RESEARCH

Adaptive recovery of dictionary-sparse signals using binary measurements

Hossein Beheshti, Sajad Daei and Farzan Haddadi[*]

*Correspondence:
farzanhaddadi@iust.ac.ir

School of Electrical Engineering,
Iran University of Science &
Technology, Tehran, Iran

Abstract

One-bit compressive sensing (CS) is an advanced version of sparse recovery in which the sparse signal of interest can be recovered from extremely quantized measurements. Namely, only the sign of each measure is available to us. The ground-truth signal is not sparse in many applications yet can be represented in a redundant dictionary. A strong line of research has addressed conventional CS in this signal model, including its extension to one-bit measurements. However, one-bit CS suffers from an extremely large number of required measurements to achieve a predefined reconstruction error level. A common alternative to resolve this issue is to exploit adaptive schemes. We utilize an adaptive sampling strategy to recover dictionary-sparse signals from binary measurements in this work. A multi-dimensional threshold is proposed for this task to incorporate the previous signal estimates into the current sampling procedure. This strategy substantially reduces the required number of measurements for exact recovery. We show that the proposed algorithm considerably outperforms the state-of-the-art approaches through rigorous and numerical analysis.

Keywords: One-bit, Dictionary-sparse signals, Adaptive measurement, High-dimensional geometry

1 Introduction

Sampling a signal heavily depends on the prior information about the signal structure. For example, if one knows the signal of interest is band-limited, the Nyquist sampling rate is sufficient for exact recovery. Signals with most of their coefficients equal to zero are called sparse. It has been observed that sparsity is a powerful assumption that significantly reduces the required number of measurements. The process of recovering a sparse signal from a small number of measurements is called compressed sensing (CS). In CS, the measurement vector is assumed to be a linear combination of the ground-truth signal, i.e.,

$$y = Ax, \tag{1}$$

where $A \in \mathbb{R}^{m \times N}$ is called the measurement matrix, and $x \in \mathbb{R}^N$ is an unknown s-sparse signal, i.e., it has at most s nonzero entries or $\|x\|_0 \leq s$. Here, $\|\cdot\|_0$ is the ℓ_0 norm which

counts the number of nonzero elements.[1] It has been shown that $\mathcal{O}(s \log(\frac{N}{s}))$ measurements are sufficient to guarantee exact recovery of the signal, by solving the convex program:

$$P_1 : \quad \min_{z \in \mathbb{R}^N} \quad \|z\|_1 \quad \text{s.t.} \quad y = Az, \tag{2}$$

with high probability (see [1, 2]).

Practical limitations force us to quantize the measurements in (1) as $y = \mathcal{Q}(A\,x)$ where $\mathcal{Q} : \mathbb{R}^m \to \mathcal{A}^m$ is a nonlinear operator that maps the measurements into a finite symbol alphabet $\mathcal{A}$. It is an interesting question to ask: What is the result of extreme quantization? [3] addressed this question: Signal reconstruction is still feasible using only one-bit quantized measurements. In one-bit compressed sensing, samples are taken as the sign of a linear transform of the signal $y = \text{sign}(Ax)$. This sampling scheme discards magnitude information. Therefore, we can only recover the direction of the signal. Fortunately, we can keep the amplitude information by using nonzero threshold. Thus, the new sampling scheme is $y = [\text{sign}](Ax - \tau)$ where τ is the threshold vector. In our work, each element of τ is generated via $\tau_i \sim \mathcal{N}(0, 1)$. While a great part of CS literature discusses sparse signals, most natural signals are dictionary-sparse, i.e., sparse in a transform domain. For instance, sinusoidal signals and natural image are sparse in Fourier and wavelet domains, respectively [4–7]. This means that the signal of interest $f \in \mathbb{R}^n$ can be expressed as $f = Dx$ where $D \in \mathbb{R}^{n \times N}$ is a redundant dictionary with the constrain $DD^H = I^2$ and $x \in \mathbb{R}^N$ is a sparse vector. With this assumption, $y = Af = ADx$ gives the measurement vector. A common approach for recovering such signals is to use the optimization problem

$$[\,P]_{1,D} : \quad \min_{z \in \mathbb{R}^N} \quad \left\| D^H z \right\|_1 \quad \text{s.t.} \quad y = Az, \tag{3}$$

which is called ℓ_1 analysis problem [5, 6].

In this work, we investigate a more practical situation where the signal of interest f is *effective s-analysis-sparse* which means that f satisfies $\|D^H f\|_1 \leq \sqrt{s}\|D^H f\|_2$. In fact, perfect dictionary sparsity is rarely satisfied in practice, since real-world signals of interest are only compressible in a domain. Our approach is adaptive which means that we incorporate previous signal estimates into the current sampling procedure. More explicitly, we solve the optimization problem

$$\min_{z \in \mathbb{R}^N} \quad \left\| D^H z \right\|_1 \text{ s.t. } y := \text{sign}(Af - \varphi) = \text{sign}(Az - \varphi), \tag{4}$$

where $\varphi \in \mathbb{R}^m$ is a vector of thresholds chosen adaptively based on previous estimations via Algorithm 1. We propose a strategy to find a best effective s-analysis-sparse approximation to a signal in $\mathbb{R}^n$.

---

[1] The used notation is introduced in Sect. 1.2.

[2] $(D)^H$ stands for the Hermitian of the matrix D

1.1 Contributions

In this section, we state our novelties compared to the previous works. To highlight the contributions, we list them as below.

1. **Proposing a novel algorithm for dictionary-sparse signals**: We introduce an adaptive thresholding algorithm for reconstructing dictionary-sparse signals in case of binary measurements. The proposed algorithm provides accurate signal estimation even in case of redundant and coherent dictionaries. The required number of one-bit measurements considerably outperforms the non-adaptive approach used in [8].

2. **Exponential decay of reconstruction error**: The reconstruction error of our algorithm reduces exponentially when the number of adaptive stages (i.e., T) increases. To be more precise, we obtain a near-optimal relation between the reconstruction error and the required number of adaptive batches. Written in mathematical form, if one takes the output of our reconstruction algorithm by $\boldsymbol{f}_T$, then we show that $\|\boldsymbol{f}_T - \boldsymbol{f}\|_2 \approx \mathcal{O}(\frac{1}{2^T})$, where $\boldsymbol{f}$ is the ground-truth signal and T is the number of stages in our adaptive algorithm (see Theorem 1 for more details).

3. **High-dimensional threshold selection**: We propose an adaptive high-dimensional threshold to extract the most information from each sample, which substantially improves performance and reduces the reconstruction error (see Algorithm 1 for more explanations).

1.2 Prior works and key differences

In this section, we review prior works about applying quantized measurements to CS framework [3, 8–13]. In what follows, we explain some of them.

The authors of [3] propose a heuristic algorithm to reconstruct the ground-truth sparse signal from extreme quantized measurements, i.e., one-bit measurements. In [9], it has been shown that conventional CS algorithms also work well when the measurements are quantized. In [10], an algorithm with simple implementation is proposed. This algorithm has less error in Hamming distance than the existing ones. Investigated from a geometric view, the authors of [11] exploit functional analysis tools to provide an almost optimal solution to the problem of one-bit CS. They show that the number of required one-bit measurements is $\mathcal{O}(s\log^2(\frac{n}{s}))$.

The work of [14] presents two algorithms for full (i.e., direction and norm) reconstruction with provable guarantees. The former approach takes advantage of the random thresholds, while the latter predicts the direction and magnitude separately. The authors of [12] introduce an adaptive thresholding scheme which utilizes a generalized approximate message passing algorithm (GAMP) [12] for recovery and thresholds update throughout sampling. In a different approach, the work [13] proposes an adaptive quantization and recovery scenario making an exponential error decay in one-bit CS frameworks. The authors of [15] use an adaptive process to take measurements around the estimated signal in each iteration. While [15] only changes the bias (mean) of the estimated signal, our algorithm also generates random dithers with adaptive variance. The variance value of thresholds initializes from an overestimation
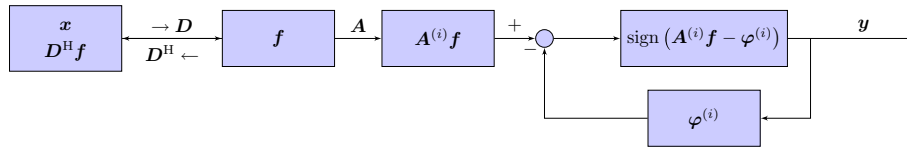
Fig. 1 Block diagram of adaptive sampling procedure

of desired signal and is divided by a factor of 2 in each iteration. This adaptive thresholding scheme forces the algorithm estimate to concentrate around the optimal solution and makes the feasible set smaller in each iteration (random part with reduced variance).

Many of the techniques mentioned for adaptive sparse signal recovery do not generalize (at least not in an obvious strategy) to dictionary-sparse signal. For example, determining a surrogate of f that is supposed to be of lower complexity with respect to D^H is non-trivial and challenging. We should emphasize that while the proofs and main parts in [13] rely on hard thresholding operator, it could not be used for either effective or exact dictionary-sparse signals. This is due to that given a vector x in the analysis domain[3] $\mathbb{R}^N$, one cannot guarantee the existence of a signal f in $\mathbb{R}^n$ such that $D^H f = x$. Recently, the work [8] shows both direction and magnitude of a dictionary-sparse signal can be recovered by a convex program with strong guarantees. The work [8] has inspired our work for recovering dictionary-sparse signal in an adaptive manner. In contrast to the existing method [8] for binary dictionary-sparse signal recovery which takes all of the measurements in one step with fixed settings, we solve the problem in an adaptive multistage way. In each stage, regarding the estimate from previous stage, our algorithm is propelled to the desired signal. In the non-adaptive work [8], the error rate is poorly large, while in our work, the error rate exponentially decays with increased number of adaptive steps.

Notation. Here, we introduce the notation used in the paper. Vectors and matrices are denoted by boldface lowercase and capital letters, respectively. A^T and A^H stand for transposition and Hermitian of A, respectively. C and c denote positive absolute constants which can be different from line to line. We use $\|v\|_2 = \sqrt{\sum_i |v_i|^2}$ for the ℓ_2-norm of a vector v in $\mathbb{R}^n$, $\|v\|_1 = \sum_i |v_i|$ for the ℓ_1-norm and $\|v\|_\infty = \max_i |v_i|$ for the ℓ_∞-norm. We write $\mathbb{S}^{n-1} := \{v \in \mathbb{R}^n : \|v\|_2 = 1\}$ for the unit Euclidean sphere in $\mathbb{R}^n$. For $x \in \mathbb{R}^n$, we define x_S as the sub-vector in $\mathbb{R}^{|S|}$ consisting of the entries indexed by the set S.

2 Method

In this section, we present the general procedure of our adaptive scheme in details. Our system model is built upon the optimization problem (4). A major part of this problem is to choose an efficient threshold $\varphi \in \mathbb{R}^m$. To this end, we propose a closed-loop feedback system (see Fig. 1) which exploits prior information from previous stages. The superscript (*i*) corresponding to an element represents the *i*-th stage of that element in the adaptive algorithm. Our adaptive approach consists of three algorithms *high-dimensional threshold generator* (HDTG), *adaptive sampling* (AS) and

---

[3] The domain in which the desired signal is being analyzed due to having a particular structure, e.g., sparsity.

adaptive recovery (AR). HDTG specifies the method of choosing the thresholds based on previous estimates. AS takes one-bit measurements of the signal by using HDTG and returns the one-bit samples and corresponding thresholds. Lastly, AR recovers the signal given the one-bit measurements. We provide a mathematical framework to guarantee our algorithm results in the following.

Theorem 1   (Main theorem) *Let $r, c_\epsilon, \gamma, \epsilon > 0$. Consider $\boldsymbol{f} \in \mathbb{R}^n$ as the desired effective s-analysis-sparse signal with $\|\boldsymbol{f}\|_2 \le r$ and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is the measurement matrix with standard normal entries where m is the total number of measurements divided into T stages. Assume that*

$$[\boldsymbol{y}, \boldsymbol{\varphi}] = \mathsf{AS}(\boldsymbol{A}, \boldsymbol{f}, \boldsymbol{D}, r, T), \tag{5}$$

$$\boldsymbol{f}_T = \mathsf{AR}(\boldsymbol{A}, \boldsymbol{D}, \boldsymbol{y}, \boldsymbol{\varphi}, r, T) \tag{6}$$

be the sampling and recovery algorithms introduced later in Algorithms 2 and 3, respectively, where $\boldsymbol{\varphi}$ is determined by Algorithm 1. Then, with probability at least $1 - \gamma \exp(\frac{-c_\epsilon m}{T})$ over the choice of $\boldsymbol{\varphi}$ and $\boldsymbol{A}$, the output of Algorithm 3 satisfies

$$\|\boldsymbol{f} - \boldsymbol{f}_T\|_2 \le \epsilon r 2^{1-T}. \tag{7}$$

1 *Proof*
See Appendix 5.1. □

A remarkable note is that if we only consider one stage, i.e., $T = 1$, the exponential behavior of our error bound disappears and reaches the state-of-the-art error bound [8, Theorem 8]. In fact, the results of [8] are a special case of our work when the thresholds are non-adaptive. The term *adaptivity* is related to the threshold updating, and the measurement matrix ($\boldsymbol{A}$) is fixed.

In what follows, we provide rigorous explanations about our proposed algorithms 1, 2 and 3 in three items.

1. **High-Dimensional Threshold**: Our algorithm for high-dimensional threshold selection is given in Algorithm 1. The algorithm output consists of two parts: deterministic (i.e., $\boldsymbol{A}\boldsymbol{f}_i$) and random (i.e., $\boldsymbol{\tau} \in \mathbb{R}^q$) . The former transfers the origin to the previous signal estimate ($\boldsymbol{f}_i$), while the latter creates measurement dithers from the origin. (The variance parameter σ^2 controls the threshold distance from $\boldsymbol{f}_i$.)

Algorithm 1 HDTG: High dimension threshold generator

Input: Mapping matrix $\boldsymbol{A}$, number of measurements q, dithers variance σ^2, signal estimation $\boldsymbol{f}_i$.
Output: High dimension threshold vector $\boldsymbol{\varphi} \in \mathbb{R}^q$.
1: $\boldsymbol{\tau} \sim N(0, \sigma^2 \boldsymbol{I}_q)$
2: $\boldsymbol{\varphi} = \boldsymbol{A}\boldsymbol{f}_i + \boldsymbol{\tau}$
3: **return** $\boldsymbol{\varphi}$

---

2. **Adaptive Sampling**: Our adaptive sampling algorithm is given in Algorithm 2. To implement this algorithm, we need the dictionary $\boldsymbol{D} \in \mathbb{R}^{n \times N}$, the measurement matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, linear measurement $\boldsymbol{A}\boldsymbol{f}$ and an over estimation of signal power r ($\|\boldsymbol{f}\|_2 \leq r$). At the first stage, we initialize signal estimation to zero vector. We choose T stages for our algorithm. At the i-th stage, the measurement matrix $\boldsymbol{A}^{(i)}$ is taken from the i-th row subset (of size $q := \lfloor \frac{m}{T} \rfloor$) of $\boldsymbol{A}$. The adaptive sampling process consists of four essential parts. First, in step 2 of the pseudocode, we generate the high-dimensional thresholds using Algorithm 1 by the parameters $\sigma^2 = 2^{1-i} r$ and $\boldsymbol{f}_i$. Second, we compare the linear measurement block $\boldsymbol{A}^{(i)}\boldsymbol{f}$ with the threshold vector $\boldsymbol{\varphi}^{(i)}$ and obtain the sample vector $\boldsymbol{y}^{(i)}$ (step 3 of Algorithm 2). Third, we implement a second-order cone program (steps 4 and 5) to find an approximate for $\boldsymbol{f}$. However, this strategy does not often lead to an effective dictionary-sparse signal. So, in step 6, we devise a strategy to find a low-complexity approximation of $\boldsymbol{f}$ with respect to operator $\boldsymbol{D}^{\mathrm{H}}$. In other words, we project the resulting signal onto the nearest effective s-analysis-sparse signal. We refer to the third part of the adaptive sampling algorithm (steps 4–6) as single-step recovery (SSR). The estimated signal at each stage ($\boldsymbol{f}_i$) builds the deterministic part of our high-dimensional threshold in step 2. Finally, Algorithm 2 returns binary vectors $\{\boldsymbol{y}^{(i)}\}_{i=1}^{T}$ and the threshold vectors $\{\boldsymbol{\varphi}^{(i)}\}_{i=1}^{T}$ to the output.

Algorithm 2 AS: Adaptive Sampling

Input: $\boldsymbol{D} \in \mathbb{R}^{n \times N}$, measurement matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, linear measurement $\boldsymbol{A}\boldsymbol{f} \in \mathbb{R}^m$, norm estimation $\|\boldsymbol{f}\|_2 \leq r$, sparsity parameter s, number of blocks T.
Output: Quantized measurements $\boldsymbol{y} \in \{\pm 1\}^m$, high dimension thresholds $\boldsymbol{\varphi} \in \mathbb{R}^m$
    Initialization : $q = \lfloor \frac{m}{T} \rfloor$, $\boldsymbol{A}^{(i)} \in \mathbb{R}^{q \times n}$, $\boldsymbol{f}_0 \leftarrow \boldsymbol{0}$
1: **for** $i = 1, \cdots, T$ **do**
2:     $\boldsymbol{\varphi}^{(i)} \leftarrow \mathsf{HDTG}(\boldsymbol{A}^{(i)}, q, 2^{1-i} r, \boldsymbol{f}_{i-1})$
3:     $\boldsymbol{y}^{(i)} = \mathsf{sign}\left(\boldsymbol{A}^{(i)}\boldsymbol{f} - \boldsymbol{\varphi}^{(i)}\right)$
4:     $\boldsymbol{\Delta}_i \leftarrow \arg\min_{\boldsymbol{z} \in \mathbb{R}^n} \|\boldsymbol{D}^{\mathrm{H}}\boldsymbol{z}\|_1$
        $s.t.\ \boldsymbol{y}^{(i)} = \mathsf{sign}\left(\boldsymbol{A}^{(i)}\boldsymbol{z} - \boldsymbol{\varphi}^{(i)}\right),\ \|\boldsymbol{z}\|_2 \leq 2^{1-i} r$
5:     $\boldsymbol{f}_{tmp} = \boldsymbol{f}_{i-1} + \boldsymbol{\Delta}_i$
6:     $\boldsymbol{f}_i \leftarrow \arg\min_{\boldsymbol{z} \in \mathbb{R}^n} \|\boldsymbol{z} - \boldsymbol{f}_{tmp}\|_2\ \ s.t.\ \ \|\boldsymbol{D}^{\mathrm{H}}\boldsymbol{z}\|_1 \leq \sqrt{s} r$
7: **end for**
8: **return** $\boldsymbol{y}^{(i)}, \boldsymbol{\varphi}^{(i)}$ for $i = 1, \cdots, T$

Algorithm 3 AR: Adaptive Recovery

Input: $\boldsymbol{D} \in \mathbb{R}^{n \times N}$, measurement matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, quantized measurements $\boldsymbol{y} \in \{\pm 1\}^m$, high dimension thresholds $\boldsymbol{\varphi} \in \mathbb{R}^m$, norm estimation $\|\boldsymbol{f}\|_2 \leq r$, sparsity parameter s, number of blocks T.
Output: Estimated vector $\boldsymbol{f}_T \in \mathbb{R}^n$.
    Initialization : $q = \lfloor \frac{m}{T} \rfloor$, $\boldsymbol{A}^{(i)} \in \mathbb{R}^{q \times n}$, $\boldsymbol{y}^{(i)} \in \{\pm 1\}^q$, $\boldsymbol{\varphi}^{(i)} \in \mathbb{R}^q$, $\boldsymbol{f}_0 \leftarrow \boldsymbol{0}$
1: **for** $i = 1, \cdots, T$ **do**
2:     $\boldsymbol{\Delta}_i \leftarrow \arg\min_{\boldsymbol{z} \in \mathbb{R}^n} \|\boldsymbol{D}^{\mathrm{H}}\boldsymbol{z}\|_1$
        $s.t.\ \boldsymbol{y}^{(i)} = \mathsf{sign}\left(\boldsymbol{A}^{(i)}\boldsymbol{z} - \boldsymbol{\varphi}^{(i)}\right),\ \|\boldsymbol{z}\|_2 \leq 2^{1-i} r$
3:     $\boldsymbol{f}_{tmp} = \boldsymbol{f}_{i-1} + \boldsymbol{\Delta}_i$
4:     $\boldsymbol{f}_i \leftarrow \arg\min_{\boldsymbol{z} \in \mathbb{R}^n} \|\boldsymbol{z} - \boldsymbol{f}_{tmp}\|_2\ \ s.t.\ \ \|\boldsymbol{D}^{\mathrm{H}}\boldsymbol{z}\|_1 \leq \sqrt{s} r$
5: **end for**
6: **return** $\boldsymbol{f}_T$

---

3. **Adaptive Recovery**: In the recovery procedure (Algorithm 3), we need the dictionary $\boldsymbol{D}$, the measurement matrix $\boldsymbol{A}$, binary measurements vector $\boldsymbol{y}$, high-dimensional

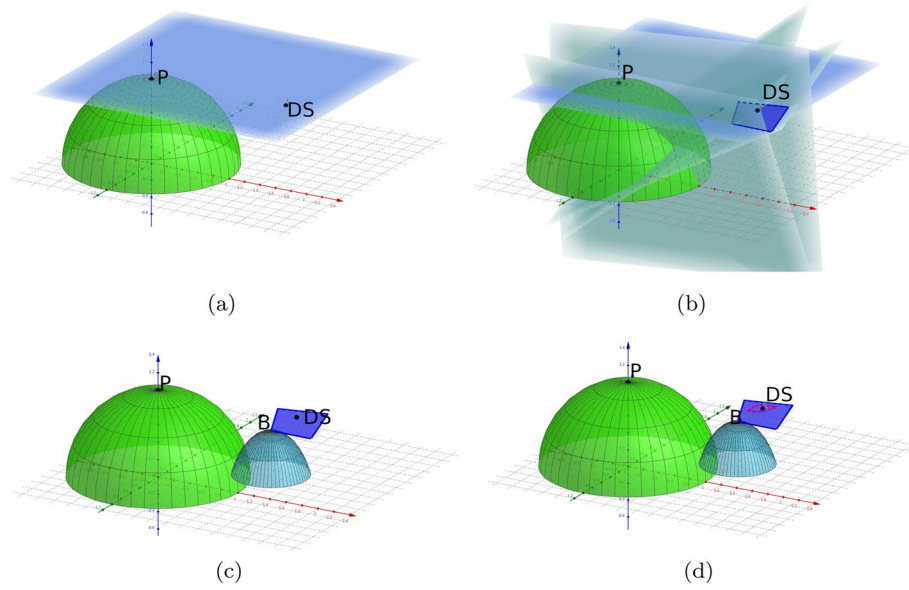Beheshti *et al. EURASIP Journal on Advances in Signal Processing* (2022) 2022:49

Page 7 of 13

(a)

(b)

(c)

(d)

Fig. 2 3D illustration of random hyperplane tessellation for the first example of intuitive description

threshold vector $\boldsymbol{\varphi}$ and an upper norm estimation of signal r. In the adaptive recovery algorithm, we first divide the inputs ($\boldsymbol{y}$ and $\boldsymbol{A}$) into T blocks (i.e., $\boldsymbol{y}^{(i)}$ and $\boldsymbol{A}^{(i)}$). Then, we simply implement SSR on each block. The output of SSR in the last stage is the final recovered signal (i.e., $\boldsymbol{f}_T$).

2.1 Intuitive description of our algorithms

We provide here an example to clarify how our algorithm works. In this example, we consider a 2D point denoted by point **DS** (Fig. 2a) as the desired signal and depict the algorithm procedure for two consecutive iterations. At the first iteration of the algorithm, we do not have any estimation of the desired signal. Therefore, we set the first estimate to zero (equivalently $[0, 0, 1]^T$ in the 3D domain or point **P** in Fig. 2a). We generate high-dimensional thresholds and take four measurements (step 2 of AS). These measurements are equivalently four hyperplane crossing the center of the hemisphere (Fig. 2b, step 3 of AS). In the next step, we implement the augmented random hyperplane tessellation on the hemisphere and compute the first estimate of the desired signal and project it to the signal domain (steps 3–6 of AS). In this example, we consider the point **B** as the first estimate ($\boldsymbol{f}_1$ in first iteration of AS). In the next iteration, we transfer the hemisphere to the adjacent point **B** and reduce the radius by a factor of two (Fig. 2c). The point **B** is the high-dimensional threshold of the second iteration of the algorithm. We perform our algorithm with four measurements. The main advantage of the reduction in the radius of the hemisphere (equivalently the variance argument of the threshold generator) is to get more precise measurements in the neighborhood of the desired signal. In the second stage of the algorithm, we take measurements (or equivalently hyperplanes) around the threshold, leading to locate measurements more concentrated around the desired signal. The intersection of hyperplanes and signal space in this
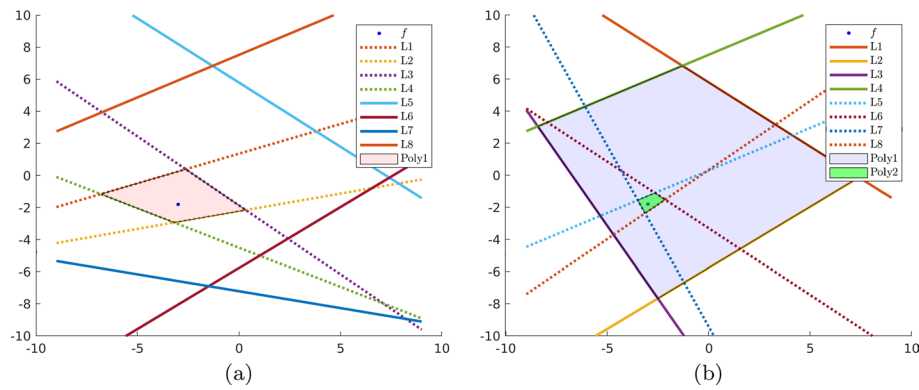
Beheshti *et al. EURASIP Journal on Advances in Signal Processing*   (2022) 2022:49

Page 8 of 13

Fig. 3 This figure compares the sampling procedure of the non-adaptive algorithm (**a**) with our adaptive algorithm (**b**)

example creates a new feasible set border which is depicted by the red line in Fig. 2d. In this intuitive description, we tried to clarify the random hyperplane tessellation usage in our proof sketch.[4].

In the second example, we follow our algorithm steps in a 2D example and compare it with the non-adaptive algorithm. In Fig. 3a, we perform the non-adaptive algorithm for eight measurements. It is obvious that the solid lines ($L_5 - L_8$) or equivalently measurements do not have any impact on the final result. However, if we get these measurements in two iterations (i.e., stages) of the adaptive algorithm, the second bunch of samples ($L_5 - L_8$ in Fig. 3(b)) is more concentrated toward the desired signal (f).

3 Experimental results and discussion

In this section, we investigate the performance of our algorithm and compare it with the two previous one-bit dictionary-sparse recovery given by [8]. The first algorithm solves linear programming optimization (LP) [8, Subsection 4.1], and the second algorithm solves a second-order cone programming (CP) optimization [8, Subsection 4.2]. First, we construct a matrix where its columns are drawn randomly and independently from $\mathbb{S}^{n-1}$. Then, the dictionary $\boldsymbol{D} \in \mathbb{R}^{n \times N}$ ($N = 1000, n = 50$) is defined as an orthonormal basis of this matrix. Then, the signal f is generated as $f = \boldsymbol{B}c$ where $\boldsymbol{B}$ is a basis of null($\boldsymbol{D}_{\overline{\mathcal{S}}}$) and c is drawn from standard normal distribution. Here, $\overline{\mathcal{S}}$ is used to denote the complement of the support of $\boldsymbol{D}^{\mathrm{H}}f$. The elements of $\boldsymbol{A}$ are chosen from i.i.d. standard normal distribution. We set $r = 2\|f\|_2, \sigma = r$ and the number of stages to $T = 10$. Define the normalized reconstruction error as $\frac{\|f - \hat{f}\|_2}{\|f\|_2}$ ($\hat{f}$ denotes the estimated signal in each algorithm, and it is represented by f_T) especially in our algorithm. The results in Fig. 4a are obtained by implementing Algorithms 2 and 3 100 times and taking the average of the normalized reconstruction error. As it is clear from Fig. 4a, LP algorithm outperforms CP by 2 *dB* on average. Our algorithm with few measurements behaves slightly weaker than others. The proposed approach requires a minimum number of measurements to act perfectly. The main reason for this behavior is our thresholding scheme. The variance of dithers is

---

[4] We prepare figures by means of GeoGebra tools, and the 3D model is available by clicking here

Table 1 Average execution time (millisecond)

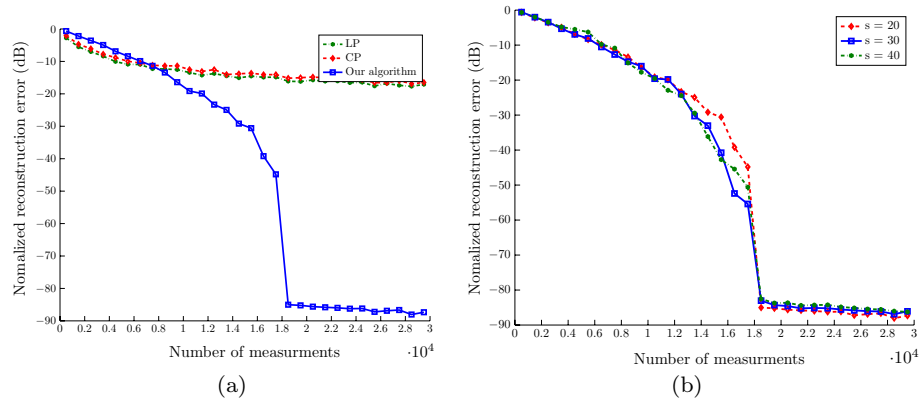| Sparsity (*s*)	20		
Number of measurements (*m*)	4500	20500	29500
LP	13.113	71.320	75.165
CP	4.088	17.325	10.370
Out algorithm	21.841	29.999	16.011

Fig. 4 Reconstruction of a dictionary-sparse signal $\boldsymbol{x} \in \mathbb{R}^{50}$ in the dictionary $\boldsymbol{D} \in \mathbb{R}^{50 \times 1000}$. Both figures show the reconstruction error averaged over 100 Monte Carlo simulations. Image **a** compares the performance of the algorithms in $s = 20$. Image **b** examines the performance of our algorithm over different sparsity levels

reduced in each iteration (i.e., stage). We also take the measurements around the previous estimate of the desired signal. On the contrary, the CP and LP algorithms only reduce the signal space uniformly. When the number of measurements is below the required bound, our first estimate gets away from the desired signal, so the adaptive algorithm tries to find it around a false estimate. However, it seems that there is a phase transition behavior in our algorithm when the number of measurements increases. In fact, after a certain number of measurements, our proposed algorithm substantially outperforms both LP and CP. Table 1 shows the average execution time (CPU time) of the three algorithms per the specified number of measurements.

In the second experiment, we examine the performance of our algorithm for multiple degrees of sparsity. Figure 4b shows the result for $s = 20, 30, 40$. (Other parameters are assumed similar to the first experiment.) As it is clear from the figure, by increasing the sparsity level, the accuracy is decreased.

In the third experiment, we consider the Shepp–Logan phantom image as the ground-truth signal. Since the required number of measurements in one-bit CS is significantly larger than that of the conventional CS (see, e.g., [13, Section 5]), we split the picture into multiple blocks of size 32×32 and process each block separately, merely due to computational complexity reasons. This block processing allows us to implement the algorithm in multiple threads and reduce the execution time. We use a redundant wavelet dictionary and 10^5 Gaussian measurements to recover each block. We evaluate the reconstruction quality of final result in terms of the peak signal-to-noise ratio (PSNR) given by
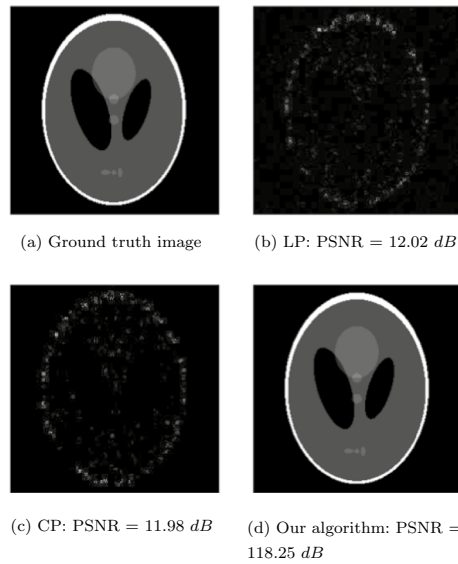
(a) Ground truth image   (b) LP: PSNR = 12.02 dB

(c) CP: PSNR = 11.98 dB (d) Our algorithm: PSNR = 118.25 dB

Fig. 5 Simulation results for reconstruction of the Shepp–Logan phantom image (phantom(256) in MATLAB) in which the picture split to 32×32 blocks and each algorithm get $m = 10^5$ measurements per block

$$\mathrm{PSNR}\left(X, \widehat{X}\right) = 20 \log_{10}\left(\frac{\|X\|_{\infty} 256}{\|X - \widehat{X}\|_F}\right), \tag{8}$$

where X and $\widehat{X}$ are the true and estimated images. As shown in Fig. 5, LP (Fig. 5b) and CP (Fig. 5c) algorithms clearly fail with poor performances, but as it is evident in Fig. 5d, the output of our algorithm is almost similar to the desired picture.

4 Conclusion

This paper proposed a one-bit compressed sensing algorithm that recovers dictionary-sparse signals from binary measurements by adaptively exploiting the inherent information in the sampling procedure. This scheme helps to reduce the number of needed measurements substantially. In addition, our algorithm exhibits an exponential decaying behavior in the reconstruction error. The proof approach is based on geometric theories in the high-dimensional estimation. In this work, we used geometric intuition to explain our result, which also can be used in other areas of signal processing. We believe our analysis can be extended to the multi-bit setting. We used a fixed reduction pattern in the thresholds dithers throughout this work. We believe that this reduction can be chosen smartly by extracting the geometric features in each algorithm step.

5 Proofs
5.1 Proof of Theorem 1

1 *Proof*

By induction law, we show that

$$\|f - f_i\|_2 \le \epsilon r 2^{1-i} \tag{9}$$

holds with high probability for any $i \in \{1, \ldots, T\}$. Consider the first step, i.e., $i = 1$ in Algorithm 2. At this step, our initial estimate $\boldsymbol{f}_0$ is equal to $\boldsymbol{0}$. Thus, the output of Algorithm 1 only contains the random part of high-dimensional thresholds (step 2 of Algorithm 2). Then, we obtain $\boldsymbol{f}_1$ by using steps 4–9 of Algorithm 2 (except that we assume $\sigma = r$ in Algorithm 2 and $q = \lfloor \frac{m}{T} \rfloor$). As a result, to verify $\|\boldsymbol{f} - \boldsymbol{f}_1\|_2 \leq \epsilon r$, we use [8, Theorem 8] to prove our result: $\qquad\qquad\square$

Theorem 2 [8, *Theorem 8*] *Let* $\epsilon, r, \sigma > 0$, *let*

$$m \geq C\left(\tfrac{r}{\sigma} + \tfrac{\sigma}{r}\right)^{-6}\left(\tfrac{r^2}{\sigma^2} + 1\right)\epsilon^{-6} s \ln(\tfrac{eN}{s}),$$

and let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ *be populated by independent standard normal random variables. Furthermore, let* $\tau_1, \ldots, \tau_m$ *be independent normal random variables with mean zero and variance* σ^2 *that are also independent from* $\boldsymbol{A}$. *Then, with failure probability at most* $\gamma \exp(\frac{-c' m \epsilon^2 r^2 \sigma^2}{(r^2 + \sigma^2)^2})$, *any signal* $\boldsymbol{f} \in \mathbb{R}$ *with* $\|\boldsymbol{f}\|_2 \leq r$, $\|\boldsymbol{D}^{\mathrm{H}}\boldsymbol{f}\|_1 \leq \sqrt{s}r$ *and observed via* $\boldsymbol{y} = \mathrm{sign}(\boldsymbol{A}\boldsymbol{f} - \boldsymbol{\tau})$ *is approximated by*

$$\boldsymbol{f}_{\mathrm{CP}} = \operatorname*{arg\,min}_{\boldsymbol{z} \in \mathbb{R}^n} \left\|\boldsymbol{D}^{\mathrm{H}}\boldsymbol{z}\right\|_1 \text{ s.t. } \boldsymbol{y} = \mathrm{sign}(\boldsymbol{A}\boldsymbol{z} - \boldsymbol{\tau}), \ \|\boldsymbol{z}\|_2 \leq r,$$

with error

$$\|\boldsymbol{f} - \boldsymbol{f}_{\mathrm{CP}}\|_2 \leq \epsilon r. \tag{10}$$

Now, suppose that the result (9) holds in the $(i-1)$-th step, i.e.,

$$\|\boldsymbol{f} - \boldsymbol{f}_{i-1}\|_2 \leq \epsilon r 2^{2-i}. \tag{11}$$

Consider i-th stage of Algorithm 2 where the high-dimensional thresholds and the measurements are obtained as

$$\boldsymbol{\varphi}^{(i)} \leftarrow \mathsf{HDTG}(\boldsymbol{A}^{(i)}, q, 2^{1-i}r, \boldsymbol{f}_{i-1}), \tag{12}$$

$$\boldsymbol{y}^{(i)} = \mathrm{sign}\left(\boldsymbol{A}^{(i)}\boldsymbol{f} - \boldsymbol{\varphi}^{(i)}\right). \tag{13}$$

By substituting (12) in (13), we reach:

$$\boldsymbol{y}^{(i)} = \mathrm{sign}\left(\boldsymbol{A}^{(i)}(\boldsymbol{f} - \boldsymbol{f}_{i-1}) - \boldsymbol{\tau}^{(i)}\right). \tag{14}$$

Since $\boldsymbol{f}$ is effective s-analysis-sparse with $\|\boldsymbol{f}\|_2 \leq 1$, we have that $\|\boldsymbol{D}^H\boldsymbol{f}\|_1 \leq \sqrt{s}\|\boldsymbol{f}\|_2$. Further, the output of Algorithm 2 at the $(i-1)$-th stage satisfies, i.e., $\|\boldsymbol{D}^H\boldsymbol{f}_{i-1}\|_1 \leq \sqrt{s}r$, and it then holds that the signal $\boldsymbol{f} - \boldsymbol{f}_{i-1}$ satisfies $\|\boldsymbol{D}^H(\boldsymbol{f} - \boldsymbol{f}_{i-1})\|_1 \leq \|\boldsymbol{D}^H\boldsymbol{f}\|_1 + \|\boldsymbol{D}^H\boldsymbol{f}_{i-1}\|_1 \leq 2\sqrt{s}r = \sqrt{2^{2i-2}\epsilon^{-2}s}\epsilon 2^{-i}r := \sqrt{s'}r'$. Now, by leveraging the assumption in (11), we can directly apply Theorem 2 to this signal. In simple words, we set

$$\begin{aligned}
\boldsymbol{f} &\leftarrow \boldsymbol{f} - \boldsymbol{f}_{i-1}, \\
r &\leftarrow \epsilon 2^{2-i} r, \\
\sigma &\leftarrow \epsilon 2^{2-i} r, \\
s &\leftarrow s'.
\end{aligned} \tag{15}$$

in Theorem 2. As a result, by exploiting this theorem and some simplifications, we shall have that

$$\left\| \boldsymbol{f} - \boldsymbol{f}_{i-1} - \boldsymbol{\Delta}_i \right\|_2 \le \epsilon_1 r' := \epsilon r 2^{-i}, \tag{16}$$

with probability at least $1 - \gamma \exp(-c'_\epsilon q)$ where in the latter expression we used $\epsilon_1 = \frac{1}{4}$ and $\boldsymbol{\Delta}_i$ is the output vector of the optimization problem in step 4 of Algorithm 2. c'_ϵ is a certain constant dependent on ϵ. Now, suppose that (16) occurs. Consider

$$\boldsymbol{f}_{\text{tmp}} = \boldsymbol{f}_{i-1} + \boldsymbol{\Delta}_i. \tag{17}$$

After applying step 6 of Algorithm 2, we obtain $\boldsymbol{f}_i$ that has the property

$$\| \boldsymbol{D}^{\text{H}} \boldsymbol{f}_i \|_1 \le \sqrt{s} r.$$

Then, we have

$$\| \boldsymbol{f} - \boldsymbol{f}_i \|_2 \le \| \boldsymbol{f} - \boldsymbol{f}_{\text{tmp}} \|_2 + \| \boldsymbol{f}_{\text{tmp}} - \boldsymbol{f}_i \|_2. \tag{18}$$

Finally, by using (16), (17), and the fact that $\| \boldsymbol{f}_i - \boldsymbol{f}_{\text{tmp}} \|_2 \le \| \boldsymbol{f} - \boldsymbol{f}_{\text{tmp}} \|_2$ (step 6), the latter equation becomes

$$\| \boldsymbol{f} - \boldsymbol{f}_i \|_2 \le \epsilon 2^{1-i} r \tag{19}$$

Since we consider T stage in Algorithms 2 and 3, we reach the error bound:

$$\left\| \boldsymbol{f} - \boldsymbol{f}_T \right\|_2 \le \epsilon r 2^{1-T}. \tag{20}$$

Abbreviations

CS	Compressed Sensing
GAMP	Generalized Approximate Message Passing
HDTG	High-Dimensional Threshold Generator
AS	Adaptive Sampling
AR	Adaptive Recovery
SSR	Single-Step Recovery
LP	Linear Programming
CP	Cone Programming
PSNR	Peak Signal-to-Noise Ratio

Beheshti *et al. EURASIP Journal on Advances in Signal Processing*   (2022) 2022:49

Page 13 of 13

Declarations

Consent for publication

The manuscript does not contain any individual person's data in any form (including individual details, images or videos), and therefore, the consent to publish is not applicable to this article.

Competing interests

The authors declare that they have no competing interests.

References

1. D.L. Donoho, Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
2. E.J. Candès, *et al.*, Compressive sampling, in *Proceedings of the International Congress of Mathematicians*, vol. 3, pp. 1433–1452 (2006). Madrid, Spain
3. P.T. Boufounos, R.G. Baraniuk, 1-bit compressive sensing, in *2008 42nd Annual Conference on Information Sciences and Systems*, pp. 16–21 (2008). IEEE
4. E.J. Candès, Y.C. Eldar, D. Needell, P. Randall, Compressed sensing with coherent and redundant dictionaries. Appl. Comput. Harmon. Anal. **31**(1), 59–73 (2011)
5. M. Genzel, G. Kutyniok, M. März, l_1-analysis minimization and generalized (co-) sparsity: when does recovery succeed? Appl. Comput. Harmon. Anal. **52**, 82–140 (2021)
6. S. Daei, F. Haddadi, A. Amini, M. Lotz, On the error in phase transition computations for compressed sensing. IEEE Trans. Inf. Theory **65**(10), 6620–6632 (2019)
7. S. Daei, F. Haddadi, A. Amini, Living near the edge: a lower-bound on the phase transition of total variation minimization. IEEE Trans. Inf. Theory **66**(5), 3261–3267 (2019)
8. R. Baraniuk, S. Foucart, D. Needell, Y. Plan, M. Wootters, One-bit compressive sensing of dictionary-sparse signals. Inf. Inference J. IMA **7**(1), 83–104 (2018)
9. A. Zymnis, S. Boyd, E. Candes, Compressed sensing with quantized measurements. IEEE Signal Process. Lett. **17**(2), 149–152 (2010)
10. L. Jacques, J.N. Laska, P.T. Boufounos, R.G. Baraniuk, Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. IEEE Trans. Inf. Theory **59**(4), 2082–2102 (2013)
11. Y. Plan, R. Vershynin, One-bit compressed sensing by linear programming. Commun. Pure Appl. Math. **66**(8), 1275–1297 (2013)
12. U.S. Kamilov, A. Bourquard, A. Amini, M. Unser, One-bit measurements with adaptive thresholds. IEEE Signal Process. Lett. **19**(10), 607–610 (2012)
13. R.G. Baraniuk, S. Foucart, D. Needell, Y. Plan, M. Wootters, Exponential decay of reconstruction error from binary measurements of sparse signals. IEEE Trans. Inf. Theory **63**(6), 3368–3385 (2017)
14. K. Knudson, R. Saab, R. Ward, One-bit compressive sensing with norm estimation. IEEE Trans. Inf. Theory **62**(5), 2748–2758 (2016)
15. F. Wang, J. Fang, H. Li, Z. Chen, S. Li, One-bit quantization design and channel estimation for massive mimo systems. IEEE Trans. Veh. Technol. **67**(11), 10921–10934 (2018)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.