Shen and Suter *EURASIP Journal on Advances in Signal Processing* (2016) 2016:71
DOI 10.1186/s13634-016-0369-4

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

One-bit compressive sampling via ℓ_0 minimization

Lixin Shen[1,2*] and Bruce W. Suter[2]

Abstract

The problem of 1-bit compressive sampling is addressed in this paper. We introduce an optimization model for reconstruction of sparse signals from 1-bit measurements. The model targets a solution that has the least ℓ_0-norm among all signals satisfying consistency constraints stemming from the 1-bit measurements. An algorithm for solving the model is developed. Convergence analysis of the algorithm is presented. Our approach is to obtain a sequence of optimization problems by successively approximating the ℓ_0-norm and to solve resulting problems by exploiting the proximity operator. We examine the performance of our proposed algorithm and compare it with the renormalized fixed point iteration (RFPI) (Boufounos and Baraniuk, 1-bit compressive sensing, 2008; Movahed et al., A robust RFPI-based 1-bit compressive sensing reconstruction algorithm, 2012), the generalized approximate message passing (GAMP) (Kamilov et al., IEEE Signal Process. Lett. 19(10):607–610, 2012), the linear programming (LP) (Plan and Vershynin, Commun. Pure Appl. Math. 66:1275–1297, 2013), and the binary iterative hard thresholding (BIHT) (Jacques et al., IEEE Trans. Inf. Theory 59:2082–2102, 2013) state-of-the-art algorithms for 1-bit compressive sampling reconstruction.

Keywords: 1-bit compressive sensing, ℓ_1 minimization, ℓ_0 minimization, Proximity operator

1 Introduction

Compressive sampling is a recent advance in signal acquisition [1, 2]. It provides a method to reconstruct a sparse signal $x \in \mathbb{R}^n$ from linear measurements

$$y = \Phi x, \tag{1}$$

where Φ is a given $m \times n$ measurement matrix with $m < n$ and $y \in \mathbb{R}^m$ is the measurement vector acquired. The objective of compressive sampling is to deliver an approximation to x from y and Φ. It has been demonstrated that the sparse signal x can be recovered exactly from y if Φ has Gaussian i.i.d. entries and satisfies the restricted isometry property [2]. Moreover, this sparse signal can be identified as a vector that has the smallest ℓ_0-norm among all vectors yielding the same measurement vector y under the measurement matrix Φ.

However, the success of the reconstruction of this sparse signal is based on the assumption that the measurements have infinite bit precision. In realistic settings, the measurements are never exact and must be discretized prior to further signal analysis. In practice, these measurements are quantized, a mapping from a continuous real value to a discrete value over some finite range. As usual, quantization inevitably introduces errors in measurements. The problem of estimating a sparse signal from a set of quantized measurements has been addressed in recent literature. Surprisingly, it has been demonstrated theoretically and numerically that 1-bit per measurement is enough to retain information for sparse signal reconstruction. As pointed out in [3, 4], quantization to 1-bit measurements is appealing in practical applications. First, 1-bit quantizers are extremely inexpensive hardware devices that test values above or below zeros, enabling simple, efficient, and fast quantization. Second, 1-bit quantizers are robust to a number of non-linear distortions applied to measurements. Third, 1-bit quantizers do not suffer from dynamic range issues. Due to these attractive properties of 1-bit quantizers, in this paper, we will develop efficient algorithms for reconstruction of sparse signals from 1-bit measurements.

The 1-bit compressive sampling framework originally introduced in [3] is briefly described as follows. Formally, it can be written as

*Correspondence: lshen03@syr.edu
[1] Department of Mathematics, Syracuse University, Syracuse, NY 13244, USA
[2] Air Force Research Laboratory. AFRL/RITB, Rome, NY 13441-4505, USA

Springer Open

$$y = A(x) := \mathrm{sign}(\Phi x), \tag{2}$$

where the function sign$(\cdot)$ denotes the sign of the variable, element-wise, and zero values are assigned to be $+1$. Thus, the measurement operator A, called a 1-bit scalar quantizer, is a mapping from $\mathbb{R}^n$ to the Boolean cube $\{-1, 1\}^m$. Note that the scale of the signal has been lost during the quantization process. We search for a sparse signal $x^\star$ in the unit ball of $\mathbb{R}^m$ such that the sparse signal $x^\star$ is consistent with our knowledge about the signal and measurement process, i.e., $A(x^\star) = A(x)$.

The problem of reconstructing a sparse signal from its 1-bit measurements is generally non-convex, and therefore it is a challenge to develop an algorithm that can find a desired solution. Nevertheless, since this problem was introduced in [3] in 2008, there are several algorithms that have been developed for attacking it [3, 5–7]. Among those existing 1-bit compressive sampling algorithms, the binary iterative hard thresholding (BIHT) [4] exhibits its superior performance in both reconstruction error and as well as consistency via numerical simulations over the algorithms in [3, 5]. When there are a lot of sign flips in the measurements, a method based on adaptive outlier pursuit for 1-bit compressive sampling was proposed in [7–9]. By formulating 1-bit compressive sampling problem in Bayesian terms, a generalized approximate message passing (GAMP) [10] was developed to the problem of reconstruction from 1-bit measurements. The algorithms in [4, 7] require the sparsity of the desired signal to be given in advance. This requirement, however, is hardly satisfied in practice. By keeping only the sign of the measurements, the magnitude of the signal is lost. The models associated with the aforementioned algorithms seek sparse vectors x satisfying consistency constraints (2) in the unit sphere. As a result, these models are essentially non-convex and non-smooth. In [6], a convex minimization problem is formulated for reconstruction of sparse signals from 1-bit measurements and is solved by linear programming. The details of the above algorithms will be briefly reviewed in the next section.

In this paper, we introduce a new ℓ_0 minimization model over a convex set determined by consistency constraints for 1-bit compressive sampling recovery and develop an algorithm for solving the proposed model. Our model does not require prior knowledge on the sparsity of the signal. Our approach for dealing with our proposed model is to obtain a sequence of optimization problems by successively approximating the ℓ_0-norm and to solve resulting problems by exploiting the proximity operator [11]. Convergence analysis of our algorithm is presented.

This paper is organized as follows. In Section 2, we review and comment current 1-bit compressive sampling models and then introduce our own model by assimilating advantages of existing models. Heuristics for solving the proposed model are discussed in Section 3. Convergence analysis of the algorithm for the model is studied in Section 4. A numerical implementable algorithm for the model is presented in Section 5. The performance of our algorithm is demonstrated and compared with the BIHT, RFPI, LP, and GAMP in Section 6. We present our conclusion in Section 7.

2 Models for one-bit compressive sampling

In this section, we begin with reviewing existing models for reconstruction of sparse signals from 1-bit measurements. After analyzing these models, we propose our own model that assimilates the advantages of the existing ones.

Using matrix notation, the 1-bit measurements in (2) can be equivalently expressed as

$$Y\Phi x \geq 0, \tag{3}$$

where $Y := \mathrm{diag}(y)$ is an $m \times m$ diagonal matrix whose ith diagonal element is the ith entry of y. The expression $Y\Phi x \geq 0$ in (3) means that all entries of the vector $Y\Phi x$ are no less than 0. Hence, we can treat the 1-bit measurements as sign constraints that should be enforced in the construction of the signal x of interest. In what follows, Eq. (3) is referred to as sign constraint or consistency condition, interchangeably.

The optimization model for reconstruction of a sparse signal from 1-bit measurements in [3] is

$$\min \|x\|_1 \quad \text{s.t.} \quad Y\Phi x \geq 0 \quad \text{and} \quad \|x\|_2 = 1, \tag{4}$$

where $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the ℓ_1-norm and the ℓ_2-norm of a vector, respectively. In model (4), the ℓ_1-norm objective function is used to favor sparse solutions, the sign constraint $Y\Phi x \geq 0$ is used to impose the consistency between the 1-bit measurements and the solution, and the constraint $\|x\|_2 = 1$ ensures a nontrivial solution lying on the unit ℓ_2 sphere.

Instead of solving model (4) directly, a relaxed version of model (4)

$$\min \left\{ \lambda \|x\|_1 + \sum_{i=1}^{m} h((Y\Phi x)_i) \right\} \quad \text{s.t.} \quad \|x\|_2 = 1 \tag{5}$$

was proposed in [3]. By employing a variation of the fixed point continuation algorithm in [12], an algorithm, which is called renormalized fixed point iteration (RFPI), was developed for solving model (5) efficiently. Here, λ is a regularization parameter and h is chosen to be the one-sided ℓ_1 (or ℓ_2) function, defined at $z \in \mathbb{R}$ as follows

$$h(z) := \begin{cases} |z| \ \left(\text{or } \frac{1}{2}z^2\right), & \text{if } z < 0; \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

We remark that the one-sided ℓ_2 function was adopted in [3] due to its convexity and smoothness properties that are required by a fixed point continuation algorithm.

In [5], a restricted-step-shrinkage algorithm was proposed for solving model (4). This algorithm is similar in sprit to trust-region methods for nonconvex optimization on the unit sphere and has a provable convergence guarantees.

Binary iterative hard thresholding (BIHT) algorithms were recently introduced for reconstruction of sparse signals from 1-bit measurements in [4]. The BIHT algorithms are developed for solving the following constrained optimization model

$$\min \sum_{i=1}^{m} h((Y\Phi x)_i) \quad \text{s.t.} \quad \|x\|_0 \leq s \quad \text{and} \quad \|x\|_2 = 1, \tag{7}$$

where h is defined by Eq. (6), s is a positive integer, and the ℓ_0-norm $\|x\|_0$ counts the number of non-zero entries in x. Minimizing the objective function of model (7) enforces the consistency condition (3). The BIHT algorithms for model (7) are a simple modification of the iterative thresholding algorithm proposed in [13]. It was shown numerically that the BIHT algorithms perform significantly better than the other aforementioned algorithms in [3, 5] in terms of both reconstruction error as well as consistency. Numerical experiments in [4] further show that the BIHT algorithm with h being the one-sided ℓ_1 function performs better in low noise scenarios while the BIHT algorithm with h being the one-sided ℓ_2 function performs better in high noise scenarios. For the measurements having noise (i.e., sign flips), a robust method for recovering signals from 1-bit measurements using adaptive outlier pursuit was proposed in [7], noise-adaptive renormalized fixed point iterative (NARFPI) was introduced in [8], and noise-adaptive restricted step shrinkage (NARSS) was developed in [9].

The algorithms reviewed above for 1-bit compressive sampling are developed for optimization problems having convex objective functions and non-convex constraints. In [6], a convex optimization program for reconstruction of sparse signals from 1-bit measurements was introduced as follows:

$$\min \|x\|_1 \quad \text{s.t.} \quad Y\Phi x \geq 0 \quad \text{and} \quad \|\Phi x\|_1 = p, \tag{8}$$

where p is any fixed positive number. The first constraint $Y\Phi x \geq 0$ requires that a solution to model (8) should be consistent with the 1-bit measurements. If a vector x satisfies the first constraint, so is ax for all $0 < a < 1$. Hence, an algorithm for minimizing the ℓ_1-norm by only requiring consistency with the measurements will yield the solution x being zero. The second constraint $\|\Phi x\|_1 = p$ is then used to prevent model (8) from returning a zero solution, thus resolves the amplitude ambiguity. By taking the first constraint into consideration, we know that $\|\Phi x\|_1 = \langle y, \Phi x\rangle$; therefore, the second constraint becomes $\langle \Phi^\top y, x\rangle = p$. This confirms that both objective function and constraints of model (8) are convex. It was further pointed out in [6] that model (8) can be cast as a linear program. The corresponding algorithm is referred to as LP. As comparing model (8) with model (4), both the constraint $\|x\|_2 = 1$ in model (4) and the constraint $\|\Phi x\|_1 = p$ in model (8), the only difference between both models, enforce a non-trivial solution. However, as we have already seen, model (8) with the constraint $\|\Phi x\|_1 = p$ can be solved by a computationally tractable algorithm.

Let us further comment on models (7) and (8). First, the sparsity constraint in model (7) is impractical since the sparsity of the underlying signal is unknown in general. Therefore, instead of imposing this sparse constraint, we consider to minimize an optimization model having the ℓ_0-norm as its objective function. Second, although model (8) can be tackled by efficient linear programming solvers and the solution of model (8) preserves the effective sparsity of the underlying signal (see [6]), the solution is not necessarily sparse in general as shown in our numerical experiments (see Section 6). Motivated by the aforementioned models and the associated algorithms, we plan in this paper to reconstruct sparse signals from 1-bit measurements via solving the following constrained optimization model

$$\min \|x\|_0 \quad \text{s.t.} \quad Y\Phi x \geq 0 \quad \text{and} \quad \|\Phi x\|_1 = p, \tag{9}$$

where p is again an arbitrary positive number. This model has the ℓ_0-norm as its objective function and inequality $Y\Phi x \geq 0$ and equality $\|\Phi x\|_1 = p$ as its convex constraints.

We remark that the actual value of p is not important as long as it is positive. More precisely, suppose that $\mathcal{S}$ and $\mathcal{S}^\diamond$ are two sets collecting all solutions of model (9) with $p = 1$ and $p = p^\diamond > 0$, respectively. If $x \in \mathcal{S}$, that is, $Y\Phi x \geq 0$ and $\|\Phi x\|_1 = 1$, then, by denoting $x^\diamond := p^\diamond x$, it can be verified that $\|x^\diamond\|_0 = \|x\|_0$, $Y\Phi x^\diamond \geq 0$, and $\|\Phi x^\diamond\|_1 = p^\diamond$. That indicates $x^\diamond \in \mathcal{S}^\diamond$. Therefore, we have that $p^\diamond \mathcal{S} \subset \mathcal{S}^\diamond$. Conversely, we can show that $\mathcal{S}^\diamond \subset p^\diamond \mathcal{S}$ by reverting above steps. Hence, $p^\diamond \mathcal{S} = \mathcal{S}^\diamond$. Without loss of generality, the positive number p is always assumed to be 1 in the rest part of the paper.

We compare model (7) and our proposed model (9) in the following result.

Proposition 1. *Let $y \in \mathbb{R}^m$ be the 1-bit measurements from an $m \times n$ measurement matrix Φ via Eq. (2) and let s be a positive integer. Assume that the vector $x \in \mathbb{R}^n$ is a solution to model (9). Then, model (7) has the unit vector*

$\frac{x}{\|x\|_2}$ as its solution if $\|x\|_0 \leq s$; otherwise, model (7) can not have a solution satisfying the consistency constraint if $\|x\|_0 > s$.

Proof. Since the vector x is a solution to model (9), then x satisfies the consistency constraint $Y\Phi x \geq 0$. Hence, it, together with definition of h in (6), implies that

$$\sum_{i=1}^{m} h\left(\left(Y\Phi\frac{x}{\|x\|_2}\right)_i\right) = 0.$$

We further note that $\left\|\frac{x}{\|x\|_2}\right\|_0 = \|x\|_0$ and $\left\|\frac{x}{\|x\|_2}\right\|_2 = 1$. Hence, the vector $\frac{x}{\|x\|_2}$ is a solution of model (7) if $\|x\|_0 \leq s$.

On the other hand, if $\|x\|_0 > s$ then all solutions to model (7) do not satisfy the consistency constraint. Suppose this statement is *false*. That is, there exists a solution of model (7), say $x^\sharp$, such that $Y\Phi x^\sharp \geq 0$, $\|x^\sharp\|_0 \leq s$, and $\|x^\sharp\|_2 = 1$ hold. Set $x^\diamond := \frac{x^\sharp}{\|\Phi x^\sharp\|_1}$. Then $\|x^\diamond\|_0 = \|x^\sharp\|_0 \leq s$, $Y\Phi x^\diamond \geq 0$, and $\|\Phi x^\diamond\|_1 = 1$. Since $\|x^\diamond\|_0 < \|x\|_0$, it turns out that x is not a solution of model (9). This contradicts our assumption on the vector x. This completes the proof of the result. □

From Proposition 1, we can see that the sparsity s for model (7) is critical. If s is set too large, a solution to model (7) may not be the sparsest solution satisfying the consistency constraint; if s is set too small, solutions to model (7) cannot satisfy the consistency constraint. In contrast, our model (9) does not require the sparsity constraint used in model (7) and delivers the sparsest solution satisfying the consistency constraint. Therefore, these properties make our model more attractive for 1-bit compressive sampling than the BIHT.

To close this section, we recall an algorithm in [10] for the recovery of signals based on generalized approximate message passing (GAMP). This algorithm exploits the prior statistical information on the signal for estimating the minimum-mean-squared error solution from 1-bit measurements. The performance of GAMP will be included in our numerical section.

3 An algorithm for 1-bit compressive sampling

In this section, we will develop algorithms for the proposed model (9). We first reformulate model (9) as an unconstrained optimization problem via the indicator function of a closed convex set in $\mathbb{R}^{m+1}$. It turns out that the objective function of this unconstrained optimization problem is the sum of the ℓ_0-norm and the indicator function composing with a matrix associated with the 1-bit measurements. Instead of directly solving the unconstrained optimization problem, we use some smooth concave functions to approximate the ℓ_0-norm and then linearize the concave functions. The resulting model can be viewed as an optimization problem of minimizing a weighted ℓ_1-norm over the closed convex set. The solution of this resulting model is served as a new point at which the concave functions will be linearized. This process is repeatedly performed until a certain stopping criterion is met. Several concrete examples for approximating the ℓ_0-norm are provided at the end of this section.

We begin with introducing our notation and recalling some background from convex analysis. For the d-dimensional Euclidean space $\mathbb{R}^d$, 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, ι_C is in $\Gamma_0(\mathbb{R}^d)$ for any closed nonempty convex set C.

Next, we reformulate model (9) as an unconstrained optimization problem. To this end, from the $m \times n$ matrix Φ and the m-dimensional vector y in Eq. (2), we define an $(m+1) \times n$ matrix

$$B := \begin{bmatrix} \mathrm{diag}(y) \\ y^\top \end{bmatrix} \Phi \tag{10}$$

and a subset of $\mathbb{R}^{m+1}$

$$\mathcal{C} := \{z : z_{m+1} = 1 \text{ and } z_i \geq 0,\ i = 1, 2, \ldots, m\}, \tag{11}$$

respectively. Then, a vector x satisfies the two constraints of model (9) if and only if the vector Bx lies in the set $\mathcal{C}$. Hence, model (9) can be rewritten as

$$\min\{\|x\|_0 + \iota_{\mathcal{C}}(Bx) : x \in \mathbb{R}^n\}. \tag{12}$$

Problem (12) is known to be NP-complete due to the non-convexity of the ℓ_0-norm. Thus, there is a need for an algorithm that can pick the sparsest vector x satisfying the relation $Bx \in \mathcal{C}$. To attack this ℓ_0-norm optimization problem, a common approach that appeared in recent literature is to approximate the ℓ_0-norm by its computationally feasible approximations. In the context of compressed sensing, we review several popular choices for defining the ℓ_0-norm as the limit of a sequence. More precisely, for a positive number $\epsilon \in (0, 1)$, we consider separable concave functions of the form

$$F_\epsilon(x) := \sum_{i=1}^{n} f_\epsilon(|x_i|), \quad x \in \mathbb{R}^n, \tag{13}$$

where $f_\epsilon : \mathbb{R}_+ \to \mathbb{R}$ is strictly increasing, concave, and twice continuously differentiable such that

$$\lim_{\epsilon \to 0+} F_\epsilon(x) = \|x\|_0, \quad \text{for all} \quad x \in \mathbb{R}^n. \tag{14}$$

The parameter ϵ plays a role of determining the quality of the approximation $F_\epsilon(x)$ to $\|x\|_0$. Since the function f_ϵ is concave and smooth on $\mathbb{R}_+ := [0, \infty)$, it can be majorized by a simple function formed by its first-order Taylor series expansion at a arbitrary point. Write $\mathcal{F}_\epsilon(x, v) := F_\epsilon(v) + \langle \nabla F_\epsilon(|v|), |x| - |v| \rangle$. Therefore, at any point $v \in \mathbb{R}^n$, the following inequality holds

$$F_\epsilon(x) < \mathcal{F}_\epsilon(x, v) \tag{15}$$

for all $x \in \mathbb{R}^n$ with $|x| \neq |v|$. Here, for a vector u, we use $|u|$ to denote a vector such that each element of $|u|$ is the absolute value of the corresponding element of u. Clearly, when v is close enough to x, $\mathcal{F}_\epsilon(x, v)$ the expression on the right-hand side of (15) provides a reasonable approximation to the one on its left-hand side. Therefore, it is considered as a computationally feasible approximation to the ℓ_0-norm of x. With such an approximation, a simplified problem is solved and its solution is used to formulate another simplified problem which is closer to the ideal problem (12). This process is then repeated until the solutions to the simplified problems become stationary or meet a termination criteria. This procedure is summarized in Algorithm 1.

Algorithm 1 (Iterative scheme for model (12))

```
Initialization: choose $\epsilon \in (0, 1)$ and let $x^{(0)} \in \mathbb{R}^n$ be an initial point.
repeat($k \geq 0$)
    Step 1: Compute $x^{(k+1)}$:
        $x^{(k+1)} \in \operatorname{argmin}\left\{\mathcal{F}_\epsilon(x, |x^{(k)}|) + \iota_C(Bx) : x \in \mathbb{R}^n\right\}$.
until a given stopping criteria is met
```

The terms $F_\epsilon(|x^{(k)}|)$ and $\langle \nabla F_\epsilon(|x^{(k)}|), |x^{(k)}| \rangle$ that appear in the optimization problem in Algorithm 1 can be ignored because they are irrelevant to the optimization problem. Hence, the expression for $x^{(k+1)}$ in Algorithm 1 can be simplified as

$$x^{(k+1)} \in \operatorname{argmin}\left\{\langle \nabla F_\epsilon(|x^{(k)}|), |x| \rangle + \iota_C(Bx) : x \in \mathbb{R}^n\right\}. \tag{16}$$

Since f_ϵ is strictly concave and increasing on $\mathbb{R}_+$, f'_ϵ is positive on $\mathbb{R}_+$. Hence, $\langle \nabla F_\epsilon(|x^{(k)}|), |x| \rangle = \sum_{i=1}^n f'_\epsilon(|x_i^{(k)}|)|x_i|$ can be viewed as the weighted ℓ_1-norm of x having $f'_\epsilon(|x_i^{(k)}|)$ as its ith weight. Thus, the objective function of the above optimization problem is convex. Details for finding a solution to the problem will be presented in the next section.

In the rest of this section, we list two possible choices of the functions in (13), namely, the Mangasarian function in [14] and the Log-Det function in [15]. Many other choices can be found from [16–21] and the references therein.

The Mangasarian function is given as follows:

$$F_\epsilon(x) = \sum_{i=1}^n \left(1 - e^{-|x_i|/\epsilon}\right), \tag{17}$$

where $x \in \mathbb{R}^n$. This function is used to approximate the ℓ_0-norm to obtain minimum-support solutions (that is, solutions with as many components equal to zero as possible). The usefulness of the Mangasarian function was demonstrated in finding sparse solutions of underdetermined linear systems (see [22]).

The Log-Det function is defined as

$$F_\epsilon(x) = \sum_{i=1}^n \frac{\log(|x_i|/\epsilon + 1)}{\log(1/\epsilon)}, \tag{18}$$

where $x \in \mathbb{R}^n$. Notice that $\|x\|_0$ is equal to the rank of the diagonal matrix $\operatorname{diag}(x)$. The function $F_\epsilon(x)$ is equal to $(\log(1/\epsilon))^{-1} \log(\det(\operatorname{diag}(x) + \epsilon I)) + n$, the logarithm of the determinant of the matrix $\operatorname{diag}(x) + \epsilon I$. Hence, it was named as the Log-Det heuristic and used for minimizing the rank of a positive semidefinite matrix over a convex set in [15]. Constant terms can be ignored since they will not affect the solution of the optimization problem (16). Hence, the Log-Det function in (18) can be replaced by

$$F_\epsilon(x) = \sum_{i=1}^n \log(|x_i| + \epsilon). \tag{19}$$

We point it out that the Mangasarian function is bounded by 1; therefore, it is non-coercive while the Log-Det function is coercive. This makes a difference in convergence analysis of the associated Algorithm 1 that will be presented in the next section. In what follows, the function F_ϵ is the Mangasarian function or the Log-Det function. We specify it only when it is noted.

4 Convergence analysis

In this section, we shall give convergence analysis for Algorithm 1. We begin with presenting the following result.

Theorem 2. *Given $\epsilon \in (0, 1)$, $x^{(0)} \in \mathbb{R}^n$, and the set $\mathcal{C}$ defined by (11), let the sequence $\{x^{(k)} : k \in \mathbb{N}\}$ be generated by Algorithm 1, where $\mathbb{N}$ is the set of all natural numbers. Then the following three statements hold:*

(i) The sequence $\{F_\epsilon(x^{(k)}) : k \in \mathbb{N}\}$ converges when F_ϵ is corresponding to the Mangasarian function (17) or the Log-Det function (19);

(ii) The sequence $\{x^{(k)} : k \in \mathbb{N}\}$ is bounded when F_ϵ is the Log-Det function;

(iii) $\sum_{k=1}^{+\infty} \left\| |x^{(k+1)}| - |x^{(k)}| \right\|_2^2$ is convergent when the sequence $\{x^{(k)} : k \in \mathbb{N}\}$ is bounded.

Proof. We first prove item (i). The key step for proving it is to show that the sequence $\{F_\epsilon(x^{(k)}) : k \in \mathbb{N}\}$ is decreasing and bounded below. The boundedness of the sequence is due to the fact that $F_\epsilon(0) \leq F_\epsilon(x^{(k)})$. From Step 1 of Algorithm 1 or Eq. (16), one can immediately have that

$$\iota_C(Bx^{(k+1)}) = 0$$

and

$$\langle \nabla F_\epsilon(|x^{(k)}|), |x^{(k+1)}|\rangle \leq \langle \nabla F_\epsilon(|x^{(k)}|), |x^{(k)}|\rangle. \tag{20}$$

By identifying $x^{(k)}$ and $x^{(k+1)}$, respectively, as v and x in (15) and using the inequality in (20), we get $F_\epsilon(x^{(k+1)}) \leq F_\epsilon(x^{(k)})$. Hence, the sequence $\{F_\epsilon(x^{(k)}) : k \in \mathbb{N}\}$ is decreasing and bounded below. Item (i) follows immediately.

When F_ϵ is chosen as the Log-Det function, the coerciveness of F_ϵ together with item (i) implies that the sequence $\{x^{(k)} : k \in \mathbb{N}\}$ must be bounded, that is, item (ii) holds.

Finally, we prove item (iii). Denote $w^{(k)} := |x^{(k+1)}| - |x^{(k)}|$. From the second-order Taylor expansion of the function F_ϵ at $x^{(k)}$, we have that

$$F_\epsilon(x^{(k+1)}) = \mathcal{F}_\epsilon(x^{(k+1)}, x^{(k)}) + \frac{1}{2}(w^{(k)})^\top \nabla^2 F_\epsilon(v) w^{(k)}, \tag{21}$$

where v is some point in the line segment linking the points $|x^{(k+1)}|$ and $|x^{(k)}|$ and $\nabla^2 F_\epsilon(v)$ is the Hessian matrix of F_ϵ at the point v.

By (20), the first term on the right hand of Eq. (21) is less than $F_\epsilon(x^{(k)})$. By Eq. (19), $\nabla^2 F_\epsilon(v)$ for v lying in the first octant of $\mathbb{R}^n$ is a diagonal matrix and is equal to $-\frac{1}{\epsilon^2}\mathrm{diag}(e^{-\frac{v_1}{\epsilon}}, e^{-\frac{v_2}{\epsilon}}, \dots, e^{-\frac{v_n}{\epsilon}})$ or $-\mathrm{diag}((v_1+\epsilon)^{-2}, (v_2+\epsilon)^{-2}, \dots (v_n+\epsilon)^{-2})$ which corresponds to F_ϵ being the Mangasarian or the Log-Det function. Hence, the matrix $\nabla^2 F_\epsilon(v)$ is negative definite. Since the sequence $\{x^{(k)} : k \in \mathbb{N}\}$ is bounded, there exists a constant $\rho > 0$ such that

$$(w^{(k)})^\top \nabla^2 F_\epsilon(v) w^{(k)} \leq -\rho \|w^{(k)}\|_2^2.$$

Putting all above results together into (21), we have that

$$F_\epsilon(x^{(k+1)}) \leq F_\epsilon(x^{(k)}) - \frac{\rho}{2}\left\| |x^{(k+1)}| - |x^{(k)}| \right\|_2^2.$$

Summing the above inequality from $k = 1$ to $+\infty$ and using item (i) we get the proof of item (iii). □

From item (iii) of Theorem 2, we have $\left\| |x^{(k+1)}| - |x^{(k)}| \right\|_2 \to 0$ as $k \to \infty$.

To further study properties of the sequence $\{x^{(k)} : k \in \mathbb{N}\}$ generated by Algorithm 1, the matrix $B^\top$ is required to have the range space property (RSP) which is originally introduced in [23]. With this property and motivated by the work in [23], we prove that Algorithm 1 can yield a sparse solution for model (12).

Prior to presenting the definition of the RSP, we introduce the notation to be used throughout the rest of this paper. Given a set $S \subset \{1, 2, \dots, n\}$, the symbol $|S|$ denotes the cardinality of S, and $S^c := \{1, 2, \dots, n\} \backslash S$ is the complement of S. Recall that for a vector u, by abuse of notation, we also use $|u|$ to denote the vector whose elements are the absolute values of the corresponding elements of u. For a given matrix A having n columns, a vector u in $\mathbb{R}^n$, and a set $S \subset \{1, 2, \dots, n\}$, we use the notation A_S to denote the submatrix extracted from A with column indices in S and u_S the subvector extracted from u with component indices in S.

Definition 3 (Range space property (RSP)). *Let A be an $m \times n$ matrix. Its transpose $A^\top$ is said to satisfy the* range space property (RSP) *of order K with a constant $\rho > 0$ if for all sets $S \subseteq \{1, \dots, n\}$ with $|S| \geq K$ and for all ξ in the range space of $A^\top$ the following inequality holds*

$$\|\xi_{S^c}\|_1 \leq \rho \|\xi_S\|_1.$$

The range space property states that the range of the matrix $A^\top$ contains no vectors where some entries have a significantly larger magnitude with respect to the others. We remark that if the transpose of an $m \times n$ matrix A has the RSP of order K with a constant $\rho > 0$, then for every non-empty set $S \subseteq \{1, \dots, n\}$, the transpose of the matrix A_S, denoted by $A_S^\top$, has the RSP of order K with constant ρ as well. We further remark that there is a relationship (see Proposition 3.6 in [23]) between the RSP and the restricted isometry property (RIP) and null space property (NSP) of A which have been widely used in the compressive sensing literature. For example, if we have a matrix satisfying the NSP or RIP, we may construct a matrix satisfying the RSP. Unfortunately, similar to the RIP and the NSP, the RSP is hard to verify in practice.

The next result shows that if the transpose of the matrix B in Algorithm 1 possesses the RSP, then Algorithm 1 can lead to a sparse solution for model (12). To this end, we define a mapping $\sigma : \mathbb{R}^d \to \mathbb{R}^d$ such that the ith component of the vector $\sigma(u)$ is the ith largest component of $|u|$.

Proposition 4. *Let B be the $(m+1) \times n$ matrix be defined by (10) and let $\{x^{(k)} : k \in \mathbb{N}\}$ be the sequence generated by Algorithm 1. Assume that the matrix $B^\top$ has the RSP of order K with $\rho > 0$ satisfying $(1 + \rho)K < n$. Suppose that the sequence $\{x^{(k)} : k \in \mathbb{N}\}$ is bounded. Then, $(\sigma(x^{(k)}))_n$ the nth largest component of $x^{(k)}$ converges to 0.*

Proof. Suppose this proposition is *false*. Then there exist a constant $\gamma > 0$ and a subsequence $\{x^{(k_j)} : j \in \mathbb{N}\}$ such that $(\sigma(x^{(k_j)}))_n \geq 2\gamma > 0$ for all $j \in \mathbb{N}$. From item (iii) of Theorem, 2 we have that

$$(\sigma(x^{(k_j+1)}))_n \geq \gamma \tag{22}$$

for all sufficiently large j. For simplicity, we set $y^{(k_j)} := \nabla F_\epsilon(|x^{(k_j)}|)$. Hence, by inequality (22) and F_ϵ, we know that

$$|x^{(k_j)}| > 0 \quad |x^{(k_j+1)}| > 0, \quad \text{and} \quad y^{(k_j)} > 0 \tag{23}$$

for all sufficient large j. In what follows, we assume that the integer j is large enough such that the above inequalities in (23) hold.

Since the vector $x^{(k_j+1)}$ is obtained through step 1 of Algorithm 1, i.e., Eq. (16), then by Fermat's rule and the chain rule of subdifferential, we have that

$$0 = \operatorname{diag}(y^{(k_j)})\partial\|\cdot\|_1(\operatorname{diag}(y^{(k_j)})x^{(k_j+1)}) + B^\top b^{(k_j+1)},$$

where $b^{(k_j+1)} \in \partial\iota_C(Bx^{(k_j+1)})$. By (23), we get

$$\partial\|\cdot\|_1(\operatorname{diag}(y^{(k_j)})x^{(k_j+1)}) = \{\operatorname{sgn}(x^{(k_j+1)})\},$$

where $\operatorname{sgn}(\cdot)$ denotes the sign of the variable element-wise. Thus

$$y^{(k_j)} = |\xi^{(k_j+1)}|,$$

where $\xi^{(k_j+1)} = B^\top b^{(k_j+1)}$ is in the range of $B^\top$.

Let S be the set of indices corresponding to the K smallest components of $|\xi^{(k_j+1)}|$. Hence,

$$\sum_{i=1}^{n-K}(\sigma(y^{(k_j)}))_i = \|\xi_{S^c}^{(k_j+1)}\|_1$$

and

$$\sum_{i=n-K+1}^{n}(\sigma(y^{(k_j)}))_i = \|\xi_S^{(k_j+1)}\|_1.$$

Since $B^\top$ has the RSP of order K with the constant ρ, we have that $\|\xi_{S^c}^{(k_j+1)}\|_1 \leq \rho\|\xi_S^{(k_j+1)}\|_1$. Therefore,

$$\sum_{i=1}^{n-K}(\sigma(y^{(k_j)}))_i \leq \rho \sum_{i=n-K+1}^{n}(\sigma(y^{(k_j)}))_i. \tag{24}$$

However, by the definition of σ, we have that

$$\sum_{i=1}^{n-K}(\sigma(y^{(k_j)}))_i \geq (n-K)(\sigma(y^{(k_j)}))_{n-K+1}$$

and

$$\sum_{i=n-K+1}^{n}(\sigma(y^{(k_j)}))_i \leq K(\sigma(y^{(k_j)}))_{n-K+1}.$$

These inequalities together with the condition $(1+\rho)K < n$ lead to

$$\sum_{i=1}^{n-K}(\sigma(y^{(k_j)}))_i > \rho \sum_{i=n-K+1}^{n}(\sigma(y^{(k_j)}))_i,$$

which contradicts to (24). This completes the proof of the proposition. □

From Proposition 4, we conclude that a sparse solution is guaranteed via Algorithm 1 if the transpose of B satisfies the RSP. Next, we answer how sparse this solution will be. To this end, we introduce some notation and develop a technical lemma. For a vector $x \in \mathbb{R}^d$, we denote by $\tau(x)$ the set of the indices of non-zero elements of x, i.e., $\tau(x) := \{i : x_i \neq 0\}$. For a sequence $\{x^{(k)} : k \in \mathbb{N}\}$, a positive number μ, and an integer k, we define $I_\mu(x^{(k)}) := \{i : |x_i^{(k)}| \geq \mu\}$.

Lemma 5. *Let B be the $(m+1) \times n$ matrix defined by (10), let F_ϵ be the Log-Det function defined by (19), and let $\{x^{(k)} : k \in \mathbb{N}\}$ be the sequence generated by Algorithm 1. Assume that the matrix $B^\top$ has the RSP of order K with $\rho > 0$ satisfying $(1+\rho)K < n$. If there exist $\mu > \rho\epsilon n$ such that $|I_\mu(x^{(k)})| \geq K$ for all sufficient large k, then there exists a $k'' \in \mathbb{N}$ such that $\|x^{(k)}\|_0 < n$ and $\tau(x^{(k+1)}) \subseteq \tau(x^{(k'')})$ for all $k > k''$.*

Proof. Set $y^{(k)} := \nabla F_\epsilon(|x^{(k)}|)$. Since $x^{(k+1)}$ is a solution to the optimization problem (16), then by Fermat's rule and the chain rule of subdifferential we have that

$$0 \in \operatorname{diag}(y^{(k)})\partial\|\cdot\|_1(\operatorname{diag}(y^{(k)})x^{(k+1)}) + B^\top b^{(k+1)},$$

where $b^{(k+1)} \in \partial\iota_C(Bx^{(k+1)})$. Hence, if $x_i^{(k+1)} \neq 0$, we have that $y_i^{(k)} = |(B^\top b^{(k+1)})_i|$.

For $i \in I_\mu(x^{(k)})$, we have that $|x_i^{(k)}| \geq \mu$ and $y_i^{(k)} = f'_\epsilon(|x_i^{(k)}|) \leq f'_\epsilon(\mu)$ for all $k \in \mathbb{N}$, where $f_\epsilon = \log(\cdot + \epsilon)$. Furthermore, there exist a k' such that $|x_i^{k+1}| > 0$ for $i \in I_\mu(x^{(k)})$ and $k \geq k'$ due to item (iii) in Theorem 2. Thus, we have for all $k \geq k'$

$$\begin{aligned}\sum_{i\in I_\mu(x^{(k)})} |(B^\top b^{(k+1)})_i| &= \sum_{i\in I_\mu(x^{(k)})} y_i^{(k)} \\ &\leq \sum_{i\in I_\mu(x^{(k)})} f'_\epsilon(\mu) \leq W^*,\end{aligned}$$

where $W^* = n\lim_{\epsilon\to 0+} f'_\epsilon(\mu) = \frac{n}{\mu}$ is a positive number dependent on μ.

Now, we are ready to prove $\|x^{(k)}\|_0 < n$ for all $k > k''$. By Proposition 4, we have that $(\sigma(x^{(k)}))_n \to 0$ when $k \to +\infty$. Therefore, there exists an integer $k'' > k'$ such that $|I_\mu(x^{(k)})| \geq K$ and $0 \leq \sigma(x^{(k)}))_n < \min\{\frac{\mu}{\rho n} - \epsilon, \mu\}$ for all $k \geq k''$. Let i_0 be the index such that $|x_{i_0}^{(k'')}| = (\sigma(x^{(k'')}))_n$. We will show that $x_{i_0}^{(k''+1)} = 0$. If this statement is not true, that is, $x_{i_0}^{(k''+1)}$ is not zero, then

$$|(B^\top b^{(k''+1)})_{i_0}| = f'_\epsilon(|x_{i_0}^{(k'')}|) = \frac{1}{|x_{i_0}^{(k'')}| + \epsilon} > \rho W^*. \tag{25}$$

However, since i_0 is not in the set $I_\mu(x^{(k'')})$ and $B^\top$ satisfies the RSP, we have that

$$\begin{aligned}|(B^\top b^{(k''+1)})_{i_0}| &\leq \sum_{i\notin I_\mu(x^{(k'')})} |(B^\top b^{(k''+1)})_i| \\ &\leq \rho \sum_{i\in I_\mu(x^{(k'')})} |(B^\top b^{(k''+1)})_i| \leq \rho W^*,\end{aligned}$$

which contradicts to (25). Hence, we have that $x_{i_0}^{(k''+1)} = 0$ and $|\tau(x^{(k''+1)})| < n$. By replacing k'' by $k''+1$ and repeating this process, we can obtain $x_{i_0}^{(k''+\ell)} = 0$ for all $\ell \in \mathbb{N}$. Therefore, $\|x\|_0 < n$ for all $k > k''$. This process can be also applied to other components satisfying $x_i^{(k''+1)} = 0$. Thus there exists a $k'' \in \mathbb{N}$ such that $\tau(x^{(k)}) \subseteq \tau(x^{(k'')})$ for all $k \geq k''$. □

With Lemma 5, the next result shows that when the transpose of B satisfies the RSP, there exists a cluster point of the sequence generated by Algorithm 1 that is sparse and satisfies the consistency condition.

Theorem 6. *Let B be the $(m+1)\times n$ matrix defined by (10), let F_ϵ be the Log-Det function defined by (19), and let $\{x^{(k)} : k \in \mathbb{N}\}$ be the sequence generated by Algorithm 1. Assume that the matrix $B^\top$ has the RSP of order K with $\rho > 0$ satisfying $(1+\rho)K < n$. Then, there is a subsequence $\{x^{(k_j)} : j \in \mathbb{N}\}$ that converges to a $\lfloor(1+\rho)K\rfloor$-sparse solution, that is $(\sigma(x^{(k_j)}))_{\lfloor(1+\rho)K+1\rfloor} \to 0$ as $j \to +\infty$ and $\epsilon \to 0$.*

Proof. Suppose the theorem is *false*. Then, there exist μ^*, for any $0 < \epsilon^* < \frac{\mu^*}{\rho n}$, there exist a $\epsilon \in (0, \epsilon^*)$ and k' such that $(\sigma(x^{(k)}))_{\lfloor(1+\rho)K+1\rfloor} \geq \mu^*$ for all $k \geq k'$. It implies that for all $k \geq k'$

$$|I_{\mu^*}(x^{(k)})| \geq \lfloor(1+\rho)K+1\rfloor > (1+\rho)K > K. \tag{26}$$

By Lemma 5, there exist a $k'' \geq k'$ such that $\|x^{(k)}\|_0 < n$ and $\tau(x^{(k+1)}) \subseteq \tau(x^{(k'')})$ for all $k \geq k''$. Let $S = \tau(x^{(k'')})$. Thus $x_{S^c}^{(k)} = 0$ for all $k \geq k''$. Therefore, the optimization problem (16) for updating $x^{(k+1)}$ can be reduced to the following one

$$x_S^{k+1} \in \arg\min\{\langle(\nabla F_\epsilon(|x^{(k)}|))_S, u\rangle + \iota((B_S)u) : u \in \mathbb{R}^{|S|}\}. \tag{27}$$

If $|\tau(x^{(k'')})| > |I_{\mu^*}(x^{(k'')})|$, from (26), we have $(1+\rho)K < |S|$. Thus, from Lemma 5 and $B_S^\top$ having RSP with the same parameters, there exist a $k''' > k''$ such that $\tau(x^{(k)}) < \tau(x^{(k'')})$ for all $k \geq k'''$. Therefore, by induction, there must exist a $\tilde{k}$ such that for all $k \geq \tilde{k}$

$$\tau(x^{(k)}) = I_{\mu^*}(x^{(k)}),\ \tau(x^k) \subseteq \tau(x^{(\tilde{k})}).$$

It means that for all $k \geq \tilde{k}$, all the nonzero components of $x^{(k)}$ are bounded below by μ^*. Therefore, for any $k \geq \tilde{k}$, the updating Eq. (16) is reduced by (27) with $S = I_{\mu^*}(x^{(k)})$. From Lemma 4, we get $[\sigma(x^{(k)})]_{|S|} \to 0$ which contradicts with $|x^k_{|S|}| \geq \mu^*$. Therefore, we get this theorem. □

5 An implementation of Algorithm 1

In this section, we describe in detail an implementation of Algorithm 1 and show how to select the parameters of the associated algorithm.

Solving problem (16) is the main issue for Algorithm 1. A general model related to (16) is

$$\min\{\|\Gamma x\|_1 + \varphi(Bx) : x \in \mathbb{R}^n\}, \tag{28}$$

where Γ is a diagonal matrix with positive diagonal elements and φ is in $\Gamma_0(\mathbb{R}^{m+1})$. In particular, if we choose $\Gamma = \nabla F_\epsilon(|x^{(k)}|)$ and $\varphi = \iota_{\mathcal{C}}$, where $x^{(k)}$ is a vector in $\mathbb{R}^n$, ϵ is a positive number, $\mathcal{C}$ is given by (11), and F_ϵ is a function given by (13), then model (28) reduces to the optimization problem in Algorithm 1.

We solve model (28) by using recently developed first-order primal-dual algorithm (see, e.g., [24–26]). To present this algorithm, we need two concepts in convex analysis, namely, the proximity operator and conjugate function. The proximity operator was introduced in [27]. 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 conjugate of$f \in \Gamma_0(\mathbb{R}^d)$ is the function$f^* \in \Gamma_0(\mathbb{R}^d)$ defined at $z \in \mathbb{R}^d$ by

$$f^*(z) := \sup\{\langle x, z\rangle - f(x) : x \in \mathbb{R}^d\}.$$

With these notation, the first-order primal-dual (PD) method for solving (28) is summarized in Algorithm 2 (referred to as PD subroutine).

Algorithm 2 PD subroutine (the first-order primal-dual algorithm for solving (28))

```
Input: the $(m+1) \times n$ matrix $B$ defined by (10); two positive numbers $\alpha$ and $\beta$ satisfying the relation $\alpha\beta < \frac{1}{\|B\|^2}$; the $n \times n$ diagonal matrix $\Gamma$ with all diagonal elements positive; and the function $\varphi \in \Gamma_0(\mathbb{R}^n)$.
Initialization: $i = 0$ and an initial guess $(u^{(-1)}, u^{(0)}, x^{(0)}) \in \mathbb{R}^{m+1} \times \mathbb{R}^{m+1} \times \mathbb{R}^n$
repeat($i \geq 0$)
    Step 1: Compute $x^{(i+1)}$:
        $x^{(i+1)} = \mathrm{prox}_{\alpha\|\cdot\|_1 \circ \Gamma}\left(x^{(i)} - \alpha B^\top (2u^{(i)} - u^{(i-1)})\right)$
    Step 2: Compute $u^{(i+1)}$:
        $u^{(i+1)} = \mathrm{prox}_{\beta\varphi^*}(u^{(i)} + \beta B x^{(i+1)})$
    Step 3: Set $i := i + 1$.
until a given stopping criteria is met and the corresponding vectors $u^{(i)}$, $u^{(i+1)}$, and $x^{(i+1)}$ are denoted by $u^{cur}$, $u^{new}$, and $x^{new}$, respectively.
Output: $(u^{cur}, u^{new}, x^{new}) = \mathrm{PD}(\alpha, \beta, B, \Gamma, \varphi, u^{(-1)}, u^{(0)}, x^{(0)})$
```

Theorem 7. *Let B be an* $(m+1) \times n$ *matrix defined by (10), let* $\mathcal{C}$ *be the set given by (11), let* α *and* β *be two positive numbers, and let L be a positive such that* $L \geq \|B\|^2$, *where* $\|B\|$ *is the largest singular value of B. If*

$$\alpha\beta L < 1,$$

then for any arbitrary initial vector $(x^{-1}, x^0, u^0) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{m+1}$, *the sequence* $\{x^k : k \in \mathbb{N}\}$ *generated by Algorithm 2 converges to a solution of model (28).*

The proof of Theorem 7 follows immediately from Theorem 1 in [24] or Theorem 3.5 in [25]. We skip its proof here.

Both proximity operators $\mathrm{prox}_{\alpha\|\cdot\|_1\circ\Gamma}$ and $\mathrm{prox}_{\beta\varphi^*}$ should be computed easily and efficiently in order to make the iterative scheme in Algorithm 2 numerically efficient. Indeed, the proximity operator $\mathrm{prox}_{\alpha\|\cdot\|_1\circ\Gamma}$ is given at $z \in \mathbb{R}^n$ as follows: for $j = 1, 2, \ldots, n$

$$\left(\mathrm{prox}_{\alpha\|\cdot\|_1\circ\Gamma}(z)\right)_j = \max\left\{|z_j| - \alpha\gamma_j, 0\right\} \cdot \mathrm{sign}(z_j), \quad (29)$$

where γ_j is the jth diagonal element of Γ. Using the well-known Moreau decomposition (see, e.g., [27, 28])

$$\mathrm{prox}_{\beta\varphi^*} = I - \beta\ \mathrm{prox}_{\frac{1}{\beta}\varphi} \circ \left(\frac{1}{\beta}I\right), \quad (30)$$

we can compute the proximity operator $\mathrm{prox}_{\beta\varphi^*}$ via $\mathrm{prox}_{\frac{1}{\beta}\varphi}$ which depends on a particular form of the function φ. As our purpose is to develop algorithms for the optimization problem in Algorithm 1, we need to compute the proximity operator of $\iota_{\mathcal{C}}^*$ which is given in the following.

Lemma 8. *If* $\mathcal{C}$ *is the set given by (11) and* β *is a positive number, then for* $z \in \mathbb{R}^{m+1}$, *we have that*

$$prox_{\beta\iota_{\mathcal{C}}^*}(z) = (z_1 - (z_1)_+, \ldots, z_m - (z_m)_+, z_{m+1} - \beta), \quad (31)$$

where $(s)_+$ *is s if* $s \geq 0$ *and 0 otherwise.*

Proof. We first give an explicit form for the proximity operator $\mathrm{prox}_{\frac{1}{\beta}\iota_C}$. Note that $\iota_C = \frac{1}{\beta}\iota_C$ for $\beta > 0$ and $\iota_C(z) = \iota_{\{1\}}(z_{m+1}) + \sum_{i=1}^m \iota_{[0,\infty)}(z_i)$, for $z \in \mathbb{R}^{m+1}$. Hence, we have that

$$\mathrm{prox}_{\frac{1}{\beta}\iota_C}(z) = ((z_1)_+, (z_2)_+, \ldots, (z_m)_+, 1), \quad (32)$$

where $(s)_+$ is s if $s \geq 0$ and 0 otherwise. Here we use the facts that $\mathrm{prox}_{\iota_{[0,+\infty)}}(s) = (s)_+$ and $\mathrm{prox}_{\iota_{\{1\}}}(s) = 1$ for any $s \in \mathbb{R}$.

By the Moreau decomposition (30), we have that $\mathrm{prox}_{\beta\iota_{\mathcal{C}}^*}(z) = z - \beta\mathrm{prox}_{\frac{1}{\beta}\iota_C}(\frac{1}{\beta}z)$. This together with Eq. (32) yields (31). □

Next, we comment on the diagonal matrix Γ in model (28). When the function φ in model (28) is chosen to be ι_C, then the relation $a\varphi = \varphi$ holds for any positive number a. Hence, by rescaling the diagonal matrix Γ in model (28) with any positive number, the solutions of model (28) are not altered. Therefore, we can assume that the largest diagonal entry of Γ is always equal to one.

In applications of Theorem 7 as in Algorithm 2, we should make the product of α and β as close to $1/\|B\|^2$ as possible. In our numerical simulations, we always set

$$\alpha = \frac{0.999}{\beta\|B\|^2}. \quad (33)$$

In such a way, β is essentially the only parameter that needs to be determined.

Prior to computing α for a given β by Eq. (33), we need to know the norm of the matrix B. When $\min\{m, n\}$ is small, the norm of the matrix B can be computed directly. When $\min\{m, n\}$ is large, an upper bound of the norm of the matrix B is estimated in terms of the size of B as follows.

Proposition 9. *Let* Φ *be an* $m \times n$ *matrix with i.i.d. standard Gaussian entries and y be an m-dimensional vector with its component being* $+1$ *or* -1. *We define an* $(m+1) \times n$ *matrix B from* Φ *and y via Eq. (10). Then,*

$$\mathbb{E}\{\|B\|\} \leq \sqrt{m+1}(\sqrt{n} + \sqrt{m}).$$

Moreover,

$$\|B\| \le \sqrt{m+1}(\sqrt{n}+\sqrt{m}+t)$$

holds with probability at least $1-2e^{-t^2/2}$ *for all* $t \ge 0$.

Proof. By the structure of the matrix B in (10), we know that

$$\|B\| \le \left\| \begin{bmatrix} \mathrm{diag}(y) \\ y^\top \end{bmatrix} \right\| \cdot \|\Phi\|.$$

Therefore, we just need to compute the norms on the right-hand side of the above inequality. Denote by I_m the $m \times m$ identity matrix and 1_m the vector with all its components being 1. Then,

$$\begin{bmatrix} \mathrm{diag}(y) \\ y^\top \end{bmatrix} \begin{bmatrix} \mathrm{diag}(y) & y \end{bmatrix} = \begin{bmatrix} I_m & 1_m \\ 1_m^\top & m \end{bmatrix},$$

which is a special arrow-head matrix and has $m+1$ as its largest eigenvalue (see [29]). Hence,

$$\left\| \begin{bmatrix} \mathrm{diag}(y) \\ y^\top \end{bmatrix} \right\| = \sqrt{m+1}.$$

Furthermore, by using random matrix theory for the matrix Φ, we know that $\mathbb{E}\{\|\Phi\|\} \le \sqrt{n}+\sqrt{m}$ and $\|\Phi\| \le \sqrt{n}+\sqrt{m}+t$ with probability at least $1-2e^{-t^2/2}$ for all $t \ge 0$ (see, e.g., [30]). This completes the proof of this proposition. □

Let us compute the norm of B numerically for 100 randomly generated matrices Φ and vectors y for the pair (m,n) with three different choices $(500, 1000)$, $(1000, 1000)$, and $(1500, 1000)$, respectively. Corresponding to these choices, the mean values of $\|B\|$ are about 815, 1276, and 1711 while the upper bounds of the expected values of $\|B\|$ by Proposition 9 are about 1208, 2001, and 2726, respectively. We can see that the norm of B varies with its size and turns to be a big number when the value of $\min\{m,n\}$ is relatively large. As a consequence, the parameter α or β must be very small relative to the other by Eq. (33). Therefore, in what follows, the used matrix B in model (28) is considered to have been rescaled in the following way:

$$\frac{B}{\|B\|} \quad \text{or} \quad \frac{B}{\sqrt{m+1}(\sqrt{n}+\sqrt{m})} \tag{34}$$

when the norm of B can be computed easily or not.

The complete procedure for model (12) and how the PD subroutine is employed are summarized in Algorithm 3.

Algorithm 3 (Iterative scheme for model (12))

```
Input: the $(m+1) \times n$ matrix $B$ formed by an $m \times n$ matrix $\Phi$ and an $m$-dimensional vector $y$ via (10); the set $\mathcal{C}$ given by (11); $\epsilon \in (0,1)$, and $\tau > 0$; $\alpha_{\max}$ and $\epsilon_{\min}$ be two real numbers; the maximum iteration number $k_{\max}$.
Initialization: normalizing $B$ according to (34); $\Gamma$ being the $n \times n$ identity matrix; an initial guess $(u^{old_0}, u^{cur_0}, x^{(0)}) \in \mathbb{R}^{m+1} \times \mathbb{R}^{m+1} \times \mathbb{R}^n$; and initial parameters $\beta$ and $\alpha = 0.999/\beta$.
while $k < k_{\max}$ do
    Step 1: Compute
        $(u^{old_{k+1}}, u^{cur_{k+1}}, x^{(k+1)}) = \mathrm{PD}(\alpha, \beta, B, \Gamma, \iota_{\mathcal{C}}, u^{old_k}, u^{cur_k}, x^{(k)})$
    Step 2: Update $\Gamma$ as the scaled matrix $\mathrm{diag}(\nabla F_\epsilon(x^{(k+1)}))$ such that the largest diagonal element of $\Gamma$ is one.
    Step 3: If $\alpha < \alpha_{\max}$, update $\alpha \leftarrow 2\alpha$,  $\beta \leftarrow \beta/2$; if $\epsilon > \epsilon_{\min}$, update $\epsilon \leftarrow \tau\epsilon$;
    Step 4: Update $k \leftarrow k+1$.
end while
Output: $x^{(k_{\max})}$
```

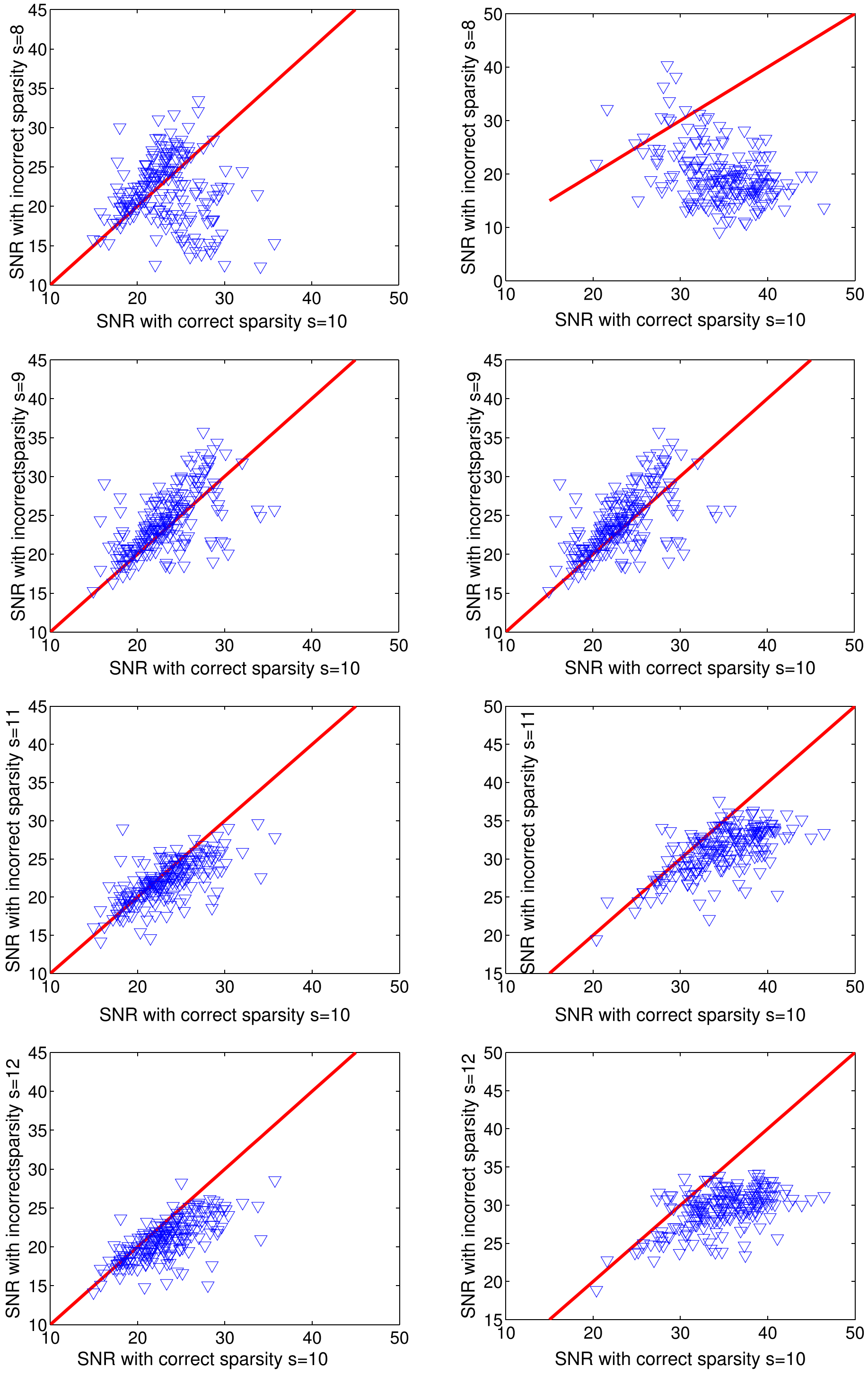
6 Numerical simulations

In this section, we demonstrate the performance of Algorithm 3 for 1-bit compressive sampling reconstruction in terms of accuracy and consistency and compare it with the BIHT, RFPI, LP, and GAMP.

Through this section, all random $m \times n$ matrices Φ and length-n, s-sparse vectors x are generated based on the following assumption: entries of Φ and x on their support are i.i.d. Gaussian random variables with zero mean and unit variances. The locations of the nonzero entries (i.e., the support) of x are randomly permuted. We then generate the 1-bit observation vector y by Eq. (2). We obtain reconstruction of $x^\star$ from y by using either the BIHT, RFPI, LP, GAMP, or Algorithm 3. Four metrics, the signal-to-noise ratio (SNR), the Hamming error, the number of missing nonzero coefficients, and the number of misidentified nonzero coefficients, respectively, are used to evaluate the quality of the reconstruction. More precisely, the signal-to-noise ratio (SNR) in dB is defined as

$$\mathrm{SNR}(x, x^\star) = 20\log_{10}\left(\left\| \frac{x}{\|x\|} \right\|_2 \Big/ \left\| \frac{x}{\|x\|} - \frac{x^\star}{\|x^\star\|} \right\|_2 \right);$$

the Hamming error is $\|y - \mathrm{sign}(\Phi x^\star)\|_0/m$ where m is the number of measurements; the number of missing nonzero coefficients refers to the number of nonzero coefficients that an algorithm “misses,” i.e., determines to be zero; the number of misidentified nonzero coefficients



(a) $m = 500$ (b) $m = 1000$

Fig. 1 The marks in the plots (from the *top row* to the *bottom row*) represent the pairs of the SNR values from the BIHT with the correct sparsity input (i.e., $s = 10$) and incorrect sparsity inputs 8, 9, 11, and 12, respectively. We fix $n = 1000$

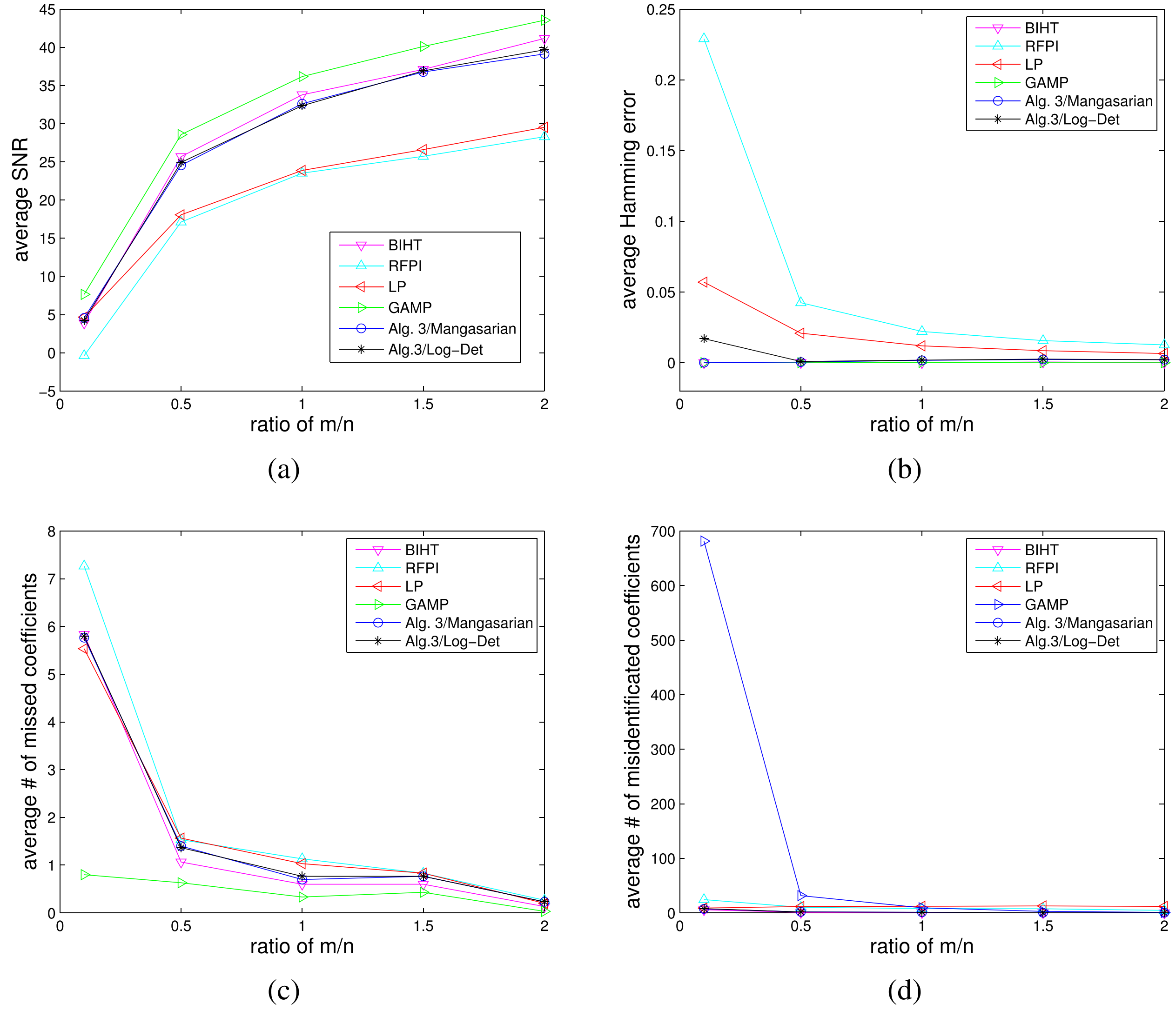
refers to the number of nonzero coefficients that are "misidentified," i.e., coefficients that are determined to be nonzero when they should be zero. The last two metrics measure how well each algorithm finds the signal support, meaning the locations of the nonzero coefficients. A higher value of SNR indicates a better reconstructed signal. The smaller the values of the rest three metrics are the better the reconstructed signals will be. The accuracy of all test algorithms is measured by the average of values of these four metrics over 100 trials unless otherwise noted. For all figures in this section, results by the BIHT, RFPI, LP, GAMP, and Algorithm 3 with the Mangasarian function (17) and the Log-Det function (19) are marked by the symbols "▽," "△," "◁," "▷," "○," and "⋆," respectively.

6.1 Effects of using inaccurate sparsity on the BIHT

The BIHT requires knowing the sparsity of the underlying signals. This requirement is, however, not known in practical applications. In this subsection, we demonstrate through numerical experiments that the mismatched sparsity for a signal will degenerate the performance of the BIHT.

To this end, we fix $n = 1000$ and $s = 10$ and consider two cases of m being 500 and 1000. For each case, we vary the sparsity input for the BIHT from 8 to 12 in which 10 is the only right choice. Therefore, there are total ten configurations. For each configuration, we record the SNR values of the reconstructed signals by the BIHT.

Figure 1 depicts the SNR values of the experiments. The plots in the left column of Fig. 1 are for the case



(a)

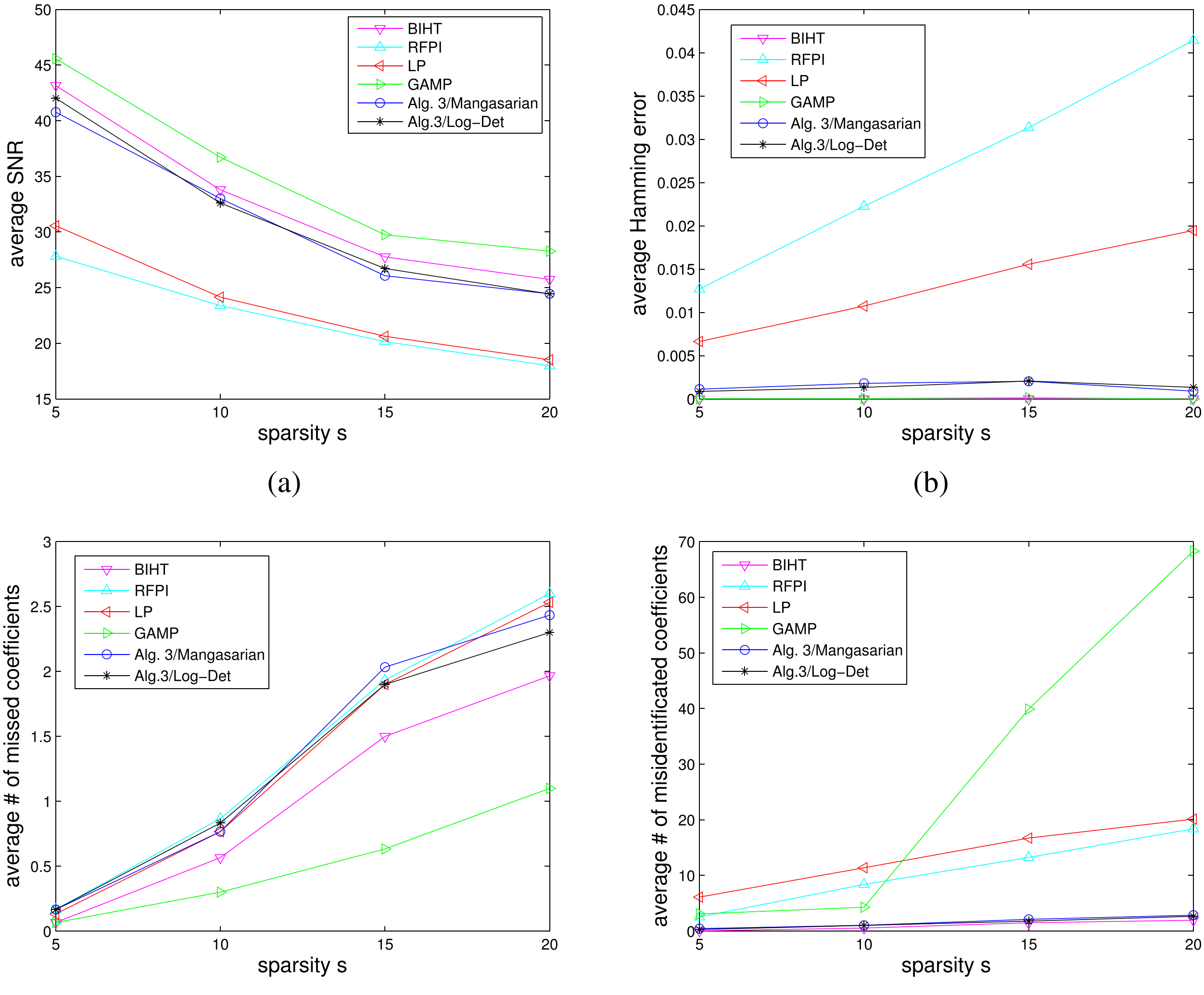
(b)

(c)

(d)

Fig. 2 Algorithm comparison with configuration 1 for fixed $n = 1000$ and $s = 10$. **a** average SNR vs m/n; **b** average Hamming error vs m/n; **c** average number of missing nonzero coefficients vs m/n; and **d** average number of misidentified nonzero coefficients vs m/n

$m = 500$ while the plots in the right column are for the case $m = 1000$. The marks in each plot represent the pairs of the SNR values with the correct sparsity input (i.e., $s = 10$) and with a mismatched sparsity input (i.e., $s = 8$, $s = 9$, $s = 11$, or $s = 12$ corresponding to the row 1, 2, 3, or 4). A mark below the red line indicates that the BIHT with the correct sparsity input works better than the one with an incorrect sparsity input. A mark that is far away from the red line indicates the BIHT with the correct sparsity input works much better than the one with an incorrect sparsity input or vice versa. Except the second plot in the left column, we can see that the BIHT with the correct sparsity input performs better than the one with an inaccurate sparsity input. In particular, when an underestimated sparsity input to the BIHT is used, the performance of the BIHT will be significantly reduced (see the plots in the first two columns of Fig. 1). When an overestimated sparsity input to the BIHT is used, majority of the marks are under the red lines and are relatively closer to the red lines than those from the BIHT with underestimated sparsity input. We further report that the average SNR values for the sparsity input $s = 8$, 9, 10, 11, and 12 for $m = 500$ are 21.89, 24.18, 23.25, 22.10, and 21.00dB, respectively. Similarly, for $m = 1000$, the average SNR values for the sparsity input $s = 8$, 9, 10, 11, and 12 are 19.77dB, 26.37dB, 34.74dB, 31.12dB, and 29.46dB, respectively. In summary, we conclude that a proper chosen sparsity constraint is critical for the success of the BIHT.



(a)

(b)

(c)

(d)

Fig. 3 Algorithm comparison with configuration 2 for fixed $m = 1000$ and $n = 1000$. **a** average SNR vs sparsity; **b** average Hamming error vs sparsity; **c** average number of missing nonzero coefficients vs sparsity; and **d** average number of misidentified nonzero coefficients vs sparsity

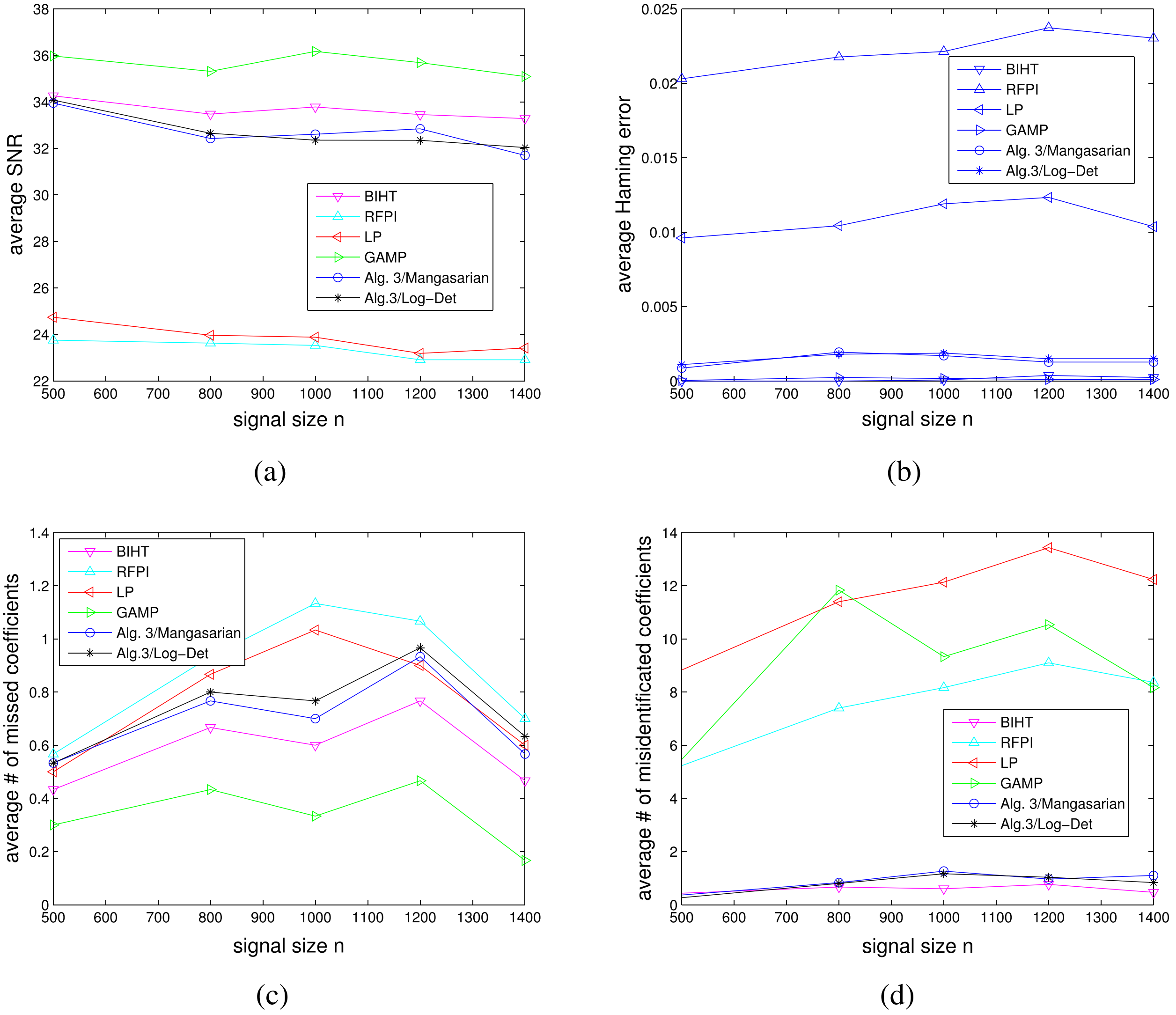
6.2 Performance of Algorithm 3

Prior to applying Algorithm 3 for 1-bit compressive sampling problem, parameters $k_{\max}$, τ, $\alpha_{\max}$, $\epsilon_{\min}$, α, and ϵ in Algorithm 3 need to be determined. Under the aforementioned setting for the random matrix Φ and sparse signal x, we fix $k_{\max} = 13$, $\tau = \frac{1}{2}$, $\alpha_{\max} = 8000$, $\epsilon_{\min} = 10^{-4}$. For the functions F_ϵ defined by (17) and (19), we set the pair of initial parameters (α, ϵ) as $(500, 0.25)$ and $(250, 0.125)$, respectively. The iterative process in the PD subroutine is forced to stop if the corresponding number of iteration exceeds 300. These parameters are used in all simulations performed by Algorithm 3 in the rest of this section.

To evaluate the performance of Algorithm 3 at various scenarios, the following three configurations for n the dimension of the signal, m the number of measurements, and s the sparsity of the vector x, are considered:

- configuration 1: $n = 1000$, $s = 10$, and $m = 100, 500, 1000, 1500$
- configuration 2: $m = 1000$, $n = 1000$, and $s = 5, 10, 15, 20$
- configuration 3: $m = 1000$, $s = 10$, and $n = 500, 800, 1200, 1400$

For every case in each configuration, we compare the accuracy of Algorithm 3 with the BIHT, RFPI, LP, and GAMP by computing the average of values of the four metrics over 100 trials. We remark that Algorithm 3, RFPI, LP, and GAMP do not require the knowledge of sparsity of original signals.



(a)

(b)

(c)

(d)

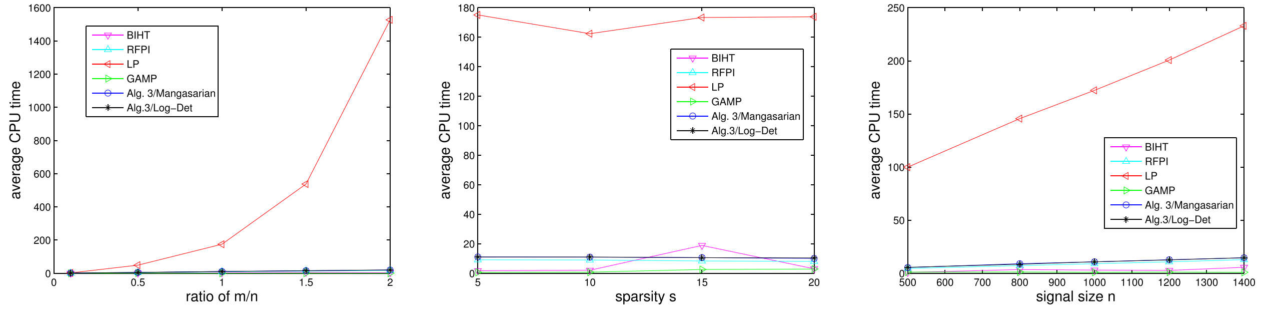
Fig. 4 Algorithm comparison with configuration 3 for fixed $m = 1000$ and $s = 10$. **a** average SNR vs signal size; **b** average Hamming error vs signal size; **c** average number of missing nonzero coefficients vs signal size; and **d** average number of misidentified nonzero coefficients vs signal size

For the first configuration, Fig. 2 displays the average values of the four metrics by the BIHT, RFPI, LP, GAMP, and Algorithm 3 with the Mangasarian function (17) and the Log-Det function (19). Figure 2a demonstrates that the GAMP performs best, the BIHT and Algorithm 3 perform similarly and exhibit much better performance than the LP and RFPI in terms of SNR values. As expected, the SNR value of the reconstruction from each algorithm increases as the number of measurements m increases. Figure 2b depicts the consistency of the algorithms through Hamming error, that is, whether the signs of measurements of the reconstruction are the same as the signs of the original measurements. We can see that the Hamming errors generated by the BIHT, GAMP, and Algorithm 3 decrease towards to zeros as m increases. However, the Hamming errors from the LP and RFPI are always above zero. Figure 2c, d is used to demonstrate how well each algorithm finds the signal support, meaning the locations of the nonzero coefficients. Figure 2c depicts that the number of missed coefficients as a function of the ratio m/n is decreasing. From this plot, we can see that the GAMP performs best and the rest algorithms perform similarly, in particular, when the ratio m/n is larger than 1.5. However, Fig. 2d depicts that the sparsity of the reconstructed signal from GAMP is higher than that from other algorithms. In summary, Algorithm 3 with the Mangasarian function and the Log-Det function performs as equally good as the BIHT in terms of the four metrics, in particular, when m/n is greater than 1, even though our algorithm does not require to know the exact sparsity of the original signal. We can also conclude that Algorithm 3 outperforms the RFPI for all metrics while the GAMP performs better than the other algorithms in terms of the metrics of SNR, the Hamming error, and the number of missing nonzero coefficients.

For the second configuration, the average values of the four metrics as a function of sparsity s are depicted in Fig. 3 for the BIHT, RFPI, LP, GAMP, and Algorithm 3 with fixed $m = 1000$ and $n = 1000$. Figure 3a, b depicts that BIHT, GAMP, and Algorithm 3 outperform the RFPI and LP in terms of values of SNR and the Hamming error. Figure 3c indicates that GAMP performs much better than the other algorithms in terms of the number of missing nonzero coefficients while Fig. 3d indicates that GAMP performs much worse than the other algorithms in terms of the number of misidentified nonzero coefficients.

For the third configuration, the average values of the four metrics as a function of signal size s are depicted in Fig. 4 for the BIHT, RFPI, LP, GAMP, and Algorithm 3 with fixed $m = 1000$ and $s = 10$. The plots in Fig. 4a–c indicate that the GAMP performs best, the BIHT and Algorithm 3 perform similarly and exhibit much better performance than the LP and RFPI in terms of values of SNR, the Hamming error, and the number of missing nonzero coefficients. Figure 4d shows that the BIHT and Algorithm 3 outperform the other algorithms for all tested values of n in terms of the number of misidentified nonzero coefficients.

Finally, we compare the speed of the algorithms by measuring the average CPU time it takes each algorithm to produce the results showed in Figs. 2, 3, and 4. The experiments are performed under Windows 7 and Matlab 7.11 (R2010b) running on a laptop equipped with an Intel Core i5-2520M CPU at 2.50GHz and 4G RAM memory. When we implemented the BIHT, the number of iterations is set to 1500. The MATLAB command `linprog` was adopted in the implementation of LP. The source code of the GAMP was downloaded from the website of the first author of [10]. The source code of the RFPI was provided by the authors of [8]. The RFPI has two loops. The suggested number of outer-loop iterations is 20 while the number of inner-loop iterations is 200. The results of the experiments are depicted in Fig. 5. We find that both BIHT and GAMP are faster than the RFPI and Algorithm 3. The CPU time consumed by LP increases significantly, in particularly, when the size of the signal or the number of measurement increases.



(a) configuration 1 (b) configuration 2 (c) configuration 3

Fig. 5 Speed comparison: average CPU time for **a** configuration 1; **b** configuration 2; and **c** configuration 3

7 Conclusions

In this paper, we proposed a new model and algorithm for 1-bit compressive sensing. The convergence analysis of the proposed algorithm was given. We demonstrated the performance of the algorithm for reconstruction from 1-bit measurements. In the future, it would be of interest to study the convergence of Algorithm 3 with the Mangasarian function. This result would be highly needed to adaptively update all the parameters in Algorithm 3 so that consistent reconstruction can be achieved with improved accuracy.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements
The authors would like to thank Ms. Na Zhang for her valuable comments and insightful suggestions which have brought improvements to several aspects of this manuscript. The authors would also like to thank Dr. Movahed for providing us his MATLAB source codes for the algorithm RFPI in [8].
The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsement, either expressed or implied, of the Air Force Research Laboratory or the US Government.
This research is supported in part by an award from National Research Council via the Air Force Office of Scientific Research and by the US National Science Foundation under grant DMS-1522332.

Received: 7 July 2015 Accepted: 2 June 2016
Published online: 14 June 2016

References

1. E Candes, J Romberg, T Tao, Stable signal recovery from incomplete and inaccurate measurements. Commun. Pure 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. PT Boufounos, RG Baraniuk, in *Proceedings of Conference on Information Science and Systems (CISS)*. 1-bit compressive sensing (IEEE, NJ, 2008), pp. 16–21
4. L Jacques, JN Laska, PT Boufounos, RG Baraniuk, Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. IEEE Trans. Inf. Theory. **59**, 2082–2102 (2013)
5. J Laska, Z Wen, W Yin, R Baraniuk, Trust, but verify: fast and accurate signal recovery from 1-bitcompressive measurements. IEEE Trans. Signal Process. **59**, 5289–5301 (2011)
6. Y Plan, R Vershynin, One-bit compressed sensing by linear programming. Commun. Pure Appl. Math. **66**, 1275–1297 (2013)
7. M Yan, Y Yang, S Osher, Robust 1-bit compressive sensing using adaptive outlier pursuit. IEEE Trans. Signal Process. **60**, 3868–3875 (2012)
8. A Movahed, A Panahi, G Durisi, A robust RFPI-based 1-bit compressive sensing reconstruction algorithm. Information Theory Workshop (ITW), 2012 IEEE, Lausanne, 567–571 (2012). doi:10.1109/ITW.2012.6404739
9. A Movahed, A Panahi, Reed MC, in *IEEE International Conference On Acoustics, Speech, and Signal processing*. Recovering signals with variable sparsity levels from the noisy 1-bit compressive measurements. (IEEE, 2014), pp. 6504–6508
10. U Kamilov, A Bourquard, A Amini, M Unser, One-bit measurements with adaptive thresholds. IEEE Signal Process. Lett. **19**(10), 607–610 (2012)
11. J-J Moreau, Proximité et dualité dans un espace hilbertien. Bull. Soc. Math. France. **93**, 273–299 (1965)
12. ET Hale, W Yin, Y Zhang, Fixed-point continuation for ℓ_1 minimization: methodology and convergence. SIAM J. Optimization. **19**, 1107–1130 (2008)
13. T Blumensath, ME Davies, Iterative hard thresholding for compressed sensing. Appl. Comput. Harmonic Anal. **27**, 265–274 (2009)
14. OL Mangasarian, Minimum-support solutions of polyhedral concave programs. Optimization. **45**, 149–162 (1999)
15. M Fazel, H Hindi, S Boyd, Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices, American Control Conference. Proceedings of the 2003. **3**, 2156–2162 (2003). doi:10.1109/ACC.2003.1243393
16. R Chartrand, Exact reconstruction of sparse signals via nonconvex minimization. Signal Process. Lett. IEEE. **14**(10), 707–710 (2007)
17. R Chartrand, V Staneva, Restricted isometry properties and nonconvex compressive sensing. Inverse Problems. **24**(3), 035020 (2008)
18. L Chen, Y Gu, The convergence guarantees of a non-convex approach for sparse recovery. IEEE Trans. Signal Process. **62**(15), 3754–3767 (2014)
19. M Hyder, K Mahata, An improved smoothed ℓ_0 approximation algorithm for sparse representation. IEEE Trans. Signal Process. **58**(4), 2194–2205 (2010)
20. H Mohimani, M Babaie-Zadeh, C Jutten, A fast approach for overcomplete sparse decomposition based on smoothed norm. Signal Process. IEEE Trans. **57**(1), 289–301 (2009)
21. R Saaba, O Yilmaz, Sparse recovery by non-convex optimization instance optimality. Appl. Comput. Harmonic Anal. **29**(1), 30–48 (2010)
22. S Jokar, ME Pfetsch, Exact and approximate sparse solutions of underdetermined linear equations. SIAM J. Sci. Comput. **31**(1), 23–44 (2008)
23. Y-B Zhao, D Li, Reweighted ℓ_1-minimization for sparse solutions to underdetermined linear system. SIAM J. Optim. **22**(3), 1065–1088 (2012)
24. A Chambolle, T Pock, A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vision. **40**, 120–145 (2011)
25. Q Li, CA Micchelli, L Shen, Y Xu, A proximity algorithm accelerated by Gauss-Seidel iterations for $\mathrm{L1}/\mathrm{TV}$ denoising models. Inverse Problems. **28**, 095003 (2012)
26. X Zhang, M Burger, S Osher, A unified primal-dual algorithm framework based on Bregman iteration. J. Sci. Comput. **46**, 20–46s (2011)
27. J-J Moreau, Fonctions convexes duales et points proximaux dans un espace hilbertien. C. R. Acad. Sci. Paris Sér. A Math. **255**, 1897–2899 (1962)
28. HL Bauschke, PL Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces, AMS Books in Mathematics*. (Springer, New York, 2011)
29. L Shen, BW Suter, Bounds for eigenvalues of arrowhead matrices and their applications to hub matrices and wireless communications. EURASIP J. Adv. Signal Process. doi:10.1155/2009/379402
30. KR Davidson, SJ Szarek, in *Handbook of the Geometry of Babach Spaces*. Local operator theory, random matrices and Banach spaces, vol. 1 (Elsevier Science, Amsterdam: North-Holland, 2001), pp. 317–366