Chen *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:65
DOI 10.1186/s13634-015-0247-5

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

A fast and accurate algorithm for ℓ_1 minimization problems in compressive sampling

Feishe Chen[1], Lixin Shen[1,2*], Bruce W. Suter[2] and Yuesheng Xu[1]

Abstract

An accurate and efficient algorithm for solving the constrained ℓ_1-norm minimization problem is highly needed and is crucial for the success of sparse signal recovery in compressive sampling. We tackle the constrained ℓ_1-norm minimization problem by reformulating it via an indicator function which describes the constraints. The resulting model is solved efficiently and accurately by using an elegant proximity operator-based algorithm. Numerical experiments show that the proposed algorithm performs well for sparse signals with magnitudes over a high dynamic range. Furthermore, it performs significantly better than the well-known algorithm NESTA (a shorthand for Nesterov's algorithm) and DADM (dual alternating direction method) in terms of the quality of restored signals and the computational complexity measured in the CPU-time consumed.

Keywords: Compressive sensing; ℓ_1 minimization; Proximity operator

1 Introduction

In this paper, we study the recovery of an unknown vector $u_0 \in \mathbb{R}^n$ from the observed data $b \in \mathbb{R}^m$ and the model

$$b = Au_0 + z, \tag{1}$$

where A is a known $m \times n$ measurement matrix and $z \in \mathbb{R}^m$ is a noise term. Under an assumption that the vector u_0 of interest is sparse, the work in [1, 2] shows that an accurate estimation of u_0 is possible even when $m < n$, that is, the observations are fewer than unknowns. Recently, there is a significant body of work that focuses on finding an approximation to u_0 by solving a convex optimization problem. In the presence of noise-free data, i.e., $z = 0$, the optimization problem is

$$\text{(BP)} \qquad \min\{\|u\|_1 : u \in \mathbb{R}^n\} \quad \text{s.t.} \quad b = Au,$$

which essentially is the basis pursuit problem proposed early in the context of time-frequency representation [3]. Here, $\|\cdot\|_1$ denotes the ℓ_1-norm of a vector in an Euclidean space. The optimization model (BP) can be solved by linear programming.

In the presence of noisy data, the linear constraint $b = Au$ in (BP) is relaxed to an inequality constraint $\|Au - b\|_2 \leq \epsilon$, where $\|\cdot\|_2$ denotes the ℓ_2-norm of a vector in an Euclidean space. As a result, the optimization model (BP) becomes the basis pursuit denoising problem

$$(\text{BP}_\epsilon) \qquad \min\{\|u\|_1 : u \in \mathbb{R}^n\} \quad \text{s.t.} \quad \|Au - b\|_2 \leq \epsilon,$$

where ϵ^2 is an estimated upper bound of the noise power.

Both problems (BP) and (BP$_\epsilon$) are closely related to the penalized least squares problem

$$(\text{QP}_\lambda) \qquad \min\left\{\frac{1}{2}\|Au - b\|_2^2 + \lambda\|u\|_1 : u \in \mathbb{R}^n\right\}.$$

A large amount of research has been done on solving problems (BP), (BP$_\epsilon$), and (QP$_\lambda$). Here, we only give a brief and non-exhaustive review of results for these problems. In [3], problems (BP) and (QP$_\lambda$) are solved by first reformulating them as perturbed linear programming and then applying a primal-dual interior-point approach [4]. Recently, many iterative shrinkage/thresholding algorithms are proposed to handle problem (QP$_\lambda$). These include the proximal forward-backward splitting [5], the gradient projection for sparse reconstruction [6], the fast iterative shrinkage-thresholding algorithm (FISTA) [7], the fixed-point continuation algorithm [8], the Bregman

*Correspondence: lshen03@syr.edu
[1] Department of Mathematics, Syracuse University, Syracuse, NY 13244, USA
[2] Air Force Research Laboratory, Rome, NY 13441, USA

iterative regularization [9, 10], and the reference therein. Problem (BP_ϵ) also frequently appears in wavelet-based signal/image restoration [11, 12] with the matrix A associated with some inverse transforms.

Problem (BP_ϵ) can be formulated as a second-order cone program and solved by interior-point algorithms. Many suggested algorithms for (BP_ϵ) are based on repeatedly solving (QP_λ) for various values of λ. Such algorithms are referred to as the homotopy method originally proposed in [13, 14]. The homotopy method is also successfully applied to (BP) in [15]. A common approach for obtaining approximate solutions to (BP_ϵ) is often accompanied by solving (QP_λ) for a decreasing sequence of values of λ [16]. The optimization theory asserts that problems (BP_ϵ) and (QP_λ) are equivalent provided that the parameters ϵ and λ satisfy certain relationship [17]. Since this relationship is hard to compute in general, solving problem (BP_ϵ) via repeatedly solving (QP_λ) for various values of λ is problematic. Recently, the NESTA [18] which employs Nesterov's optimal gradient method was proposed for solving relaxed versions of (BP) and (BP_ϵ) via Nesterov's smoothing technique [19]. Clearly, the closeness of the solution to the relaxed version of (BP) (or the relaxed version of (BP_ϵ)) to the solution to (BP) (or (BP_ϵ)) is determined by the level of the closeness of the smoothed ℓ_1-norm to the ℓ_1-norm itself. Certainly, the performance of these approaches depends on the fine tuning of the parameter λ in (QP_λ) or a parameter that controls the degree of the closeness of the ℓ_1-norm and its smoothed version.

In this paper, we consider solving problems (BP) and (BP_ϵ) by a different approach. We convert the constrained optimization problems to unified unconstrained one via an indicator function. The corresponding objective function for the unconstrained optimization problem is the sum of the ℓ_1-norm of the underlying signal u and the indicator function of a set in $\mathbb{R}^m$, which is $\{0\}$ for (BP) or the ϵ-ball for (BP_ϵ), composing with the affine transformation $Au - b$. Non-differentiability of both the ℓ_1-norm and the indicator of the set imposes challenges for solving the associated optimization problem. Fortunately enough, their proximity operators have explicit expressions. The solutions for the problem can be viewed as fixed-points of a coupled equation formed in terms of these proximity operators. An iterative algorithm for finding the fixed-points is then developed. The main advantage of this approach is that solving (QP_λ) or smoothing the ℓ_1-norm are no longer necessary. This makes the proposed algorithm attractive for solving (BP) and (BP_ϵ). The efficiency of fixed-point-based proximity algorithms has been demonstrated in [5] and [20–22] for various image processing models.

The rest of the paper is organized as follows: in Section 2, we reformulate the ℓ_1-norm minimization problems (BP) and (BP_ϵ) via an indicator function and characterize solutions of the proposed model in terms of fixed-point equations. We also point out the connection between the proposed model and (QP_λ) through the Moreau envelope. In Section 3, we develop an algorithm for the resulting minimization problem based on the fixed-point equations arising from the characterization of the proposed model. Numerical experiments are presented in Section 4. We draw our conclusions in Section 5.

2 An ℓ_1-norm optimization model via an indicator function

In this section, we consider a general optimization model that includes models (BP) and (BP_ϵ) as its special cases and characterize solutions to the proposed model.

We begin with introducing our notation and recalling necessary background from convex analysis. For the usual d-dimensional Euclidean space denoted by $\mathbb{R}^d$, we define $\langle x, y\rangle := \sum_{i=1}^d x_i y_i$, for $x, y \in \mathbb{R}^d$, the standard inner product in $\mathbb{R}^d$. The class of all lower semicontinuous convex functions $f : \mathbb{R}^d \to (-\infty, +\infty]$ such that $\mathrm{dom} f := \{x \in \mathbb{R}^d : f(x) < +\infty\} \neq \emptyset$ is denoted by $\Gamma_0(\mathbb{R}^d)$. The indicator function of a closed convex set C in $\mathbb{R}^d$ is defined, at $u \in \mathbb{R}^d$, as

$$\iota_C(u) := \begin{cases} 0, & \text{if } u \in C, \\ +\infty, & \text{otherwise.} \end{cases}$$

Clearly, the indicator function ι_C is in $\Gamma_0(\mathbb{R}^d)$ for any closed nonempty convex set C. In particular, we define a ball in $\mathbb{R}^m$ centered at the origin with radius ϵ as $B_\epsilon := \{v : v \in \mathbb{R}^m \text{ and } \|v\|_2 \leq \epsilon\}$.

Given a matrix $A \in \mathbb{R}^{m\times n}$ and a vector $b \in \mathbb{R}^m$, we consider the following optimization problem

$$\min\left\{\|u\|_1 + \iota_{B_\epsilon}(Au - b) : u \in \mathbb{R}^n\right\}. \tag{2}$$

We can easily see that if $\epsilon = 0$, then model (2) reduces to (BP), and if $\epsilon > 0$, then model (2) reduces to (BP_ϵ). In other words, both constrained optimization problems (BP) and (BP_ϵ) can be unified as the unconstrained optimization problem (2) via the indicator function ι_{B_ϵ}.

In the following, we shall focus on characterizing solutions of model (2) using fixed-point equations. To characterize solutions of model (2), we first need two concepts, namely, the proximity operator and subdifferential of functions in $\Gamma_0(\mathbb{R}^d)$. For a function $f \in \Gamma_0(\mathbb{R}^d)$, the proximity operator of f with parameter λ, denoted by $\mathrm{prox}_{\lambda f}$, is a mapping from $\mathbb{R}^d$ to itself, defined for a given point $x \in \mathbb{R}^d$ by

$$\mathrm{prox}_{\lambda f}(x) := \mathrm{argmin}\left\{\frac{1}{2\lambda}\|u - x\|_2^2 + f(u) : u \in \mathbb{R}^d\right\}.$$

The subdifferential of a proper convex function $\psi \in \Gamma_0(\mathbb{R}^d)$ at a given vector $u \in \mathbb{R}^d$ is the set defined by

$$\partial\psi(u) := \left\{ v \in \mathbb{R}^d : \psi(w) \geq \psi(u) + \langle v, w-u \rangle,\ \forall\, w \in \mathbb{R}^d \right\}.$$

The subdifferential and the proximity operator of the function ψ are related in the following way (see, e.g. [21]): for u in the domain of ψ and $v \in \mathbb{R}^d$

$$v \in \partial\psi(u) \text{ if and only if } u = \operatorname{prox}_{\psi}(u+v). \tag{3}$$

Now, with the help of the subdifferential and the proximity operator, we can characterize a solution of the indicator function based on model (2) via fixed-point equations.

Proposition 2.1. *Let ϵ be a nonnegative number, let B_ϵ be the ball in $\mathbb{R}^m$ centered at the origin with radius ϵ, let b be a point in $\mathbb{R}^m$, and let A be an $m \times n$ matrix. If $u \in \mathbb{R}^n$ is a solution to model (2), then for any $\alpha > 0$ and $\beta > 0$, there exists a vector $v \in \mathbb{R}^m$ such that*

$$u = prox_{\frac{1}{\alpha}\|\cdot\|_1}\left(u - \frac{\beta}{\alpha}A^\top v\right), \tag{4}$$

$$v = \left(I - prox_{\iota_{B_\epsilon}(\cdot - b)}\right)(Au + v). \tag{5}$$

Conversely, if there exist $\alpha > 0$, $\beta > 0$, $u \in \mathbb{R}^n$, and $v \in \mathbb{R}^m$ satisfying (4) and (5), then u is a solution of model (2).

Proof. We first assume that $u \in \mathbb{R}^n$ is a solution of model (2). Set $\varphi := \iota_{B_\epsilon}(\cdot - b)$. Hence, $Au - b$ must be in the ball B_ϵ. Therefore, both sets $\partial\|\cdot\|_1(u)$ and $\partial\varphi(Au)$ are nonempty. By Fermat's rule, we have that $0 \in \partial\|\cdot\|_1(u) + A^\top\partial\varphi(Au)$. Therefore, for any $\alpha > 0$ and $\beta > 0$, there exist $w \in \frac{1}{\alpha}\partial\|\cdot\|_1(u)$ and $v \in \frac{1}{\beta}\partial\varphi(Au)$ such that $0 = \alpha w + \beta A^\top v$, i.e., $w = -\frac{\beta}{\alpha}A^\top v$. By using (3), inclusion $w \in \frac{1}{\alpha}\partial\|\cdot\|_1(u)$ implies $u = \operatorname{prox}_{\frac{1}{\alpha}\|\cdot\|_1}(u+w)$, which is (4). Since $\frac{1}{\beta}\varphi = \varphi$ for any $\beta > 0$, inclusion $v \in \frac{1}{\beta}\partial\varphi(Au)$ leads to $Au = \operatorname{prox}_{\varphi}(v + Au)$, which is equivalent to (5).

Conversely, if (4) and (5) are satisfied for some $\alpha > 0$, $\beta > 0$, $u \in \mathbb{R}^n$, and $v \in \mathbb{R}^m$, using (3) again, we have that $-\frac{\beta}{\alpha}A^\top v \in \partial\left(\frac{1}{\alpha}\|\cdot\|_1\right)(u)$ and $v \in \partial\varphi(Au)$. Since $\partial\left(\frac{1}{\alpha}\|\cdot\|_1\right)(u) = \frac{1}{\alpha}\partial\|\cdot\|_1(u)$ and $\partial\varphi(Au) = \beta\partial\varphi(Au)$, we know from the above that $0 \in \partial\|\cdot\|_1(u) + A^\top\partial\varphi(Au)$. This indicates that u is a solution of model (2). The proof is complete. □

We remark that the above fixed-point characterization can be identified as a special case of Proposition 1 in [22]. We include the proof of Proposition 2.1 here for making the paper self-contained.

The proximity operators of the functions $\|\cdot\|_1$ and $\iota_{B_\epsilon}(\cdot - b)$ involved in the characterization can be computed efficiently. Indeed, the proximity operator $\operatorname{prox}_{\frac{1}{\alpha}\|\cdot\|_1}$ is the soft-thresholding operator defined for $u \in \mathbb{R}^n$ by:

$$\left(\operatorname{prox}_{\frac{1}{\alpha}\|\cdot\|_1}(u)\right)[i] = \max\left\{|u[i]| - \frac{1}{\alpha}, 0\right\}\operatorname{sign}(u[i]), \tag{6}$$

for $i = 1, 2, \ldots, n$.

The proximity operator $\operatorname{prox}_{\iota_{B_\epsilon}(\cdot - b)}$ is given by the following lemma.

Lemma 2.2. *Let ϵ be a nonnegative number and let b be a point in $\mathbb{R}^m$. Then, for a given $v \in \mathbb{R}^m$*

$$prox_{\iota_{B_\epsilon}(\cdot - b)}(v) = b + \min\left\{1, \frac{\epsilon}{\|v - b\|_2}\right\}(v - b).$$

Proof. By the definition of the proximity operator, we can verify directly that $\operatorname{prox}_{\iota_{B_\epsilon}(\cdot - b)} = b + \operatorname{prox}_{\iota_{B_\epsilon}}(\cdot - b)$ and $\operatorname{prox}_{\iota_{B_\epsilon}}$ is the projection operator onto the ball B_ϵ. The result of this lemma follows immediately. □

3 An algorithm and its convergence

In this section, we develop an algorithm for finding a solution of model (2) and provide a convergence analysis for the developed algorithm.

As we already know, all solutions of model (2) should satisfy the fixed-point equations given by (4) and (5). By introducing an auxiliary variable $w = \operatorname{prox}_{\iota_{B_\epsilon}(\cdot - b)}(Au + v)$, we have the following equivalent form of (4) and (5)

$$\begin{cases} u = \operatorname{prox}_{\frac{1}{\alpha}\|\cdot\|_1}\left(\left(I - \frac{\beta}{\alpha}A^\top A\right)u - \frac{\beta}{\alpha}A^\top(v - w)\right), \\ w = \operatorname{prox}_{\iota_{B_\epsilon}(\cdot - b)}(Au + v), \\ v = Au + v - w. \end{cases} \tag{7}$$

Based on the above fixed-point equations in terms of u, w, and v, for arbitrary initial vectors $u^0 \in \mathbb{R}^n$, $w^0, v^0 \in \mathbb{R}^m$, we generate the sequence $\{u^k : k \in \mathbb{N}_0\}$ with $\mathbb{N}_0 := \{0, 1, \ldots\}$ by the following iterative scheme

$$\begin{cases} u^{k+1} = \operatorname{prox}_{\frac{1}{\alpha}\|\cdot\|_1}\left(\left(I - \frac{\beta}{\alpha}A^\top A\right)u^k - \frac{\beta}{\alpha}A^\top\left(v^k - w^k\right)\right), \\ w^{k+1} = \operatorname{prox}_{\iota_{B_\epsilon}(\cdot - b)}\left(Au^{k+1} + v^k\right), \\ v^{k+1} = Au^{k+1} + v^k - w^{k+1}. \end{cases} \tag{8}$$

To show convergence of the iterative scheme (8), we recall a result from [20].

Lemma 3.1 (Theorem 3.5 in [20]). *If x is a vector in $\mathbb{R}^n$, A is an $m \times n$ matrix, φ is in $\Gamma_0(\mathbb{R}^m)$, and α, β, λ are positive numbers such that $\frac{\beta}{\lambda\alpha} < \frac{1}{\|A\|^2}$, then the sequence $\{u^k : k \in \mathbb{N}_0\}$ generated by the following iterative scheme*

$$\begin{cases} u^{k+1} = x + prox_{\frac{1}{\alpha}\|\cdot\|_1}\left(\left(I - \frac{\beta}{\lambda\alpha}A^\top A\right)u^k \right. \\ \qquad\qquad \left. -x - \frac{\beta}{\lambda\alpha}A^\top\left(v^k - w^k\right)\right), \\ w^{k+1} = prox_{\frac{1}{\beta}\varphi}\left(Au^{k+1} + v^k\right), \\ v^{k+1} = Au^{k+1} + v^k - w^{k+1} \end{cases} \qquad (9)$$

converges to a solution of the optimization problem

$$\min\left\{\lambda\|u - x\|_1 + (\varphi \circ A)(u) : u \in \mathbb{R}^m\right\}. \qquad (10)$$

With the help of Lemma 3.1, the following result shows that under appropriate conditions on parameters α and β, the sequence $\{u^k : k \in \mathbb{N}_0\}$ converges to a solution of model (2).

Theorem 3.2. *Let ϵ be a nonnegative number, let B_ϵ be the ball in $\mathbb{R}^m$ centered at the origin with radius ϵ, let b be a point in $\mathbb{R}^m$, and let A be an $m \times n$ matrix. If*

$$\frac{\beta}{\alpha} < \frac{1}{\|A\|^2}, \qquad (11)$$

then for arbitrary initial vectors $u^0 \in \mathbb{R}^n$, $w^0, v^0 \in \mathbb{R}^m$, the sequence $\{u^k : k \in \mathbb{N}_0\}$ generated by the iterative scheme (8) converges to a solution of model (2).

Proof. By setting $x = 0$ and $\lambda = 1$ and identifying $\varphi = \iota_{B_\epsilon}(\cdot - b)$ in model (10), the iterative scheme (8) can be viewed as a special case of the one given in (9). The desired result follows immediately from Lemma 3.1. □

The convergence result given by Theorem 3.2 offers a practical way to find a solution of model (2). Since the explicit forms of the proximity operators $\mathrm{prox}_{\frac{1}{\alpha}\|\cdot\|_1}$ and $\mathrm{prox}_{\iota_{B_\epsilon}(\cdot - b)}$ are given by (6) and Lemma 2.2, respectively, based on Theorem 3.2, a unified approach for solving both (BP) and (BP$_\epsilon$) is depicted in Algorithm 1.

Algorithm 1 The iterative scheme for model (BP$_\epsilon$) with $\epsilon \geq 0$

```
Initialization: $v^0 \in \mathbb{R}^m$, $w^0 \in \mathbb{R}^m$, $u^0 \in \mathbb{R}^n$, $\epsilon > 0$, $\alpha > 0$, and $\beta > 0$ with $\frac{\beta}{\alpha} < \frac{1}{\|A\|^2}$.
repeat($k \geq 0$)
    Step 1:
        $u^{k+1} \leftarrow \mathrm{prox}_{\frac{1}{\alpha}\|\cdot\|_1}\left(\left(I - \frac{\beta}{\alpha}A^\top A\right)u^k - \frac{\beta}{\alpha}A^\top(v^k - w^k)\right)$
    Step 2: Denote $p^k := Au^{k+1} + v^k - b$.
        $w^{k+1} \leftarrow \begin{cases} Au^{k+1} + v^k, & \text{if } \|p^k\|_2 < \epsilon; \\ b + \epsilon\frac{p^k}{\|p^k\|_2}, & \text{otherwise.} \end{cases}$
    Step 3: $v^{k+1} \leftarrow Au^{k+1} + v^k - w^{k+1}$
until a given stopping criteria is met
```

We remark that Algorithm 1 derived from the fixed-point characterization of model (2) is closed related to existing algorithms based on the idea of augmented direction method for model. We briefly review an alternating direction method for model (2) that is equivalently written as the following constrained optimization problem

$$\min\left\{\|u\|_1 + \iota_{B_\epsilon}(w - b) : Au = w, u \in \mathbb{R}^n, w \in \mathbb{R}^m\right\}. \qquad (12)$$

The primal and dual alternating direction methods for solving (12) can be found in [23]. The generalized alternating direction method for (12), proposed in [24], iterates as follows: given $(u^0, w^0, \lambda^0) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$,

$$\begin{cases} u^{k+1} = \mathrm{argmin}_u\left\{\|u\|_1 + \langle\lambda^k, Au - w^k\rangle + \frac{\beta}{2}\|Au - w^k\|_2^2 + \frac{1}{2} \right. \\ \qquad\qquad \left. (u - u^k)P(u - u^k)\right\}, \\ w^{k+1} = \mathrm{argmin}_w\left\{\iota_{B_\epsilon}(w - b) + \langle\lambda^k, Au^{k+1} - w\rangle + \frac{\beta}{2} \right. \\ \qquad\qquad \left. \|Au^{k+1} - w\|_2^2\right\}, \\ \lambda^{k+1} = \lambda^k + \beta\left(Au^{k+1} - w^{k+1}\right), \end{cases} \qquad (13)$$

where the matrix $P = \alpha I - \beta A^\top A$ is positive definite. The condition $\frac{\beta}{\alpha} < \frac{1}{\|A\|^2}$ ensures the positive definiteness of P. The technique of introducing the term $(u - u^k)P(u - u^k)$ was used earlier in [25]. It can be easily seen that the iterative scheme (13) is equivalent to (8) with $\lambda^k = \beta v^k$. It is worth pointing out that if P is the zero matrix, then the iterative scheme (13) reduces to the conventional alternative direction method of multipliers (ADMM) for the constrained optimization problem (12) (see, e.g., [26]); in this case, the u-subproblem in (13) has no explicit solution and must be solved by an appropriate iterative algorithm, for example, FISTA in [7].

We further make some comments on Algorithm 1. Step 1 of computing u^{k+1} is from the first equation in (8); step 2 of computing w^{k+1} is from the second equation in (8) and Lemma 2.2; step 3 of computing v^{k+1} is exactly the same as the last equation in (8). This algorithm can be presented in a more computationally efficient way by combining step 2 and step 3 and eliminating the intermediate variable w^k. The motivation comes from the observation $Au^k - w^k = v^k - v^{k-1}$ which is due to the third step of Algorithm 1. Substituting $Au^k - w^k$ in step 1 by $v^k - v^{k-1}$ yields that

$$u^{k+1} = \mathrm{prox}_{\frac{1}{\alpha}\|\cdot\|_1}\left(u^k - \frac{\beta}{\alpha}A^\top\left(2v^k - v^{k-1}\right)\right), \qquad (14)$$

with an assumption $v^{-1} = v^0 - (Au^0 - w^0)$ for the given u^0, w^0, and v^0. We can further substitute w^{k+1} computed in step 2 into step 3. In this way, the intermediate variable w^k is no longer needed. Hence, these simplifications yield Algorithm 2, a variant of Algorithm 1. When $\epsilon = 0$, all vectors w^{k+1} in Algorithm 1 are equal to the constant vector b for all $k \geq 0$. Because of this, we would like to set $w^0 = b$ in both Algorithms 1 and 2. Finally, it is more efficient to update u^{k+1} with step 1 of Algorithm 2 than with step 1 of Algorithm 1 in each iteration since the matrix-vector multiplication involving A is not required in (14). However, updating u^{k+1} via the formulation of step 2 in Algorithm 1 can be implemented through the use of the component-wise Gauss-Seidel iteration which may accelerate the rate of convergence of the algorithm and therefore reduce the total CPU time consumed. The efficiency of component-wise Gauss-Seidel iteration has been verified in [20, 21].

Algorithm 2 A variant of Algorithm 1 for model (BP_ϵ)

```
Initialization: $v^0 \in \mathbb{R}^m$, $u^0 \in \mathbb{R}^n$, $\epsilon > 0$, $\alpha > 0$, and $\beta > 0$ with $\frac{\beta}{\alpha} < \frac{1}{\|A\|^2}$; set $v^{-1} = v^0 - (Au^0 - d^0)$,
repeat($k \geq 0$)
    Step 1:
    $u^{k+1} \leftarrow \mathrm{prox}_{\frac{1}{\alpha}\|\cdot\|_1}\left(u^k - \frac{\beta}{\alpha}A^\top\left(2v^k - v^{k-1}\right)\right)$
    Step 2: Denote $p^k := Au^{k+1} + v^k - b$.
    $v^{k+1} \leftarrow \begin{cases} 0, & \text{if } \|p^k\|_2 < \epsilon; \\ \left(1 - \frac{\epsilon}{\|p^k\|_2}\right)\left(p^k\right), & \text{otherwise.} \end{cases}$
until a given stopping criteria is met
```

Algorithm 2 for model (2) can be viewed as the primal-dual algorithm proposed in [27]. To make this connection, we need the notion of the conjugate function. The conjugate of $f \in \Gamma_0(\mathbb{R}^d)$ is the function $f^* \in \Gamma_0(\mathbb{R}^d)$ defined at $y \in \mathbb{R}^d$ by $f^*(y) := \sup\{\langle x, y\rangle - f(x) : x \in \mathbb{R}^d\}$. By the Fenchel-Moreau theorem in convex analysis, $f = f^{**}$ for all $f \in \Gamma_0(\mathbb{R}^d)$. In particular, we have that $\iota^*_{B_\epsilon} = \epsilon\|\cdot\|_2$ and $\iota_{B_\epsilon} = \iota^{**}_{B_\epsilon} = \sup\{\langle v, \cdot\rangle - \epsilon\|v\|_2 : v \in \mathbb{R}^m\}$. Since $\iota_{B_\epsilon} = \beta\iota_{B_\epsilon}$ for any $\beta > 0$, we have that $\iota_{B_\epsilon}(p) = \sup\{\beta\langle v, p\rangle - \beta\epsilon\|v\|_2 : v \in \mathbb{R}^m\}$ for $p \in \mathbb{R}^m$ and $\beta > 0$. Therefore, the saddle point problem associated with model (2) is

$$\min_{u\in\mathbb{R}^n}\max_{v\in\mathbb{R}^m}\left\{\|u\|_1 + \beta\langle v, Au - b\rangle - \beta\epsilon\|v\|_2\right\}, \tag{15}$$

where u is the primal variable and v is the dual variable. An alternating iterative scheme for solving the saddle point problem (15) proposed in [27] is as follows:

$$\begin{cases} u^{k+1} = \mathrm{argmin}_u\left\{\|u\|_1 + \beta\langle 2v^k - v^{k-1}, Au - b\rangle + \frac{\alpha}{2}\|u - u^k\|_2^2\right\}, \\ v^{k+1} = \mathrm{argmax}_v\left\{\beta\langle v, Au^{k+1} - b\rangle - \beta\epsilon\|v\|_2 - \frac{\beta}{2}\|v - v^k\|_2^2\right\}. \end{cases} \tag{16}$$

In terms of proximity operator, the updates u^{k+1} and v^{k+1} in (16) are identical to the update u^{k+1} in step 1 and the update v^{k+1} in step 2 of Algorithm 2, respectively.

4 Numerical simulations

This section is devoted to showing the numerical performance of the proposed algorithms for compressive sampling. We use NESTA [18] and dual alternating direction method (DADM) [23] as a comparison. In the comparisons, the NESTA with continuation in available code NESTA_v1.1 is applied and DADM for model (BP_ϵ) is chosen. We focus on sparse signals with various dynamic ranges and various measurement matrices including randomly partial discrete cosine transforms (DCTs), randomly partial discrete Walsh-Hadamard transforms (DWHTs), and random Gaussian matrices and evaluate performance of algorithms in terms of various error metrics, speed, and robustness-to-noise. All the experiments are performed in MATLAB 7.11 on DELL XPS 14 with Intel Core i5, 4GB RAM on Windows 8 operating system.

We begin with a description of generating the $m \times n$ sensing matrix A and length-n and s-sparse signals. The sensing matrices are divided into two categories. In the first category, the sensing matrices A satisfy $AA^\top = I$ while in the other one condition $AA^\top = I$ is not satisfied. In the first category, when the $m \times n$ sensing matrix A is partial DCT or DWHT, it is generated by randomly picking m rows from the $n \times n$ DCT matrix or DWHT matrix; when A is random Gaussian, it is elements are randomly generated independently from standard normal distribution and then its rows are orthonormalized. In the second category, elements of A are randomly generated independently from Gaussian distribution. Sparse signals u used in our experiments are generated according to [18]. In each experimental trial, a length-n, s-sparse signal (a signal having exactly s nonzero components), is generated in such a way that non-zero components are given by

$$\eta_1 10^{\theta\eta_2}, \tag{17}$$

where $\eta_1 = \pm 1$ with probability $1/2$ and η_2 is uniformly distributed in $[0, 1]$. The locations of the nonzero components are randomly permuted. Clearly, the range of the magnitude of nonzero components of an s-sparse signal is $[1, 10^\theta]$ with the parameter θ controlling this dynamic range. An observed signal (data) is collected by $b = Au + z$, where z represents a Gaussian noise.

The accuracy of a solution obtained from a specific algorithm is quantified by the relative ℓ_2-error, the relative ℓ_1-error, and the absolute ℓ_∞-error defined, respectively, as follows:

$$\|u - u_\diamond\|/\|u\|, \quad |\|u\|_1 - \|u_\diamond\|_1|/\|u\|_1, \quad \|u - u_\diamond\|_\infty, \tag{18}$$

where u is the true data and $u_\diamond$ is the restored data. All results reported in this section are the means of these relative errors and CPU time consumed from simulations that were performed 50 trials.

To use Algorithm 2, one needs to fix the parameters α and β such that $\beta/\alpha < 1/\|A\|^2$ (see Theorem 3.2).

From step 1 of the algorithm, the ratio β/α plays a role of step-size of changing u^k. We now investigate the performance of Algorithm 2 with various ratio $\beta/\alpha = \frac{0.999}{\|A\|^2}, \frac{0.999}{2\|A\|^2}, \frac{0.999}{4\|A\|^2}$, and α is fixed. We consider the configuration of $n = 2^{15}$, $m = n/2$, $s = 0.05n$, the dynamic range parameter $\theta = 1$ and the sensing matrix A is the partial DCT. The observed data is free of noise. The performance of Algorithm 2 in terms of the relative ℓ_2 error against iteration with various values of β/α is shown in Fig. 1. As it can be seen, the performance with the largest ratio $\frac{\beta}{\alpha} = \frac{0.999}{\|A\|^2}$ is the best. We therefore set

$$\beta = \frac{0.999}{\|A\|^2}\alpha \tag{19}$$

in our numerical experiments. In such the way, α is essentially the only parameter that needs to be determined. We now investigate the impact of the parameter α on the performance of Algorithm 2.

To investigate the impact of varying the parameter α on the performance of Algorithm 2, we consider the configurations of $n = 2^{15}$, $m = n/2$, $s = 0.05n$, the dynamic range parameter $\theta = 5$, and the sensing matrix A is partial DCT. The observed data is noise free. Six different values of α, namely, 0.0025, 0.005, 0.01, 0.02, 0.04, and 0.08, are tested. Figure 2a depicts the traces of the relative ℓ_1-error (see (18)) against the number of iterations for each α. As it can be seen from this figure, for $\alpha = 0.0025$, the smallest value in our test, the relative ℓ_1-error drops rapidly from 1 to 10^{-4}, stabilizes with insignificant changes for about 1200 iterations, and then quickly drops again to the level of 10^{-16}. When α increases from 0.0025 to 0.08, the number of iterations required for the relative ℓ_1-error dropping from 1 to 10^{-4} increases. Meanwhile, the numbers of iterations for the transitions from the first sharp jump region to the second one decrease. For example, it is about 700 for $\alpha = 0.005$ and only few iterations for $\alpha = 0.08$. These observations motivate us to extend Algorithm 2 to a scenario in which the parameter α can be updated during the iteration with the goal of reducing the number of iterations. The proposed approach is rather simple. It begins with a relative small α and then increases it for every given amount of iterations. A detailed flow of this new approach is given in Algorithm 3.

Algorithm 3 A variant of Algorithm 1 for model (BP$_\epsilon$)

```
Given: integers $p > 0$, $\tau > 1$, and $T > 0$; $\epsilon > 0$
Initialization: $v^0 \in \mathbb{R}^m$, $u^0 \in \mathbb{R}^n$, $\alpha > 0$, and $\beta > 0$ with $\frac{\beta}{\alpha} < \frac{1}{\|A\|^2}$; set $v^{-1} = v^0 - (Au^0 - d^0)$
repeat($k \geq 0$)
    Step 1: Compute $u^{k+1}$ using step 1 of Algorithm 2
    Step 2: Compute $v^{k+1}$ using step 2 of Algorithm 2
    Step 3: If $k$ is a multiple of $p$ and the number of changing the parameters $\alpha$ and $\beta$ does not exceed $T$, update $\alpha \leftarrow \tau\alpha$, $\beta \leftarrow \tau\beta$
until a given stopping criteria is met
```

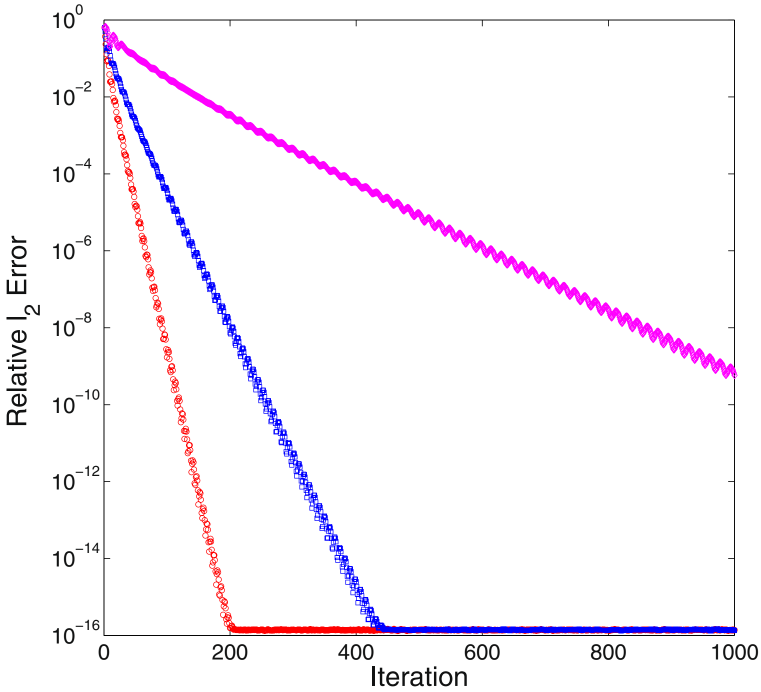

Fig. 1 The relative ℓ_2 error (the *vertical axis* with a base 10 logarithmic scale) versus the number of iterations (the *horizontal axis*) with various ratio of $\frac{\beta}{\alpha}$. *Red circle* $\frac{\beta}{\alpha} = \frac{0.999}{\|A\|^2}$, *blue square* $\frac{\beta}{\alpha} = \frac{0.999}{2\|A\|^2}$, *magenta diamond* $\frac{\beta}{\alpha} = \frac{0.999}{4\|A\|^2}$

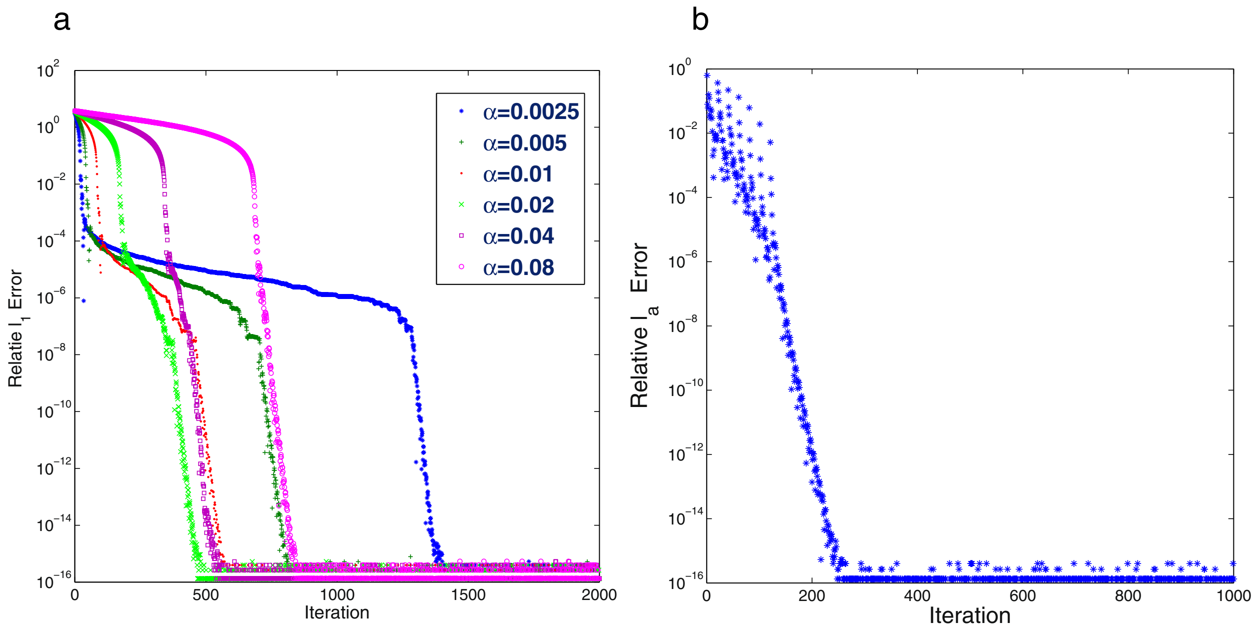

Fig. 2 The relative ℓ_1 error (the *vertical axis* with a base 10 logarithmic scale) versus the number of iterations (the *horizontal axis*). **a** Convergence of Algorithm 2 with different values of α. **b** Convergence of Algorithm 3

Three new parameters introduced in Algorithm 3 are integers $p > 0$, $\tau > 1$, and $T > 0$. The parameter T is the allowable maximum number of updating the parameters α and β. For each update, the pair (α, β) will change to $(\tau\alpha, \tau\beta)$ that will keep the ratio β/α unchanged. The parameter p is to indicate that the underlying algorithm with a pair (α, β) will iterate p times before the algorithm with the pair $(\tau\alpha, \tau\beta)$ runs another p times. We now demonstrate the efficiency of varying the parameters α and β via applying Algorithm 3 for the same data used in Fig. 2a. We set $T = 6$, $\tau = 4$, and $p = 20$ and initialize $\alpha = \frac{m}{n}\frac{20\|A\|^2}{\|A^\top b\|_\infty}$. Again, we choose β by using (19). The corresponding result is shown in Fig. 2b. It is clear to see that it takes about 200 iterations to drop the relative ℓ_1 error down below 10^{-14}. Hence, the strategy of updating the parameters α and β as described in Algorithm 3 is reasonable.

The rest of this section consists of two subsections. The first subsection focuses on comparisons of proposed algorithm to NESTA and DADM for sensing matrices A with $AA^\top = I$, while the second subsection only focuses on numerical performance of proposed algorithms for random Gaussian sensing matrices.

4.1 Numerical comparisons

This subsection consists of three parts. Part one contains the comparisons of Algorithm 3, DADM, and NESTA for data setting with partial DCT measurement matrices, part two contains that for data setting with partial DWHT measurement matrices, and part three contains results on random matrices with orthonormalized rows.

4.1.1 *Numerical comparison with partial DCT sensing matrices*

First of all, we compare the performance of Algorithm 3 with that of NESTA and that of DADM [23] for noise-free data. The algorithm NESTA was developed by applying a smoothing technique for the nonsmooth ℓ_1-norm and an accelerated first-order scheme, both from Nesterov's work [19]. A parameter denoted by μ is used to control how close the smoothed ℓ_1-norm to the ℓ_1-norm will be. To obtain high accuracy of restored signal for NESTA, $\mu = 10^{-10}$ is used for partial DCT sensing matrices and various dynamic range parameters. A parameter Tol for tolerance in NESTA varies for different values of the smoothing parameter μ and different settings of generated data and needs to be determined. We choose the tolerance to obtain reasonable results. We finally choose Tol $= 10^{-12}$, 10^{-14}, 10^{-15}, respectively, for data generated with dynamic range parameters $\theta = 1, 3, 5$. For DADM, two parameters γ and β have to be predetermined. $\gamma = 1.618$ is chosen in all settings, while β varies in different settings to obtain reasonable results. We choose parameters $\beta = \frac{\|b\|_1}{m2^1}, \frac{\|b\|_1}{m2^3}, \frac{\|b\|_1}{m2^6}$ for dynamic range parameters $\theta = 1, 3, 5$, respectively. For Algorithm 3, we set $p = 20$ and T to be the smallest integer that is greater than $\log_{10}(\frac{n}{m}\|A^\top b\|_\infty)$. In our experiments, we notice that T is θ or $\theta + 1$. The stopping criterion of Algorithm 3 and DADM is that the relative errors between the successive iterates of the reconstructed signal should satisfy the inequality $\|u^{k+1} - u^k\|/\|u^k\| < \text{Tol}$. We choose $\text{Tol} = 10^{-15}$ for data generated by partial DCT for Algorithm 3 and DADM.

For the noise-free data, the problem (BP) is used to recover underlying signals in experiments. The dimensions n are chosen from $\{2^{13}, 2^{15}\}$ for data generated with partial DCT. The number of nonzero entries s is set to be $0.02n$, $0.01n$, respectively, for the number of measurements $m = n/4$, $n/8$. The performance of different algorithms are reported in Tables 1 and 2. Based on these two tables, the performance of Algorithm 3 and DADM is comparable in terms of accuracy of recovered data for various values of dynamic range parameter θ and measurement ratio m/n. But Algorithm 3 outperforms DADM in terms of computational cost (CUP time or iterations) for data with high value of dynamic range parameter (e.g., $\theta = 5$). The performance of NESTA is inferior to that of Algorithm 3 and DADM in terms of accuracy and computational cost for various values of θ and measurement ratio m/n. We also observe that the relative ℓ_2-error and ℓ_1-error of the results recovered by Algorithm 3 along with iterations consumed are quite robust with respect to the dynamic ranges of the unknown signals.

Table 1 Numerical results with partial DCT sensing matrices for noise-free data. The number of measurements m is $m = n/4$, and the test signals are s-sparse with $s = 0.02n$. Each value in a cell represents the mean over 50 trials

Method	ℓ_2-error	ℓ_1-error	ℓ_∞-error	CPU time(s)	Iterations
$n = 2^{13}$					
Algorithm 3	4.99e−15	6.77e−16	6.55e−14	1.0153	387
DADM	4.48e−15	6.45e−16	5.24e−14	1.1525	391
NESTA	8.29e−11	1.75e−10	6.10e−10	1.8168	469
$n = 2^{15}$					
Algorithm 3	3.04e−15	3.81e−16	4.71e−14	5.1618	394
DADM	2.10e−15	2.96e−16	2.96e−14	5.6775	398
NESTA	8.48e−11	1.77e−10	6.80e−10	7.7550	477
$n = 2^{13}$					
Algorithm 3	6.20e−15	1.07e−15	4.65e−12	1.0640	394
DADM	3.96e−15	1.20e−15	2.72e−12	1.1331	388
NESTA	1.41e−12	4.70e−12	6.34e−10	2.8384	742
$n = 2^{15}$					
Algorithm 3	3.18e−15	5.39e−16	2.81e−12	5.3503	403
DADM	2.35e−15	3.96e−16	1.84e−12	5.7291	395
NESTA	1.48e−12	4.78e−12	6.95e−10	12.1293	748
$n = 2^{13}$					
Algorithm 3	4.69e−15	9.25e−16	2.94e−10	1.0637	397
DADM	3.01e−15	1.46e−15	1.74e−10	1.9593	665
NESTA	2.05e−14	1.96e−14	8.12e−10	4.7221	1236
$n = 2^{15}$					
Algorithm 3	3.04e−15	4.74e−16	2.08e−10	5.4653	404
DADM	2.05e−15	4.85e−16	1.28e−10	10.2053	691
NESTA	2.09e−14	3.04e−14	7.93e−10	19.5025	1209

Table 2 Numerical results with partial DCT sensing matrices for noise-free data. The number of measurements m is $m = n/8$, and the test signals are s-sparse with $s = 0.01n$. Each value in a cell represents the mean over 50 trials

Method	ℓ_2-error	ℓ_1-error	ℓ_∞-error	CPU time(s)	Iterations
$n = 2^{13}$					
Algorithm 3	1.24e−14	1.65e−15	1.47e−13	2.0684	776
DADM	1.11e−14	4.34e−15	1.33e−13	2.3184	803
NESTA	1.96e−10	3.89e−10	1.61e−09	2.8518	764
$n = 2^{15}$					
Algorithm 3	5.67e−15	5.81e−16	8.02e−14	10.2525	799
DADM	4.58e−15	6.12e−16	6.15e−14	12.0084	832
NESTA	2.00e−10	3.89e−10	1.79e−09	11.9268	761
$n = 2^{13}$					
Algorithm 3	9.65e−15	1.99e−15	6.91e−12	1.9584	758
DADM	1.04e−14	6.61e−15	7.02e−12	2.2513	791
NESTA	3.14e−12	9.96e−12	1.49e−09	4.5468	1216
$n = 2^{15}$					
Algorithm 3	5.09e−15	6.58e−16	4.29e−12	9.7934	762
DADM	5.21e−15	6.98e−16	4.21e−12	11.5441	817
NESTA	3.18e−12	9.97e−12	1.64e−09	18.3234	1187
$n = 2^{13}$					
Algorithm 3	1.36e−14	2.22e−15	5.62e−10	1.8825	727
DADM	7.87e−15	7.69e−15	4.09e−10	3.2409	1129
NESTA	4.68e−14	1.04e−13	2.09e−09	7.3318	1950
$n = 2^{15}$					
Algorithm 3	5.26e−15	7.99e−16	3.39e−10	9.4868	739
DADM	4.32e−15	7.42e−16	2.59e−10	17.2200	1202
NESTA	5.19e−14	3.54e−14	2.35e−09	24.6468	1793

Next, the comparison of different algorithms for noisy data is discussed. The underlying data is recovered by solving problem (BP_ϵ). The settings of dimension, sparsity, and dynamic range of unknown signals for problem (BP_ϵ) are the same as those for problem (BP). The only difference is that measurements are contaminated with noise. In our experiments, noise levels in the measurements vary with the dynamic ranges of the unknown signals. More precisely, the noise levels σ are set to be 0.05, 1.0, and 5.0 corresponding to choices 1, 3, and 5 of the dynamic range parameter θ, respectively. It turns out that the noise power is $\epsilon^2 = m\sigma^2$. The setting of parameters for Algorithm 3 remains the same. For DADM, we choose parameters $\beta = \frac{\|b\|_1}{m2^1}, \frac{\|b\|_1}{m2^3}, \frac{\|b\|_1}{m2^3}$ for dynamic range parameters $\theta = 1, 3, 5$, respectively, for better performance in terms of accuracy and computational cost. For the smoothing parameter μ in the NESTA, we choose the default setting $\mu = \max\{0.1\sigma, 0.01\}$. The stopping criteria for our algorithm and DADM is that the relative errors between the successive iterates of the reconstructed signal should satisfy the inequality $\|u^{k+1} - u^k\|/\|u^k\| < 10^{-5}$. And the stopping criterion for NESTA is $\text{Tol} < 10^{-5}$. The performance of different algorithms are reported in Tables 3 and 4. The accuracy of recovered data from all three algorithms for each data setting is comparable. The computational cost of DADM is comparable to or slightly better than that of Algorithm 3 for data with dynamic range parameters $\theta = 1, 3$. Both of Algorithm 3 and DADM outperform NESTA for data with dynamic range parameters $\theta = 1, 3$ in terms of computational cost. For data with high dynamic range (e.g., $\theta = 5$), Algorithm 3 performs the best in terms of computational cost while DADM performs the worst.

Table 3 Numerical results with partial DCT sensing matrices for noisy data. The number of measurements m is $m = n/4$, and the test signals are s-sparse with $s = 0.02n$. Each value in a cell represents the mean over 50 trials

Method	ℓ_2-error	ℓ_1-error	ℓ_∞-error	CPU time(s)	Iterations
$n = 2^{13}$					
Algorithm 3	6.06e−2	6.28e−3	5.49e−1	0.2309	82
DADM	6.06e−2	6.23e−3	5.49e−1	0.2268	75
NESTA	7.25e−2	2.47e−2	6.68e−1	0.4006	123
$n = 2^{15}$					
Algorithm 3	6.10e−2	6.28e−3	6.15e−1	1.0700	80
DADM	6.10e−2	6.23e−3	6.15e−1	1.0925	76
NESTA	7.23e−2	2.29e−2	7.14e−1	1.7906	123
$n = 2^{13}$					
Algorithm 3	1.90e−2	1.76e−3	10.0453	0.2684	99
DADM	1.89e−2	1.72e−3	10.0370	0.2181	71
NESTA	2.05e−2	1.61e−2	12.0646	0.4353	132
$n = 2^{15}$					
Algorithm 3	1.88e−2	1.60e−3	11.3331	1.5018	111
DADM	1.88e−2	1.54e−3	11.3232	1.0662	71
NESTA	2.09e−2	1.54e−2	13.0586	1.8931	132
$n = 2^{13}$					
Algorithm 3	1.13e−3	1.03e−4	49.7915	0.2740	101
DADM	1.13e−3	5.85e−4	50.3671	0.5953	199
NESTA	1.28e−3	1.13e−3	61.2107	0.4243	125
$n = 2^{15}$					
Algorithm 3	1.18e−3	5.73e−5	56.1854	1.3543	102
DADM	1.18e−3	5.49e−4	56.8402	2.9696	200
NESTA	1.34e−3	1.11e−3	66.0787	1.7721	126

Table 4 Numerical results with partial DCT sensing matrices for noisy data. The number of measurements m is $m = n/8$, and the test signals are s-sparse with $s = 0.01n$. Each value in a cell represents the mean over 50 trials

Method	ℓ_2-error	ℓ_1-error	ℓ_∞-error	CPU time(s)	Iterations
$n = 2^{13}$					
Algorithm 3	1.02e−1	1.94e−2	8.48e−1	0.3296	122
DADM	1.02e−1	1.94e−2	8.48e−1	0.2790	94
NESTA	1.20e−1	3.07e−2	1.0099	0.4606	145
$n = 2^{15}$					
Algorithm 3	1.02e−1	1.83e−2	9.37e−1	1.5691	121
DADM	1.02e−1	1.82e−2	9.37e−1	1.4009	95
NESTA	1.22e−1	2.74e−2	1.1065	2.0506	149
$n = 2^{13}$					
Algorithm 3	2.97e−2	5.30e−3	15.1517	0.2853	102
DADM	2.97e−2	5.21e−3	15.1429	0.3028	99
NESTA	3.10e−2	2.08e−2	17.2106	0.5012	160
$n = 2^{15}$					
Algorithm 3	2.92e−2	5.89e−3	16.8347	1.5609	120
DADM	2.92e−2	5.79e−3	16.8203	1.4675	99
NESTA	3.16e−2	1.93e−2	19.4426	2.2300	160
$n = 2^{13}$					
Algorithm 3	1.94e−3	2.87e−4	75.5390	0.3231	115
DADM	1.92e−3	3.49e−4	75.4992	0.6975	230
NESTA	1.93e−3	1.50e−3	90.2023	0.4981	157
$n = 2^{15}$					
Algorithm 3	1.89e−3	2.00e−4	86.3350	1.5025	114
DADM	1.88e−3	2.06e−4	86.2110	3.3468	233
NESTA	2.03e−3	1.41e−3	99.4225	2.2662	158

4.1.2 Numerical comparison with partial DWHT sensing matrices

The performance of the three algorithms will be discussed in this part. The performance of the algorithms will be presented in a different manner from the previous part with partial DCT sensing matrices. In all of those three algorithms, the computational cost is mainly attributed to the matrix-vector multiplication involving A or $A^\top$. Under the assumption that $AA^\top = I$, the three algorithms only have two such multiplications, one involving A and the other involving $A^\top$ in each iteration. Hence, we will only use the number of iterations to represent the computational cost. For the accuracy, only the relative ℓ_2- error will be selected. The setting of parameters of the three algorithms remains almost the same except that

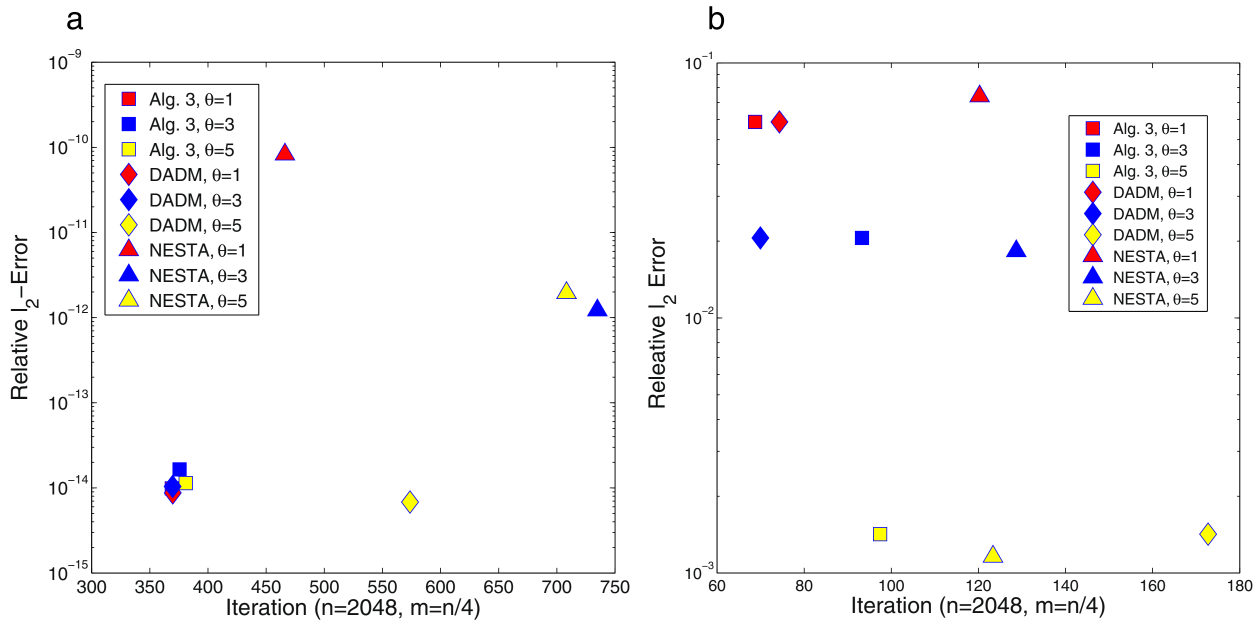

Fig. 3 Numerical results with partial DWHT sensing matrices. The relative ℓ_2 errors (the *vertical axis* with base 10 logarithmic scale) versus the iteration consumed (the *horizontal axis*) for **a** the noise-free case and **b** the noise case (the right). The colors *red*, *blue*, and *yellow* represent the dynamic ranges of the tested signals with θ being 1, 3, and 5, respectively

$\mu = 10^{-8}$, Tol $= 10^{-13}$ is used in NESTA for noise-free data with dynamic range parameter $\theta = 5$.

Figure 3 shows the results of Algorithm 3, DADM, and NESTA when the dimension of the tested signals n is 2048 and the number of measurements m is $n/4$. The symbols "□," "◇," and "∇" denote the results produced by Algorithm 3, DADM, and NESTA, respectively. The colors "red," "blue," and "yellow" represent the dynamic ranges of the tested signals with θ being 1, 3, and 5, respectively. The relative ℓ_2-error is displayed with a base 10

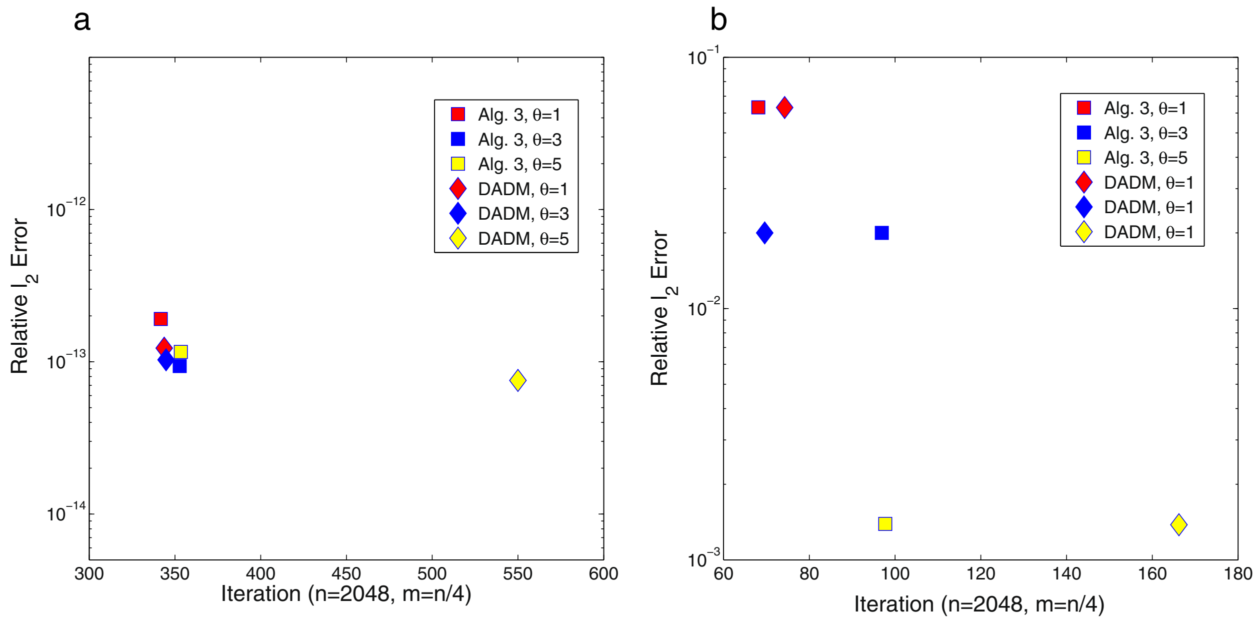

Fig. 4 Numerical results with Gaussian sensing matrices whose rows are orthonormalized. The relative ℓ_2 errors (the *vertical axis* with base 10 logarithmic scale) versus the iteration consumed (the *horizontal axis*) for **a** the noise-free case and **b** the noise case. The colors *red*, *blue*, and *yellows* represent the dynamic ranges of the tested signals with θ being 1, 3, and 5, respectively

Table 5 Numerical results with Gaussian measurement matrices for noise-free data. The test signals have size $n = 4096$. Each value in a cell represents the mean over 50 trials

m	s	θ	ℓ_2-error	ℓ_1-error	ℓ_∞-error	CPU time(s)	Iterations
$n/4$	$0.02n$	1	1.64e−13	2.31e−14	2.07e−12	6.0768	844
$n/8$	$0.01n$	1	3.28e−13	4.93e−14	3.70e−12	5.1040	1305
$n/4$	$0.02n$	3	1.32e−13	3.13e−14	9.93e−11	5.7287	799
$n/8$	$0.01n$	3	3.93e−13	1.06e−13	2.61e−10	4.9531	1255
$n/4$	$0.02n$	5	1.08e−13	2.55e−14	5.76e−09	5.3643	752
$n/8$	$0.01n$	5	4.01e−13	1.25e−13	2.10e−08	4.5700	1157

logarithmic scale plot for the vertical axis. We see clearly that performance of the three algorithms follows the similar phenomena that was seen in the numerical results with partial DCT measurements. The same conclusions can be drown for the results with $m = n/8$ as well.

4.1.3 Numerical comparison with orthonormal Gaussian sensing matrices

In this part, the comparisons of numerical results with orthonormal Gaussian sensing matrices will be shown. Due to the unavailability of source code of NESTA for such sensing matrices, only the comparison between Algorithm 3 and DADM is provided. The setting of parameters of Algorithms 3 and DADM are the same as above except that the stopping criterion Tol $= 10^{-14}$ is used for noise-free data. The numerical result is reported in Fig. 4 in the same manner as in previous part with DWHT sensing matrices. For the noise-free data or noisy data, the performance of Algorithm 3 and DADM is comparable in terms of relative ℓ_2 error. For noise-free data, the performance of Algorithm 3 and DADM is comparable only for data with dynamic range parameters $\theta = 1,\ 3$ in terms of computational cost; while Algorithm 3 outperforms DADM in terms of computational cost (e.g., iteration) for data with $\theta = 5$. For noisy data and in terms of computational cost, performance of the two algorithms is comparable for $\theta = 1$; DADM performs slightly better than Algorithm 3 for $\theta = 3$; and Algorithm 3 outperforms DADM for $\theta = 5$. The same conclusions can be drawn for the results with $m = n/8$ as well.

4.2 Simulation with Gaussian sensing matrices

In this subsection, we only focus on the simulation of Algorithm 3 for data generated by general Gaussian sensing matrices (e.g., rows are not orthonormal), that is, $AA^\top \neq I$. In such scenario, we do not compare Algorithm 3 with NESTA and DADM since the available source code of NESTA does not apply, and DADM needs an inner loop in each of iteration. The setting of parameters for Algorithm 3 is the same as the setting for data with orthonormal Gaussian sensing matrices. The results for noise-free data and noisy data are reported in Tables 5 and 6, respectively. It can be seen that the underlying signal can be recovered with high accuracy for noise-free data and with reasonable high accuracy for noisy data.

5 Conclusions

We reformulated the ℓ_1-norm minimization problems (BP) and (BP$_\epsilon$) via indicator functions as unconstrained minimization problems. The objective function for each unconstrained problem is the sum of the ℓ_1-norm of the underlying signal u and the indicator function of a set in $\mathbb{R}^m$, which is $\{0\}$ for (BP) or the ϵ-ball for (BP$_\epsilon$), composing with the affine transformation $Au - b$. Due to the structure of this objective function and the availability of the explicit forms of the proximity operators for both the ℓ_1-norm and the indicator function, an accurate and efficient algorithm is developed for recovering sparse signals based on fixed-point equation. The algorithm outperforms NESTA in terms of the relative ℓ_2, the relative ℓ_1, and the absolute ℓ_∞ error measures as well as

Table 6 Numerical results with Gaussian measurement matrices for noisy data. The test signals have size $n = 4096$. Each value in a cell represents the mean over 50 trials

m	s	θ	ℓ_2-error	ℓ_1-error	ℓ_∞-error	CPU time(s)	Iterations
$n/4$	$0.02n$	1	1.01e−3	4.08e−4	8.99e−3	2.3762	315
$n/8$	$0.01n$	1	1.65e−3	5.13e−4	1.29e−2	1.8412	455
$n/4$	$0.02n$	3	3.57e−4	1.68e−4	0.1729	1.2184	160
$n/8$	$0.01n$	3	6.10e−4	2.46e−4	0.2602	0.8990	211
$n/4$	$0.02n$	5	4.51e−05	5.52e−05	2.1526	1.0484	139
$n/8$	$0.01n$	5	6.22e−05	4.56e−05	3.1652	0.7387	165

the computational cost for tested signals ranging from a low dynamic range to a high dynamic range with different sizes. For signal with high dynamic range, the proposed algorithm also outperforms DADM in terms of computational cost but yields comparable accuracy. Further, the proposed algorithms also solve general problems without requiring condition $AA^\top = I$ efficiently and accurately.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

Lixin Shen was partially supported by the US National Science Foundation under grant DMS-1115523 and by an award from National Research Council via the Air Force Office of Scientific Research. Yuesheng Xu was partially supported by the US National Science Foundation under grant DMS-1115523.

Received: 9 April 2015 Accepted: 13 July 2015
Published online: 01 August 2015

References

1. E Candes, J Romberg, T Tao, Stable signal recovery from incomplete and inaccurate measurements. Commun. Pur. Appl. Math. **59**(8), 1207–1223 (2006)
2. E Candes, T Tao, Near optimal signal recovery from random projections: universal encoding strategies? IEEE Trans. Inf. Theory. **52**(12), 5406–5425 (2006)
3. SS Chen, DL Donoho, MA Saunders, Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. **20**, 33–61 (1998)
4. SJ Wright, *Primal-Dual Interior-Point Methods*. (Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997)
5. P Combettes, V Wajs, Signal recovery by proximal forward-backward splitting. Multiscale Model. Simul. A SIAM Interdiscip. J. **4**, 1168–1200 (2005)
6. MAT Figueiredo, SJ Wright, RD Nowak, Gradient projection for sparse reconstruction: applications to compressed sensing and other inverse problems. IEEE J. Selected Topics Signal Process. **1**, 586–597 (2007)
7. A Beck, M Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**, 183–202 (2009)
8. ET Hale, W Yin, Y Zhang, Fixed-point continuation for ℓ_1 minimization: methodology and convergence. SIAM J. Optim. **19**, 1107–1130 (2008)
9. J-F Cai, S Osher, Z Shen, Split Bregman methods and frame based image restoration. Multiscale Model. Simul.: A SIAM Interdiscip. J. **2**, 337–369 (2009)
10. W Yin, S Osher, D Goldfarb, J Darbon, Bregman iterative algorithms for ℓ^1 minimization with applications to compressed sensing. SIAM J. Imaging Sci. **1**, 143–168 (2008)
11. A Chambolle, RA DeVore, N-Y Lee, BJ Lucier, Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. IEEE Trans. Image Process. **7**, 319–335 (1998)
12. R Chan, T Chan, L Shen, Z Shen, Wavelet algorithms for high-resolution image reconstruction. SIAM J. Sci. Comput. **24**(4), 1408–1432 (2003)
13. B Efron, T Hastie, I Johnstone, R Tibshirani, Least angle regression. Ann. Stat. **32**, 407–451 (2004)
14. MR Osborne, B Presnell, BA Turlach, A new approach to variable selection in least squares problems. IMA J. Numeric. Anal. **20**, 389–403 (2000)
15. D Donoho, Y Tsaig, Fast solution of ℓ_1-norm minimization problems when the solution may be sparse. IEEE Trans. Inf. Theory. **54**(11), 4789–4812 (2008)
16. E van den Berg, MP Friedlander, Probing the Pareto frontier for basis pursuit solutions. SIAM J. Scie. Comput. **31**, 890–912 (2008)
17. RT Rockafellar, *Convex Analysis*. (Princeton University Press, Princeton, NJ, 1970)
18. S Becker, J Bobin, E Candes, NESTA: a fast and accurate first-order method for sparse recovery. SIAM J. Imaging Sci. **4**(1), 1–39 (2009)
19. Y Nesterov, Smooth minimization of non-smooth functions. Mathematical Programming, Series A. **103**, 127–152 (2005)
20. Q Li, CA Micchelli, L Shen, Y Xu, A proximity algorithm accelerated by Gauss-Seidel iterations for $\mathrm{L1}/\mathrm{TV}$ denoising models. Inverse Probl. **28**, 095003 (2012)
21. CA Micchelli, L Shen, Y Xu, Proximity algorithms for image models: denoising. Inverse Probl. **27**, 045009–30 (2011)
22. CA Micchelli, L Shen, Y Xu, X Zeng, Proximity algorithms for the L1/TV image denoising model. Adv. Comput. Math. **38**, 401–426 (2013)
23. J Yang, Y Zhang, Alternating direction algorithms for l1-problems in compressive sensing. SIAM J. Scie. Comput. **33**, 250–278 (2011)
24. W Deng, W Yin, On the global and linear convergence of the generalized alternating direction method of multipliers (ADMM). Technical report, UCLA, Center for Applied Mathematics (2012)
25. X Zhang, M Burger, S Osher, A unified primal-dual algorithm framework based on Bregman iteration. J. Sci. Comput. **46**, 20–46 (2011)
26. S Boyd, N Parikh, E Chu, B Peleato, J Eckstein, Distributed optimization and statistical learning via alternating direction method of multipliers. Foundations Trends Mach. Learn. **3**, 1–122 (2010)
27. A Chambolle, T Pock, A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. **40**, 120–145 (2011)