL1	43	99	185	365	488
FISTA	72	261	1002	3477	12462
FPC-AS	115	167	159	371	281
FPC-AS (CG)	142	210	198	481	355
FPC	472	386	466	1144	9734
FPC-BB	164	164	202	276	1092
BREGMAN-BB	211	223	309	455	1408

Table 6 Recovery results of an approximately sparse signal (with Gaussian noise of variance 1 added) and with (14) as a stopping rule.

Method	Run 1	Run 2	Run 3	Run 4	Run 5
PARNES	838	810	1038	1098	654
NESTA	8817	10867	9887	9093	11211
NESTA + CT	3807	3045	3047	3225	2735
GPSR	DNC	DNC	DNC	DNC	DNC
GPSR + CT	DNC	DNC	DNC	DNC	DNC
SPARSA	2143	2353	1977	1613	DNC
SPGL1	916	892	1115	1437	938
FISTA	3375	2940	2748	2538	3855
FPC-AS	DNC	DNC	DNC	DNC	DNC
FPC-AS (CG)	DNC	DNC	DNC	DNC	DNC
FPC	DNC	DNC	DNC	DNC	DNC
FPC-BB	5614	7906	5986	4652	6906
BREGMAN-BB	3288	1281	1507	2892	3104

Table 7 Recovery results of an approximately sparse signal (with Gaussian noise of variance 0.1 added) and with (14) as a stopping rule.

Method	Run 1	Run 2	Run 3	Run 4	Run 5
PARNES	1420	1772	1246	1008	978
NESTA	11573	10457	10705	8807	13795
NESTA + CT	7543	13655	11515	3123	2777
GPSR	DNC	DNC	DNC	DNC	DNC
GPSR + CT	DNC	DNC	DNC	DNC	DNC
SPARSA	12509	DNC	DNC	3117	DNC
SPGL1	1652	1955	2151	1311	2365
FISTA	10845	12165	10050	7647	11997
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	3900	3684	2045	3292	3486

upon setting $\gamma_k = 1$ and

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

in (45) in [32]. 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 suggested to decrease L adaptively by a constant factor until (45) is violated then backtrack and repeat the iteration (cf. Note 6 in [32]). 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 (15) is satisfied for all k, and likewise for the tolerance η in Algorithm 2. 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 the results given by currently available state-of-the-art methods.

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.

References

1. Barzilai, J., Borwein, J.: Two point step size gradient method. IMA J. Numer. Anal. **8**(1), 141-148 (1988)

[1] `http://www.acm.caltech.edu/~nesta/NESTA_ExperimentPackage.zip`

[2] `http://www.cs.ubc.ca/labs/scl/spgl1`

2. Beck, A., Teboulle, M.: Fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183-202 (2009)
3. Becker, S., Bobin, J., Candès, E. J.: NESTA: a fast and accurate first-order method for sparse recovery. Preprint, (2009)
4. Bertsekas, D. P.: Nonlinear Programming. Belmont, MA (1999)
5. Birgin, E., Martínez, J., Raydan, M.: Nonmonotone spectral projected-gradient methods on convex sets. SIAM J. Optim. **10**(4), 1196–1211 (2000)
6. Bobin, J., Stark, J.-L., Ottensamer, R.: Compressed sensing in astronomy. IEEE J. Selected Top. Signal Process. **2**(5), 718-726 (2008)
7. Candès, E. J., Tao, T.: Decoding by linear programming. IEEE Trans. Inform. Theory **51**(12), 4203-4215 (2005)
8. Candès, E. J., Tao, T.: The Dantzig selector: statistical estimation when p is much larger than n. Ann. Statist. **35**(6), 2313-2351 (2007)
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)
10. Candès, E. J., Romberg, J., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Comm. Pure Appl. Math. **59**(8), 1207-1223 (2006)
11. Dembo, R. S., Eisenstat, S. C., Steihaug, T.: Inexact newton methods. SIAM J. Numer. Anal. **19**(2), 400-408 (1982)
12. Chen, S., Donoho, D. L., Saunders, M.: Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. **20**(1), 33-61 (1998)
13. Donoho, D. L.: For most large underdetermined systems of linear equations the ℓ_1-norm solution is also the sparsest solution. Comm. Pure Appl. Math. **59**(6), 797-829 (2006)
14. 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)
15. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Ann. Statist. **32**(2), 407-499 (2004)
16. Figueiredo, M., Nowak, R., Wright, S.: Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. IEEE J. Selected Top. Signal Process. **1**(4), 586-597 (2007)
17. Figueiredo, M., Nowak, R., Wright, S.: Sparse reconstruction by separable approximation. IEEE Trans. Signal Process. (to appear)
18. Garey, M. R., Johnson, D.S.: Computers and Intractability. A guide to the theory of NP-completeness. W. H. Freeman, New York, NY (1979)
19. 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)
20. Hale, E. T., Yin, W., Zhang, Y.: Fixed-point continuation for ℓ_1-minimization: Methodology and convergence. SIAM J. Optimization **19**(3), 1107-1130 (2008)
21. Hennenfent, G., Herrmann, F. J.: Simply denoise: wavefield reconstruction via jittered undersampling. Geophysics **73**(3), V19-V28 (2008)
22. 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)
23. Natarajan, B. K.: Sparse approximate solutions to linear systems. SIAM J. Comput. **24**(2), 227-234 (1995)
24. 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)
25. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. **103**(1), 127-152 (2005)
26. Osborne, M. R., Presnell, B., Turlach, B. A.: On the LASSO and its dual. J. Comput. Graph. Statist. **9**(2), 319-337 (2000)
27. 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)
28. Rockafellar, R. T.: Convex Analysis. Princeton University Press, Princeton, NJ (1970)
29. Romberg, J.: Imaging via compressive sensing. IEEE Trans. Signal Process. **25**(2), 14-20 (2008)
30. Tibshirani, R.: Regression shrinkage and selection via the LASSO. J. Roy. Statist. Soc. Ser. B **58**(1), 267-288 (1996)
31. Tropp, J. A.: Just relax: Convex programming methods for identifying sparse signals in noise. IEEE Trans. Inform. Theory **52**(3), 1030-1051 (2006)
32. Tseng, P.: On accelerated proximal gradient methods for convex-concave optimization. Preprint (2008)
33. 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)
34. 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. Software **35**(4) Art. 29, pp. 16 (2009)
35. Wen, Z., Yin, W., Goldfarb, D., Zhang, Y.: A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization and continuation. Preprint (2009)
36. Yin, W., Osher, S., Goldfarb, D., Darbon, J.: Bregman iterative algorithms for l_1 minimization with applications to compressed sensing. SIAM J. Imaging Sci. **1**(1) 143-168 (2008)

37. Afonso, M., Bioucas-Dias, J., Figueiredo, M.: Fast image recovery using variable splitting and constrained optimization. IEEE Transactions on Image Processing **19**(9) 2345-2356 (2010)
38. Afonso, M., Bioucas-Dias, J., Figueiredo, M.: Fast frame-based image deconvolution using variable splitting and constrained optimization. IEEE/SP 15th Workshop on Statistical Signal Processing, 2009. SSP '09, 109-112 (2009)