A robust parallel algorithm for combinatorial compressed sensing

Rodrigo Mendoza-Smith*†, Jared Tanner*†, and Florian Wechsung*

* Mathematical Institute, University of Oxford, Oxford, OX2 6GG.
† Alan Turing Institute, British Library, London, NW1 2DB.

Abstract

It was shown in [1] that a vector $x \in \mathbb{R}^n$ with at most $k < n$ nonzeros can be recovered from an expander sketch Ax in $\mathcal{O}(\mathrm{nnz}(A) \log k)$ operations via the Parallel-ℓ_0 decoding algorithm, where $\mathrm{nnz}(A)$ denotes the number of nonzero entries in $A \in \mathbb{R}^{m \times n}$. In this paper we present the Robust-ℓ_0 decoding algorithm, which robustifies Parallel-ℓ_0 when the sketch Ax is corrupted by additive noise. This robustness is achieved by approximating the asymptotic posterior distribution of values in the sketch given its corrupted measurements. We provide analytic expressions that approximate these posteriors under the assumptions that the nonzero entries in the signal and the noise are drawn from continuous distributions. Numerical experiments presented show that Robust-ℓ_0 is superior to existing greedy and combinatorial compressed sensing algorithms in the presence of small to moderate signal-to-noise ratios in the setting of Gaussian signals and Gaussian additive noise.

I. Introduction

Compressed sensing is a well studied method by which a sparse or compressible vector can be acquired by a number of measurements proportional to the number of its dominant entries [2], [3]. To fix notation, let χ_k^n be the set of vectors in $\mathbb{R}^n$ that have at most k nonzero entries, let $x \in \chi_k^n$ and let $A \in \mathbb{R}^{m \times n}$ be a matrix with $m < n$. We will refer to A as the measurement matrix, x as the signal and $y = Ax$ as the measurements. The goal of compressed sensing is to recover the sparsest, most parsimonious, $x \in \mathbb{R}^n$ from the measurements y and the matrix A. Letting $\|\cdot\|_0$ denote the number of non-zeros in x, the problem of finding x can be written as

$$\min_{x \in \mathbb{R}^n} \|x\|_0 \text{ s.t. } Ax = y.$$

Many algorithms have been developed to solve this problem or equivalent formulations and there are good theoretical results on when and how fast recovery of a signal is possible given certain types of measurement matrix A and signal x. These algorithms can be broadly categorized into convex optimization based algorithms like those implemented in [4]–[7] and greedy algorithms [8]–[14], and were designed and analysed for the setting of dense sensing matrices; e.g. independent (sub-)Gaussian entries or randomly subsampled Fourier matrices. For a more detailed introduction to compressed sensing see [15].

Here we extend an algorithm proposed in [1] which can be used to recover exactly a sparse signal from its expander sketch (see Section II for details). Specifically, [1] proposed Parallel-ℓ_0 (Algorithm 2), for noiseless combinatorial compressed sensing which is guaranteed to converge in $\mathcal{O}(\mathrm{nnz}(A)\log(k))$ where the sensing matrix A is an expander matrix (Definition I.1) and the signal $x \in \chi_k^n$ is *dissociated* in the sense of Definition I.2 or the signal is drawn independently of A. For alternative combinatorial compressed sensing algorithms see, for example, [16]–[21]. We borrow notation from combinatorics and use the shorthands $[n] := \{1, \dots, n\}$, $[n]^{(k)} := \{S \subset [n] : |S| = k\}$ where $|S|$ denotes the cardinality of the set S, and $[n]^{(\leq k)} := \cup_{\ell \leq k} [n]^{(\ell)}$ for $n, k \in \mathbb{N}$ and $k < n$.

Definition I.1 (Expander matrices [1])**.** The matrix $A \in \{0,1\}^{m \times n}$ is a (k, ε, d)-expander matrix if $\sum_{i=1}^m \mathbb{1}_{|A_{i,j}|>0} = d$ for all $j \in [n]$ and

$$\left|\left\{ i \in [m] : \sum_{j \in S} \mathbb{1}_{|A_{i,j}|>0} \right\}\right| > (1 - \varepsilon) d |S|$$

RMS acknowledges the support of CONACyT
This work was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1
Furthermore, it is based on work partially supported by the EPSRC Centre For Doctoral Training in Industrially Focused Mathematical Modelling (EP/L015803/1) in collaboration with PA Consulting Group

for all $S \in [n]^{(\leq k)}$. We denote by $\mathbb{E}_{k,\varepsilon,d}^{m\times n}$ the set of (k,ε,d)-expander matrices of dimension $m \times n$.

Definition I.2 (Dissociated signals [1])**.** A signal $x \in \mathbb{R}^n$ is said to be dissociated if

$$\sum_{j\in S_1} x_j \neq \sum_{j\in S_2} x_j \qquad \forall S_1,\ S_2 \subset \operatorname{supp}(x) \text{ s.t. } S_1 \neq S_2.$$

An example of (almost surely) dissociated signals are those drawn from a continuous distribution. It is shown in [1] that if $y = Ax$ is an expander sketch and $x \in \chi_k^n$ is dissociated, then there exists a subset $T \subset [n]$ such that, for each $j \in T$, $|\{i \in [m] : y_i = x_j\}|$ is bounded below by a positive constant depending on d and ε. This guarantees that if $|\{i \in \operatorname{supp}(a_j) : y_i = y_\ell\}| > d/2$ then $x_j = y_\ell$. Parallel-ℓ_0 (Algorithm 2) implements this observation by letting $\hat{x} = 0$ and estimating the decrease in $\|y\|_0$ when performing the update $\hat{x}_j \leftarrow \hat{x}_j + y_\ell$. We denote for $j \in [n]$ its neighbours by $\mathcal{N}(j) := \{i \in [m] : |A_{ij}| > 0\}$; to estimate the decrease in $\|y\|_0$, Parallel-ℓ_0 computes

$$n_e \leftarrow |\{\ell \in \mathcal{N}(j) : y_i = y_\ell\}|\,, \tag{1}$$

$$n_z \leftarrow |\{\ell \in \mathcal{N}(j) : y_\ell = 0\}|\,. \tag{2}$$

We extend their approach to the additive noise signal model of $\hat{y} = y + \eta$ with $y = Ax$ and $\eta \in \mathbb{R}^m$ by replacing (1)-(2) with scores estimating the distribution of n_e and n_z, e.g. (8)-(9). That is, we follow a Bayesian approach to the computation of these scores and estimate:

1) The probability of $y_i = 0$ given that we observe $\hat{y}_i$.

$$p_z(\omega) := \mathbb{P}\left(y_i = 0 \mid \hat{y}_i = \omega\right). \tag{3}$$

2) The probability of $y_{i_1} = y_{i_2}$ given that we observe $\hat{y}_{i_1} - \hat{y}_{i_2}$.

$$p_e(\omega) := \mathbb{P}\left(y_{i_1} = y_{i_2} \mid \hat{y}_{i_1} - \hat{y}_{i_2} = \omega\right) \tag{4}$$

Among our contributions are series approximations for (3)-(4) when the signals and measurements are generated according to the generating model given in Definition I.3 and illustrated in Figure 1. In what follows, we let $\mathbb{D}(\mathbb{R})$ be the set of distributions supported on $\mathbb{R}$. If $\mu \in \mathbb{D}(\mathbb{R})$, we write $z \sim \mu$ to denote that z was drawn according to the distribution μ. We also use the notation $v_i \overset{\text{i.i.d.}}{\sim} \mu$ to denote that each v_i is drawn independently at random from μ. Finally, we use $U(S)$ to denote the uniform distribution over a set S.

Definition I.3 (Generating model $\mathrm{GM}(n,m,k,d,\mu,\nu)$)**.** Let $n,m,k,d \in \mathbb{N}$ be such that $k < m < n$ and $d \ll m$. Let $\mu,\nu \in \mathbb{D}(\mathbb{R})$. Then, the problem $(A,\hat{y})$ is drawn from the model $\mathrm{GM}(n,m,k,d,\mu,\nu)$ if $A \in \{0,1\}^{m\times n}$ and $\hat{y} \in \mathbb{R}^m$ are such that

1) each column of A has a support drawn uniformly at random from $[m]^{(d)}$;
2) $\operatorname{supp}(x)$ is drawn uniformly at random from $[n]^{(k)}$;
3) $x_j \overset{\text{i.i.d.}}{\sim} \mu$ for each $j \in \operatorname{supp}(x)$;
4) $\eta_i \overset{\text{i.i.d.}}{\sim} \nu$ for each $i \in [m]$;
5) η_i is independent of x_j for all $i \in [m], j \in \operatorname{supp}(x)$;
6) $y = Ax$ and $\hat{y} = y + \eta$.

We write $(A,\hat{y}) \sim \mathrm{GM}(n,m,k,d,\mu,\nu)$ to denote problem instances drawn from this signal model.

Remark. It is important to note that a matrix $A \in \mathbb{R}^{m\times n}$ generated under the model presented in Definition I.3 and Figure 1, is a (k,ε,d)-expander matrix with high probability, see [22], [15, Theorem 13.6].

Moreover, the generating model in Definition I.3 also allows us to define robust estimates for (3)-(4) for general noise and signal distributions and to any degree of accuracy under the assumption that these probability measures are available. From there we can define noisy analogues to the values n_e and n_z in (1)-(2) used in Parallel-ℓ_0 but which are robust to additive noise. The contributions of this work are two-fold: (i) to present principled ways to compute (3)-(4) in the case where the nonzeros in η and x are drawn from continuous probability distributions; (ii) to provide a variation of Parallel-ℓ_0 that is robust to noise. While other similar generating models can be considered using the techniques presented here, for ease of exposition and clarity, we restrict our description to this model.

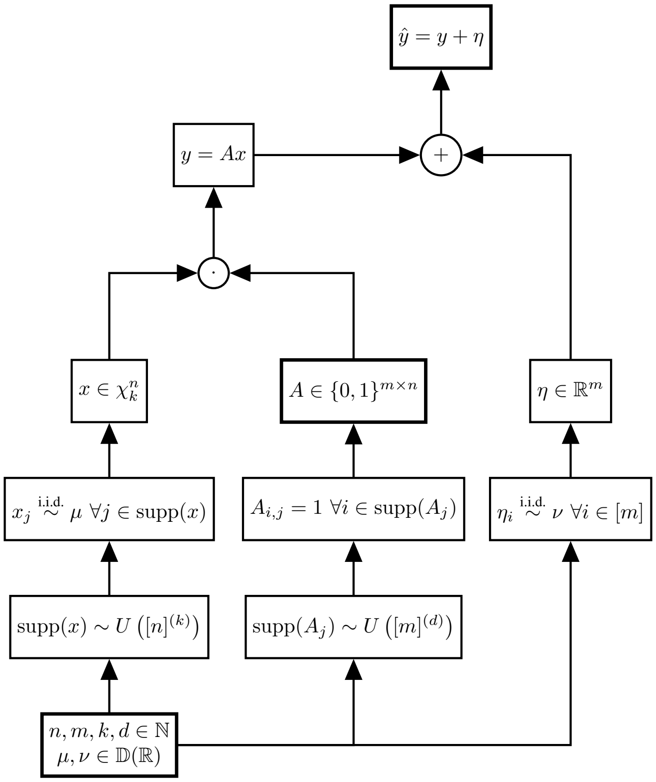

$\hat{y} = y + \eta$
$+$
$y = Ax$
$\cdot$
$x \in \chi_k^n$
$A \in \{0,1\}^{m \times n}$
$\eta \in \mathbb{R}^m$
$x_j \overset{\text{i.i.d.}}{\sim} \mu \ \forall j \in \operatorname{supp}(x)$
$A_{i,j} = 1 \ \forall i \in \operatorname{supp}(A_j)$
$\eta_i \overset{\text{i.i.d.}}{\sim} \nu \ \forall i \in [m]$
$\operatorname{supp}(x) \sim U\left([n]^{(k)}\right)$
$\operatorname{supp}(A_j) \sim U\left([m]^{(d)}\right)$
$n, m, k, d \in \mathbb{N}$
$\mu, \nu \in \mathbb{D}(\mathbb{R})$

Fig. 1: Generating model $\mathrm{GM}(n, m, k, d, \mu, \nu)$.

Theorem I.4 (Probabilities for general signal and noise distributions)**.** Let $\delta, \rho \in (0,1)$. For each $n > 1$, let $m = \delta n$, $k = \rho m$ and $d < m$. If $\mu, \nu \in \mathbb{D}(\mathbb{R})$ and $(A, \hat{y}) \sim \mathrm{GM}(n, m, k, d, \mu, \nu)$. Then as $n \to \infty$,

$$p_z(\omega) \to \frac{\nu(\omega)}{\sum_{q \geq 0} \frac{(d\rho)^q}{q!} (\nu * \mu_q)(\omega)}, \tag{5}$$

$$p_e(\omega) \to \frac{\tilde{\nu}(\omega)}{\sum_{q \geq 0} \frac{(2d\rho)^q}{q!} (\tilde{\nu} * \bar{\mu}_q)(\omega)}. \tag{6}$$

Where $\mu_q, \bar{\mu}_q, \tilde{\nu}$ are probability measures constructed as in Definition III.1.

Equations (5) and (6) allows us to quantify the uncertainty associated with computing the score for Parallel-ℓ_0 under the presence of additive noise. Note that equations (5) and (6) can be easily adapted to alternative generative models, such as where the expected density of nonzeros per row varies, but for expository clarity we restrict our discussion to this somewhat generic model. It will be discussed in Section III-C that equations (5) and (6) should not be used directly, but instead be scaled by considering *normalised* functions $\breve{p}_e$ and $\breve{p}_z$ defined as,

$$\breve{p}_e(\omega) = \frac{p_e(\omega)}{\max_s p_e(s)}, \quad \breve{p}_z(\omega) = \frac{p_z(\omega)}{\max_s p_z(s)}. \tag{7}$$

In the most general case n_e and n_z can be written as the sum of individual scores q_e and q_z as follows

$$n_e \leftarrow \sum_{\ell \in \mathcal{N}(j)} q_e\left(r_{i_1} - r_{i_2} \mid t\right) \tag{8}$$

$$n_z \leftarrow \sum_{\ell \in \mathcal{N}(j)} q_z\left(r_i \mid t\right) \tag{9}$$

A confidence threshold $t > 0$ needs to be given for some variants of our algorithm, so to simplify the exposition we include this parameter in the scores $q_e(\cdot \mid t)$ and $q_z(\cdot \mid t)$ regardless of whether it is used or not. A summary of the score functions are given in Table I.

	Robust-ℓ_0 Continuous	Robust-ℓ_0 Quantised	Parallel-ℓ_0
$q_e(r_{i_1} - r_{i_2} \mid t)$	$\breve{p}_e(r_{i_1} - r_{i_2})$	$\mathbb{1}_{\{\breve{p}_e(r_{i_1} - r_{i_2}) \geq t\}}$	$\mathbb{1}_{\{r_{i_1} = r_{i_2}\}}$
$q_z(r_i \mid t)$	$\breve{p}_z(r_i)$	$\mathbb{1}_{\{\breve{p}_z(r_i) \geq 1-t\}}$	$\mathbb{1}_{\{r_i = 0\}}$

TABLE I: Extensions of scores used in Expander ℓ_0-decoding to identify candidate updates to the sparse signal $\hat{x}$.

```
Algorithm 1: Robust-$\ell_0$
Data: $A \in \mathbb{E}^{m \times n}_{k,\varepsilon,d}$; $y = Ax \in \mathbb{R}^m$ for $x \in \chi^n_k$; $\alpha \in (1, d]$; $\mu, \nu \in \mathbb{D}(\mathbb{R})$; $c \in (0, 1)$
Result: $\hat{x} \in \mathbb{R}^n$ s.t. $\hat{x} \approx x$
Estimate $\breve{p}_z = \breve{p}_z(d, k, m, n, \mu, \nu)$ as in (7);
Estimate $\breve{p}_e = \breve{p}_e(d, k, m, n, \mu, \nu)$ as in (7);
if quantised then
    Use *quantised* scores given in Table I.
else
    Use *continuous* scores given in Table I.
end
$\hat{x} \leftarrow 0$;
$\hat{r} \leftarrow y$;
$t \leftarrow 1$;
while *not converged and* $t > 0$ do
    $x' \leftarrow \hat{x}$;
    $r \leftarrow \hat{r}$;
    for $j \in [n]$ do
        for $i \in \mathcal{N}(j)$ do
            if $1 - \breve{p}_z(r_i) \geq t$ then
                $n_e \leftarrow \sum_{\ell \in \mathcal{N}(j)} q_e(r_i - r_\ell \mid t)$;
                $n_z \leftarrow \sum_{\ell \in \mathcal{N}(j)} q_z(r_\ell \mid t)$;
                $\omega \leftarrow \frac{1}{n_e} \sum_{\ell \in \mathcal{N}(j)} r_\ell q_e(r_i - r_\ell \mid t)$;
                if $\|r - \omega e_i\|_1 \leq \|r\|_1$ *and* $n_e - n_z \geq \alpha$ then
                    $x'_j \leftarrow x'_j + \omega$;
                end
            end
        end
    end
    $x' \leftarrow \mathcal{H}_k(x')$;
    $r \leftarrow y - Ax'$;
    $t \leftarrow t - c$;
    if $\|r\|_1 < \|\hat{r}\|_1$ then
        $\hat{x} \leftarrow x'$;
        $\hat{r} \leftarrow r$;
        if adaptive_k then
            $k_0 \leftarrow k - \sum_{j \in [n]} \breve{p}_z(\hat{x}_j)$;
            $k_0 \leftarrow \max(k_0, \lfloor \frac{m}{100} \rfloor)$;
            Recompute $\breve{p}_z = \breve{p}_z(k_0, m, n, \mu, \nu)$;
            Recompute $\breve{p}_e = \breve{p}_e(k_0, m, n, \mu, \nu)$;
        end
    end
end
```

A. Outline of the manuscript

The structure of this paper is as follows: Section II comprises a review of combinatorial compressed sensing and the Parallel-ℓ_0 decoding algorithm extended here. Robust-ℓ_0 decoding and the associated scores (8)-(9) are presented in Section III. In Section IV we present numerical experiments which demonstrate Robust-ℓ_0 to perform superior to a number of leading greedy and combinatorial compressed sensing algorithms.

II. Background: Combinatorial Compressed Sensing and ℓ_0-decoding

As mentioned in the previous section, the branch of combinatorial compressed sensing measures $x \in \chi_k^n$ with the adjacency matrix of an expander graph. These matrices are of very low complexity in terms of generation and storage, and also promise faster encoding and decoding than their dense counterparts, see Theorem II.2. In this section we review the basic elements of expander graphs and combinatorial compressed sensing.

There have been various algorithms proposed to reconstruct a sparse vector x from measurements $y = Ax$ when A is an expander matrix, see [17], [18], [20] and [19]; the work presented here starts with the Parallel-ℓ_0 algorithm and to improve this algorithm by making it robust to noisy measurements, results in Robust-ℓ_0 (Algorithm 1). The key observation for the Parallel-ℓ_0 algorithm is given by the following Lemma.

Algorithm 2: Parallel-ℓ_0 [1]

```
Data: $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$; $y = Ax \in \mathbb{R}^m$ for $x \in \chi_k^n$; $\alpha \in (1, d]$
Result: $\hat{x} \in \mathbb{R}^n$ s.t. $\hat{x} = x$
$\hat{x} \leftarrow 0$, $r \leftarrow y$;
while not converged do
    for $j \in [n]$ do
        $u \leftarrow 0$;
        for $i \in \mathcal{N}(j)$ do
            if $r_i \neq 0$ then
                $n_e \leftarrow |\{\ell \in \mathcal{N}(j) : r_i = r_\ell\}|$;
                $n_z \leftarrow |\{\ell \in \mathcal{N}(j) : r_\ell = 0\}|$;
                if $n_e - n_z \geq \alpha$ then
                    $u_j \leftarrow r_i$;
                end
            end
        end
    end
    $\hat{x} \leftarrow \hat{x} + u$;
    $r \leftarrow r - A\hat{x}$;
end
```

Lemma II.1. Let $y = Ax$, x dissociated, $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$ with $\varepsilon < \frac{1}{4}$. Then there exists a nonempty set $T \subset [n] \times \mathbb{R}$ such that

$$|\{i \in \mathcal{N}(j) : y_i = \omega\}| \geq (1 - 2\varepsilon)d \ \ \forall (j, \omega) \in T$$

and for every tuple in T that satisfies this property, we have $w = x_j$.

This means that at each iteration, if the residual r is non-zero, i.e. if we have not yet found the correct x, then there is a set of entries in x that we can change so that we reduce the number of non-zeros in r by at least $|T|(1 - 2\varepsilon)d$.

Theorem II.2 (Convergence of Algorithm 2 [1])**.** Let $A \in \mathbb{E}_{k,\varepsilon,d}^{m\times n}$ and let $\varepsilon \leq \frac{1}{4}$, and $x \in \chi_k^n$ be dissociated. Then, Parallel-ℓ_0 with $\alpha = (1-2\varepsilon)d$ can recover x from $y = Ax$ in $\mathcal{O}(\log k)$ iterations of complexity $\mathcal{O}(dn)$.

To put this result into context and show its applicability, we recall the remark after Definition I.3, stating that random matrices as considered in this work are indeed expander matrices with high probability.

We furthermore emphasize the fact that the algorithm is designed in a way that allows for massively parallel implementations.

III. Main contributions: ℓ_0-decoding for Noisy Measurements

We now consider the case where the measurements y are subject to additive noise, i.e. instead of $y = Ax$, we measure $\hat{y} = y + \eta$ where η is a realization of a random variable with $\eta_i \sim \nu$.

Parallel-ℓ_0 is not able to cope with additive noise, as it needs to make decisions whether a value in $\hat{y}$ is zero and whether two values in $\hat{y}$ are equal to each other. While for very small noise levels we could consider two values as equal if they are within a certain number of standard deviations, for larger noise levels the decision becomes more challenging. A discussed in Section I we need to know $p_z(\hat{y}_i)$ and $p_e(\hat{y}_{i_1} - \hat{y}_{i_2})$ which correspond, respectively, to the probability of $y_i = 0$ given that we observe $\hat{y}_i$ and the probability of $y_{i_1} = y_{i_2}$ given that we observe $\hat{y}_{i_1} - \hat{y}_{i_2}$. The functions p_z and p_e depend on the parameters fed into the generative model in Definition I.3 and in particular on the distribution of y and $\hat{y}$. Hence, since y is a vector of sparse inner products we need to understand the limiting behaviour of sparse sums.

We include Definition III.1 in order to remind the reader of some actions on measures used in this manuscript as so as to be relatively self contained.

Definition III.1 (Measures [23])**.** Let $\mathcal{B}(\mathbb{R})$ denote the Borel σ-algebra over $\mathbb{R}$. If $E \in \mathcal{B}(\mathbb{R})$ let $-E := \{-x : x \in E\}$. Let $\mu \in \mathbb{D}(\mathbb{R})$, we define the following measures.

1) The q-convolution,

$$\begin{aligned} \mu_0(E) &= \delta_0(E) = \begin{cases} 1 & 0 \in E \\ 0 & 0 \notin E \end{cases}, \forall E \in \mathcal{B}(\mathbb{R}) \\ \mu_1(E) &= \mu(E), \forall E \in \mathcal{B}(\mathbb{R}) \\ \mu_{q+1}(E) &= (\mu_q * \mu)(E), \forall E \in \mathcal{B}(\mathbb{R}), q \in \mathbb{N} \end{aligned}$$

2) The negative measure,

$$\mu^-(E) = \mu(-E), \ \ \forall E \in \mathcal{B}(\mathbb{R})$$

3) The symmetrized measure,

$$\bar{\mu}(E) = \frac{\mu(E) + \mu(-E)}{2}, \ \ \forall E \in \mathcal{B}(\mathbb{R})$$

4) The measure associated with the difference of two random variables,

$$\tilde{\mu}(E) = (\mu * \mu^-)(E), \ \ \forall E \in \mathcal{B}(\mathbb{R}).$$

Lemma III.2 (Limiting distribution for sparse sums of random variables)**.** Let $p \in (0, 1)$, let $\mu \in \mathbb{D}(\mathbb{R})$ and let $\mu_q \in \mathbb{D}(\mathbb{R})$ be its the q-fold convolution. For each $n \geq 1$, let

$$s_n := \sum_{j=1}^{n} b_j x_j \tag{10}$$

be such that,

1) $x_j \overset{\text{i.i.d.}}{\sim} \mu$ for each $j \in [n]$,
2) $b_j \overset{\text{i.i.d.}}{\sim} \text{Ber}(\frac{p_n}{n})$ for each $j \in [n]$ with $p_n \to p \in \mathbb{R}$ as $n \to \infty$.

Then, as $n \to \infty$ it holds that $s_n \overset{(d)}{\to} s$ where

$$s \sim \exp(-p) \sum_{q \geq 0} \frac{p^q}{q!} \mu_q. \tag{11}$$

Proof. Let $\psi_{s_n}(t)$ be the characteristic function of s_n. Let $x \sim \mu$, then

$$\begin{aligned} \psi_{s_n}(t) &= \mathbb{E}\left[\exp(its_n)\right] \\ &= \prod_{j=1}^{n} \mathbb{E}\left[\exp(itb_j x_j)\right] \\ &= \left(\left(1 - \frac{p_n}{n}\right) + \left(\frac{p_n}{n}\right)\psi_x(t)\right)^n \\ &= \left(1 + \frac{p_n(\psi_x(t) - 1)}{n}\right)^n. \end{aligned}$$

Taking the limit $n \to \infty$ we see that

$$\lim_{n\to\infty} \psi_{s_n}(t) = \exp(-p)\exp(p\psi_x(t)). \tag{12}$$

Letting $w_q = \sum_{j=1}^q x_j$ it holds by the independence of $\{x_1, \dots, x_q\}$ that $w_q \sim \mu_q$ and

$$[\psi_x(t)]^q = \psi_{w_q}(t). \tag{13}$$

Now, consider a random variable z distributed according to

$$z \sim \exp(-p) \sum_{q\geq 0} \frac{p^q}{q!}\mu_q. \tag{14}$$

The characteristic function of z is given by

$$\begin{aligned}\psi_z(t) &= \mathbb{E}\left[\exp(itz)\right] \\ &= \exp(-p)\sum_{q\geq 0}\frac{p^q}{q!}\psi_{w_q}(t) \\ &= \exp(-p)\sum_{q\geq 0}\frac{p^q}{q!}\left(\psi_x(t)\right)^q \\ &= \exp(-p)\sum_{q\geq 0}\frac{\left(p\psi_x(t)\right)^q}{q!} \\ &= \exp(-p)\exp(p\psi_x(t)). \end{aligned} \tag{15}$$

Therefore (15) equals (12). By Lévy's continuity Theroem pointwise convergence of the characteristic functions implies weak convergence of the random variables (cf. [23, Theorem 15.23]) and hence the statement follows. □

Theorem III.3 (Distribution of $\hat{y}_i$ and $\hat{y}_{i_1} - \hat{y}_{i_2}$)**.** Fix $\delta, \rho \in (0,1)$ and for $n \in \mathbb{N}$ let $m = \delta n$, $k = \rho m$ and $d \ll m$. Furthermore, let μ and ν be measures and assume that $\hat{y} = y + \eta = Ax + \eta$ is drawn from the model $\mathrm{GM}(n, m, k, d, \mu, \nu)$. Then as $n \to \infty$

$$\hat{y}_i \overset{(d)}{\to} \hat{y}_i^* \quad \text{and} \quad \hat{y}_{i_1} - \hat{y}_{i_2} \overset{(d)}{\to} \hat{g}^*$$

where

$$\hat{y}_i^* \sim \exp(-d\rho)\sum_{q\geq 0}\frac{(d\rho)^q}{q!}\nu * \mu_q, \tag{16}$$

$$\hat{g}^* \sim \exp(-2d\rho)\sum_{q\geq 0}\frac{(2d\rho)^q}{q!}\tilde{\nu} * \bar{\mu}_q. \tag{17}$$

Proof. To show (16), let $i \in [m]$ and

$$y_i = \sum_{j=1}^n A_{i,j}x_j.$$

By our assumptions on A and x,

$$\begin{aligned}\mathbb{P}\left(A_{i,j}x_j \neq 0\right) &= \mathbb{P}\left(A_{i,j} \neq 0 \wedge x_j \neq 0\right) \\ &= \mathbb{P}\left(A_{i,j} \neq 0\right)\mathbb{P}\left(x_j \neq 0\right) \\ &= \frac{d}{m}\frac{k}{n} \\ &= \frac{d\rho}{n}\end{aligned}$$

where for two events E_1 and E_2 we let $E_1 \wedge E_2$ be the conjunction of the events. Note that if $A_{i,j}x_j \neq 0$, then $j \in \mathrm{supp}(x)$ so $A_{i,j}x_j = x_j \sim \mu$. Hence, letting $b_j \sim \mathrm{Ber}\left(\frac{d\rho}{n}\right)$,

$$y_i \overset{(d)}{=} \sum_{j=1}^n b_j x_j.$$

Invoking Lemma III.2 with $p_n = p = d\rho$ we obtain that $y_i \to y_i^*$ as $n \to \infty$ where

$$y_i^* \sim \exp(-d\rho) \sum_{q \geq 0} \frac{(d\rho)^q}{q!} \mu_q.$$

By the independence of y_i^* and η_i, since $\hat{y}_i = y_i + \eta_i$, the distribution of $\hat{y}_i^*$ is given by

$$\hat{y}_i^* \sim \left(\exp(-d\rho) \sum_{q \geq 0} \frac{(d\rho)^q}{q!} \mu_q \right) * \nu. \tag{18}$$

Equation (16) follows from (18) and the distributivity of the convolution operator.

To show (17), let $i_1, i_2 \in [m]$ be such that $i_1 \neq i_2$ and let

$$y_{i_1} - y_{i_2} = \sum_{j=1}^{n} \left(A_{i_1,j} - A_{i_2,j}\right) x_j.$$

Similarly to the previous case, we compute

$$\begin{aligned} \mathbb{P}\left(\left(A_{i_1,j} - A_{i_2,j}\right) x_j \neq 0\right) &= \mathbb{P}\left(A_{i_1,j} - A_{i_2,j} \neq 0 \wedge x_j \neq 0\right) \\ &= \mathbb{P}\left(A_{i_1,j} - A_{i_2,j} \neq 0\right) \mathbb{P}\left(x_j \neq 0\right) \\ &= 2\frac{d}{m} \frac{(m-1)-(d-1)}{m-1} \frac{k}{n} \\ &= \frac{2d\rho}{n}\left(1 - o(1)\right) \end{aligned}$$

Note that if $(A_{i_1,j} - A_{i_2,j})x_j \neq 0$, then $j \in \operatorname{supp}(x)$ and $(A_{i_1,j} - A_{i_2,j})$ is either $+1$ with probability $\frac{1}{2}$ or -1 with probability $\frac{1}{2}$. Then, Hence,

$$(A_{i_1,j} - A_{i_2,j})x_j \sim \left\{ \begin{array}{ll} \mu & \text{with probability } \frac{1}{2}, \\ \mu^- & \text{with probability } \frac{1}{2}, \end{array} \right.$$

then,

$$(A_{i_1,j} - A_{i_2,j})x_j \sim \bar{\mu}.$$

Letting $b'_j \sim \operatorname{Ber}\left(\frac{2d\rho}{n}(1 - o(1))\right)$,

$$y_{i_1} - y_{i_2} \overset{(d)}{=} \sum_{j=1}^{n} b'_j x_j.$$

Again, invoking Lemma III.2 with $p_n = 2d\rho(1 - o(1))$ we obtain that $p = 2d\rho$ and also that as $n \to \infty$, $y_{i_1} - y_{i_2} \to g^*$ with

$$y_{i_1}^* - y_{i_2}^* \sim \exp(-2d\rho) \sum_{q \geq 0} \frac{(2d\rho)^q}{q!} \bar{\mu}_q. \tag{19}$$

Therefore, Given that $\eta_{i_1} - \eta_{i_2} \sim \nu * \nu^-$ and that

$$\hat{y}_{i_1} - \hat{y}_{i_2} = \left(y_{i_1} - y_{i_2}\right) + \left(\eta_{i_1} - \eta_{i_2}\right),$$

we convolve (19) with $\nu * \nu^-$ to recover (17). □

We are now ready to prove Theorem I.4,

Proof. Theorem I.4. Using Bayes rule we write

$$\mathbb{P}(y_i = 0 | \hat{y}_i = \omega) = \frac{\mathbb{P}(\hat{y}_i = \omega \wedge y_i = 0)}{\mathbb{P}(\hat{y}_i = \omega)}. \tag{20}$$

From (16) we can deduce that

$$\mathbb{P}(\hat{y}_i = \omega \wedge y_i = 0) = \exp(-d\rho)\nu(\omega) \tag{21}$$

and using equation (16) from Theorem III.3 we obtain that as $n \to \infty$,

$$\mathbb{P}(\hat{y}_i = \omega) \to \exp(-d\rho) \sum_{q \geq 0} \frac{(d\rho)^q}{q!} (\nu * \mu_q)(\omega). \tag{22}$$

Coupling (20), (22) and (21) yields (5).

Again, by Bayes rule,

$$\mathbb{P}\left(y_{i_1} = y_{i_2} | \hat{y}_{i_1} - \hat{y}_{i_2} = \omega\right) = \frac{\mathbb{P}\left(y_{i_1} = y_{i_2} \wedge \hat{y}_{i_1} - \hat{y}_{i_2} = \omega\right)}{\mathbb{P}\left(\hat{y}_{i_1} - \hat{y}_{i_2} = \omega\right)}. \tag{23}$$

Noting that $\hat{y}_{i_1} - \hat{y}_{i_2} = (y_{i_1} - y_{i_2}) + (\eta_{i_1} - \eta_{i_2})$,

$$\mathbb{P}\left(y_{i_1} = y_{i_2} \wedge \hat{y}_{i_1} - \hat{y}_{i_2} = \omega\right) = \tilde{\nu}(\omega) \tag{24}$$

By (16) from Theorem III.3 we obtain that as $n \to \infty$,

$$\mathbb{P}(\hat{y}_{i_1} - \hat{y}_{i_2} = \omega) \to \exp(-2d\rho) \sum_{q \geq 0} \frac{(2d\rho)^q}{q!} \tilde{\nu} * \bar{\mu}_q(\omega) \tag{25}$$

Coupling (23), (24), and (25) yields (6).

□

A. Explicit formulas for the centred Gaussian case

We further elucidate (5)-(6) from Theorem I.4 in the case when μ and ν are Gaussian with mean zero, which are the distributions considered in Section IV. To this end, let $\mu = \mathcal{N}(0, \sigma_s^2)$ and $\nu = \mathcal{N}(0, \sigma_n^2)$. We observe that in this case $\bar{\mu} = \mu$, $\mu_q = \bar{\mu}_q = \mathcal{N}(0, q\sigma_s^2)$ and $\tilde{\nu} = \mathcal{N}(0, 2\sigma_n^2)$. Hence, denoting by $\varphi(\cdot \mid \sigma^2)$ the probability density function of a centred Gaussian random variable with variance σ^2, we obtain

$$p_z(\omega) \to \frac{\varphi(\omega \mid \sigma_n^2)}{\sum_{q \geq 0} \frac{(d\rho)^q}{q!} \varphi(\omega \mid q\sigma_s^2 + \sigma_n^2)} \tag{26}$$

$$p_e(\omega) \to \frac{\varphi(\omega \mid 2\sigma_n^2)}{\sum_{q \geq 0} \frac{(2d\rho)^q}{q!} \varphi(\omega \mid q\sigma_s^2 + 2\sigma_n^2)}. \tag{27}$$

1) Estimating the tails in the Gaussian case: We approximate the infinite sum in in the denominators of (26) and (27) by doing an approximation to the tail of this summation.

Lemma III.4 (Sums of centred density functions)**.** Let μ_i be the density function of a random varible with mean zero and variance σ_i^2 and let $\alpha_i > 0$ be such that $\sum_i \alpha_i = 1$. Then, $\mu = \sum_i \alpha_i \mu_i$ is the density function of a random variable with mean zero and variance $\sum_i \alpha_i \sigma_i^2$.

Proof. Let $x \sim \mu$ and $x_i \sim \mu_i$ be such that $\mathbb{E}\left[x_i\right] = 0$ and $\mathbb{V}ar\left[x_i\right] = \sigma_i^2$.

$$\begin{aligned}
\mathbb{V}ar\left[x\right] &= \mathbb{E}[x^2] \\
&= \int \omega^2 \mu(\omega) d\omega \\
&= \int \omega^2 \left(\sum_i \alpha_i \mu_i(\omega) \right) d\omega \\
&= \sum_i \alpha_i \int \omega^2 \mu_i(\omega) d\omega \\
&= \sum_i \alpha_i \sigma_i^2
\end{aligned}$$

□

To simplify notation let $\sigma_q^2 = q\sigma_s^2 + \sigma_n$ for $q \in \mathbb{N} \cup \{0\}$. Consider the series in the denominator of (22)

$$S(\omega) := \sum_{q=0}^{\ell} \frac{(d\rho)^q}{q!} \varphi(\omega \mid \sigma_q^2) + \sum_{q=\ell+1}^{\infty} \frac{(d\rho)^q}{q!} \varphi(\omega \mid \sigma_q^2).$$

Let,

$$R_z(\ell) := \exp(d\rho) - \sum_{q=0}^{\ell} \frac{(d\rho)^q}{q!}$$

and write

$$S_a(\omega) := \sum_{q=0}^{\ell} \frac{(d\rho)^q}{q!}\varphi(\omega \mid \sigma_q^2),$$

$$S_b(\omega) := \sum_{q=\ell+1}^{\infty} \frac{(d\rho)^q}{q!}\varphi(\omega \mid \sigma_q^2).$$

Note that S_b/R_z satisfies the conditions of Lemma III.4 so it corresponds to the density function of a centred random variable with variance

$$\begin{aligned}\sigma_z^2 &= \frac{1}{R_z}\sum_{q=\ell+1}^{\ell} \frac{(d\rho)^q}{q!}(q\sigma_s^2 + \sigma_n^2) \\ &= \frac{\sigma_s^2(d\rho)R_z(\ell-1) + \sigma_n^2 R_z(\ell)}{R_z(\ell)}\end{aligned}$$

Therefore,

$$p_z(\omega) \approx \frac{\varphi(\omega \mid \sigma_n^2)}{\sum_{q=0}^{\ell} \frac{(d\rho)^q}{q!}\varphi(\omega \mid \sigma_q^2) + R_z(\ell)\varphi(\omega \mid \sigma_z^2)}. \tag{28}$$

A similar argument shows that

$$p_e(\omega) \approx \frac{\varphi(\omega \mid 2\sigma_n^2)}{\sum_{q=0}^{\ell} \frac{(2d\rho)^q}{q!}\varphi(\omega \mid \sigma_{q,e}^2) + R_e(\ell)\varphi(\omega \mid \sigma_e^2)}. \tag{29}$$

Where $\sigma_{q,e}^2 = q\sigma_s^2 + \sigma_n^2$, and

$$R_e(\ell) = \exp(2d\rho) - \sum_{q=0}^{\ell} \frac{(d\rho)^q}{q!},$$

$$\sigma_e^2 = \frac{\sigma_s^2(2d\rho)R_e(\ell-1) + 2\sigma_n^2 R_e(\ell)}{R_e(\ell)}.$$

B. Comparison with empirical probabilities

We test the approximations given in (28) and (29) by randomly generating $\hat{y}$ and y according the generating models $\mathrm{GM}(n, \delta n, \rho\delta n, 7, \mathcal{N}(0,1), \mathcal{N}(0,\sigma^2))$, for $\delta = 0.3$, $\rho \in \{0.1, 0.3\}$ and $\sigma \in \{10^{-3}, 10^{-2}\}$. The results can be seen in Figure 2.

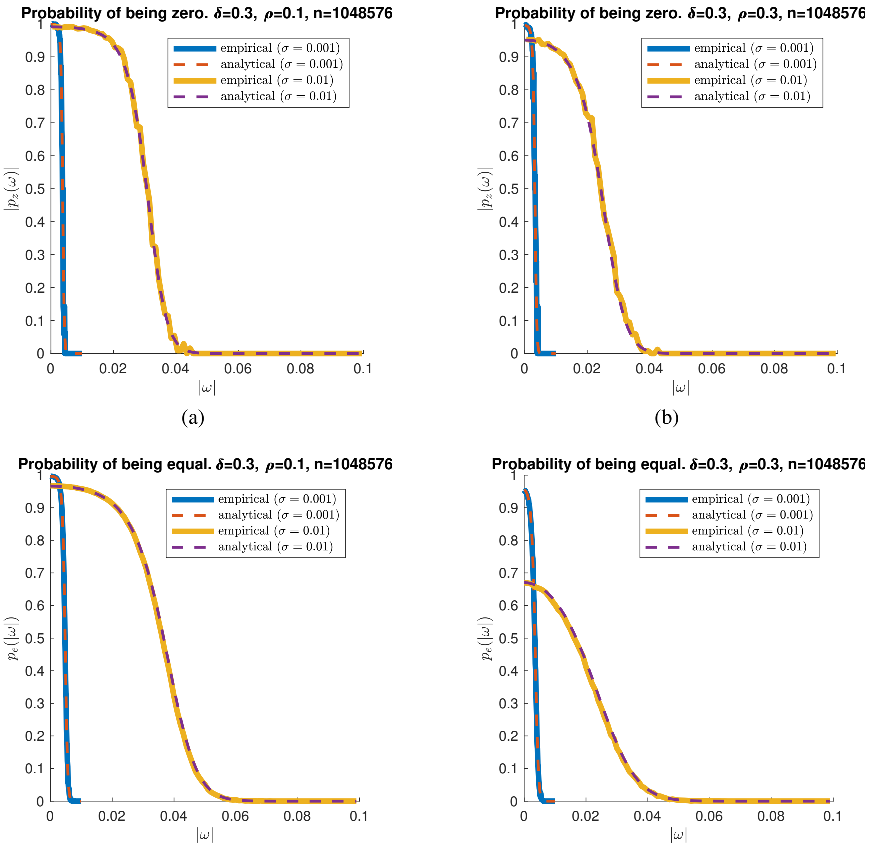
Overall the analytical expressions fit the empirical probabilities very well, indicating that the approximations we made in the calculations above are justified. However, we observe that as ρ and especially σ increase, both functions drop significantly. This means that for large values of these parameters, the noise eventually dominates and it is difficult to decided whether values are zero or equal.

C. Scaled probabilities $\breve{p}_z$ and $\breve{p}_e$

We mentioned in Section I that our algorithms do not implement the functions p_z and p_e exactly, but a scaled version of these. As observed in Figure 2, the value of $\max_s p_e(s)$ and $\max_s p_z(s)$ varies significantly as σ and ρ change. Algorithm 1 evaluates whether a given score is large or not by implementing a sweeping parameter t that is set to one at the beginning of the algorithm and decreased by a constant c after every iteration. In order to use a fixed initial t we consider the scaled probabilities $\breve{p}_e$ and $\breve{p}_z$ in (7); otherwise the initial value of t would depend on σ and ρ.

D. Adaptive k

Algorithm 1 can optionally account for the sparsity of the current estimate via the flag `adaptive_k`. In the noiseless model of $r = A\hat{x}$, an update of the form $\hat{x} \leftarrow \hat{x} + \omega e_j$ with Parallel-ℓ_0 guarantees that $j \in \mathrm{supp}(x)$ so that at the next iteration the problem with residual $r - a_j\omega$ and $(k-1)$-sparse signal is considered. The `adaptive_k` flag updates the sparsity prior in $\hat{x}$ after every update in hope of having more reliable estimates of p_e and p_z. We will see in the numerical experiments that under the data generating model that we tested this strategy does not bring substantial benefits. We don't rule out the posssibility that there are other signal and noise distributions for which this flag becomes especially useful, but we leave that as future work.



(a)

(b)

(c)

(d)

Fig. 2: Comparison of analytical and empirical probabilities of a value in the residual being zero or two values in the residual being equal for $\rho \in \{0.1, 0.3\}$ and $\sigma_n \in \left\{10^{-3}, 10^{-2}\right\}$.

IV. Numerical experiments

In this section we present numerical experiments which validate the efficacy of Robust-ℓ_0 decoding. In particular, we contrast Robust-ℓ_0 with other state-of-the-art greedy algorithms for compressed sensing in terms of their ability to recover the measured signal for varying problem sizes (k, m, n) as well as their computational complexity. To facilitate reproducibility we begin by describing the stopping conditions and measures used to denote successful recovery in the presence of noise in Section IV-A, along with how the parameter c is varied in Section IV-B. We then present the main numerical results in Section IV-C where the algorithms phase transitions and runtime are presented, along with Sections IV-D and IV-E which show further details on Robust-ℓ_0 decoding's performance as a function of noise variance and for extreme subsampling respectively.

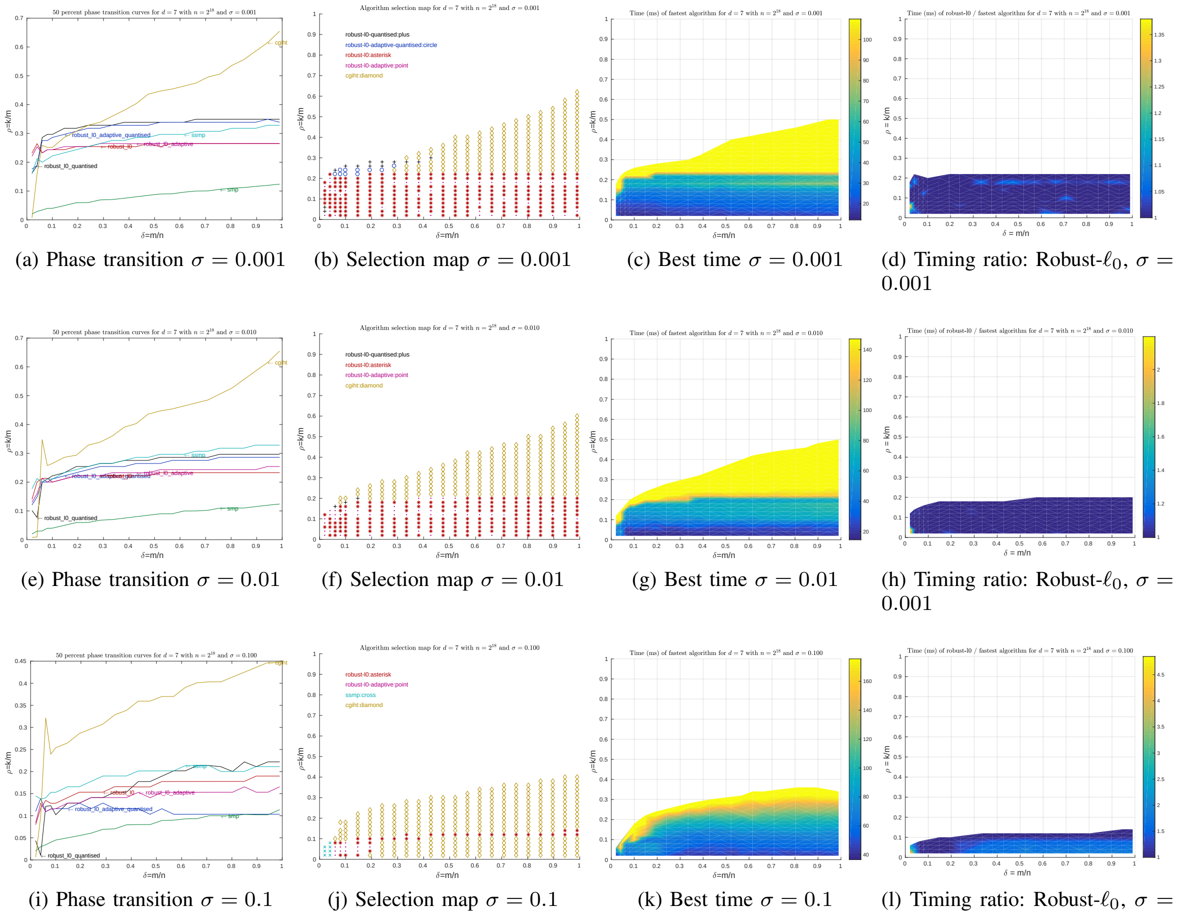
A. *Stopping conditions*

We are interested in the signal model $y = Ax + \eta$ where $\eta \sim \mathcal{N}(0, \sigma^2 I_{m \times m})$. If $\hat{x}$ is an approximation to x the residual is $r = y - A\hat{x}$. Note that if $\hat{x} = x$, then

$$\begin{aligned} \|r\|_1 &= \|y - A\hat{x}\|_1 \\ &= \|y - Ax\|_1 \\ &= \|\eta\|_1 \end{aligned}$$

and we should not seek reductions in the residual below $\|\eta\|_1$ which would result in fitting to the additive noise. We further account for the variance of $\|\eta\|_1$ and denote the algorithm to have successfully recovered x if $\hat{x}$ satisfies

$$\frac{\|x - \hat{x}\|_1}{\|x\|_1} \leq \frac{\mathbb{E}\left[\|\eta\|_1\right] + C_1 \sqrt{\mathrm{Var}\left[\|\eta\|_1\right]}}{\|x\|_1} \tag{30}$$



(a) Phase transition $\sigma = 0.001$ (b) Selection map $\sigma = 0.001$ (c) Best time $\sigma = 0.001$ (d) Timing ratio: Robust-ℓ_0, $\sigma = 0.001$

(e) Phase transition $\sigma = 0.01$ (f) Selection map $\sigma = 0.01$ (g) Best time $\sigma = 0.01$ (h) Timing ratio: Robust-ℓ_0, $\sigma = 0.001$

(i) Phase transition $\sigma = 0.1$ (j) Selection map $\sigma = 0.1$ (k) Best time $\sigma = 0.1$ (l) Timing ratio: Robust-ℓ_0, $\sigma = 0.001$

Fig. 3: Phase transitions, selection maps and timings for $n = 2^{18}$.

for some $C_1 \geq 0$. We should be aware that the right hand side of (30) might be greater than 1 for some choices of k, m and σ. When this happens, the stopping condition (30) becomes invalid since we expect $\hat{x}$ to have captured a proportion of the ℓ_1-energy of $\|x\|_1$. Hence, if the right hand side of (30) is greater than $\frac{1}{10}$ we clip the upper bound at this value and use the stopping condition

$$\frac{\|x - \hat{x}\|_1}{\|x\|_1} \leq \min\left(\frac{\mathbb{E}\left[\|\eta\|_1\right] + C_1\sqrt{\mathrm{Var}\left[\|\eta\|_1\right]}}{\|x\|_1}, \frac{1}{10}\right). \tag{31}$$

For the numerical experiments conducted in this section we consider nonzero entries in x drawn as $x_i \sim \mathcal{N}(0,1)$, and noise $\eta_i \sim \mathcal{N}(0,\sigma^2)$ for which $\mathbb{E}\left[\|x\|_1\right] = k\sqrt{\frac{2}{\pi}}$ and $\mathbb{E}\left[\|\eta\|_1\right] = m\sigma\sqrt{\frac{2}{\pi}}$ and moreover $\mathrm{Var}\left[\|\eta\|_1\right] = m\sigma^2\left(1 - \frac{2}{\pi}\right)$, see e.g. [24].

B. Selection of parameter c in Algorithm 1

The sweeping parameter t in Algorithm 1 is initialised at 1 and updated by decreasing it by a constant value c. We observed in our experiments that the quality of the phase transitions are sensitive on the parameter c especially for low δ and ρ. We do not provide a way to fine-tune c, but we run our phase transitions with the following choices:

1) If the algorithm is `quantised`,

$$c = \begin{cases} 0.01 & \delta \leq 0.05 \\ 0.05 & \delta > 0.05 \text{ and } \rho \in (0, 0.1] \\ 0.075 & \delta > 0.05 \text{ and } \rho \in (0.1, 0.2] \\ 0.1 & \delta > 0.05 \text{ and } \rho \in (0.2, 1) \end{cases}. \tag{32}$$

Algorithm	adaptive_k	quantised
Robust-ℓ_0	No	No
Robust-ℓ_0-adaptive	Yes	No
Robust-ℓ_0-quantised	No	Yes
Robust-ℓ_0-adaptive-quantised	Yes	Yes

TABLE II: Variants of Robust-ℓ_0

2) If the algorithm is `continuous`,

$$c = \begin{cases} 0.01 & \delta \leq 0.05 \\ 0.025 & \delta > 0.05 \end{cases} . \tag{33}$$

The values in (32) and (33) were chosen heuristically for ν and μ Gaussian.

C. Phase transitions and runtime

We benchmark the variants of Robust-ℓ_0 against other greedy algorithms via their *phase-transitions and runtime*. The user can supply two binary flags, `adaptive_k` and `quantised` which yield four different variants of Robust-ℓ_0. We assigned a unique label to each of these variants as described in Table II.

The phase transition of a compressed-sensing algorithm [25] is the largest value of k/m for which the algorithm is typically able recovery all k sparse vectors with sparsity less than k for a fixed m/n. Hence, for a fixed value of $\delta = m/n$ the phase transition of an algorithm is the largest value $\rho^*(\delta)$ for which the algorithm converges for all $\rho(\delta) < \rho^*(\delta)$. The value $\rho^*(m/n)$ often converges to a fixed value as $n \to \infty$, so phase transitions often partition the $\delta \times \rho$ space into two regions: One in which the algorithm converges with high probability and another in which the algorithm doesn't converge with high probability. We benchmark Robust-ℓ_0 against the algorithms presented in [18], [19], [26]. Specifically, our tests include the following algorithms,

$$\{\text{Robust-}\ell_0,\ \text{Robust-}\ell_0\text{-adaptive},\ \text{Robust-}\ell_0\text{-quantised},\ \text{Robust-}\ell_0\text{-adaptive-quantised},\ \text{SSMP},\ \text{SMP},\ \text{CGIHT}\}.$$

In the deterministic case Parallel-ℓ_0 was compared against a range of combinatorial compressed sensing algorithms in [1]; out of those we have selected SSMP and SMP as these perform best and are similar in nature to Robust-ℓ_0. We also compare with CGIHT, as this algorithm was shown to be the fastest among the greedy algorithms compared in [13], [26]. Figures 3a, 3e and 3i show the phase transition curves for these algorithms with $\sigma = 10^{-3}$, 10^{-2}, 10^{-1} respectively. The curves were computed by setting $n = 2^{18}$, $d = 7$, and using the stopping condition $\|r\|_1 \leq \mathbb{E}\left[\|\eta\|_1\right] = m\sigma\sqrt{\frac{2}{\pi}}$ and a success condition (30) with $C_1 = 1$. The testing is done at $m = \delta_p n$ for

$$\delta_p \in \{0.02p : p \in [4]\} \cup \left\{0.1 + \frac{89}{1900}(p-1) : p \in [20]\right\}.$$

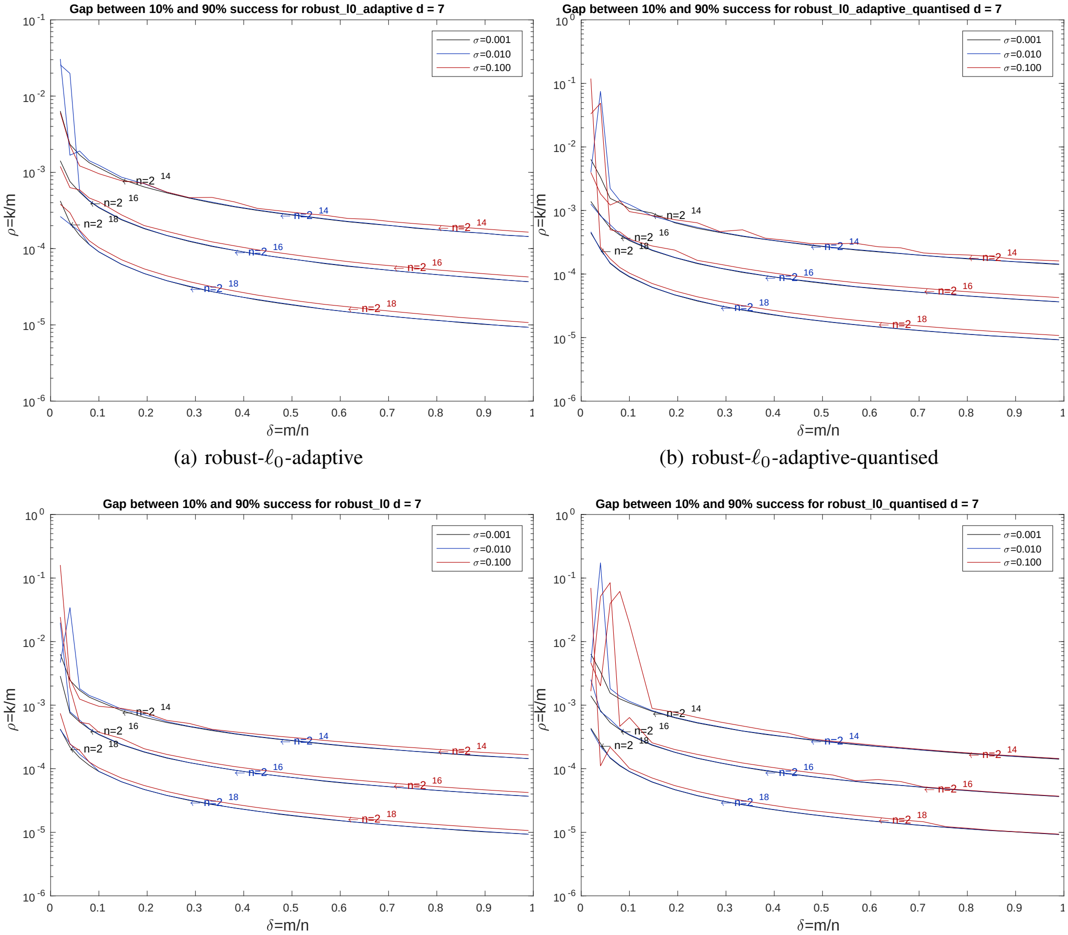
For each δ_p, we set $\rho = 0.01$ and generate 10 synthetic problems to apply the algorithms to, with a problem generated as in GM with the given parameters and μ and ν being normal Gaussian. If at least one such problem was recovered successfully, the sparsity ratio ρ is increased by 0.01 and the experiment is repeated. Following the testing framework in [27], the recovery data is fitted using a logistic function and finally the 50% recovery transition function is computed and presented in the phase transition plots contained herein. Figures 3b, 3f and 3j show a selection map for these algorithms. Namely, these plots indicate which algorithm requires the least computational time[1] at each point in the $\delta \times \rho$ space where the algorithm converges. Finally, Figures 3c, 3g and 3k show the total time for convergence in milliseconds for the fastest algorithm at each point in the $\delta \times \rho$ space. The ratio of the time for the second fastest algorithm over the time for the fastest algorithm is given in Figures 3d, 3h, and 3l.

We can see from Figure 3 that CGIHT [26] dominates the upper region of the phase transition space, while the Robust-ℓ_0 algorithms only converge for $\rho \lessapprox 0.3$ which is consistent with the observed phase transitions for Parallel-ℓ_0 [1]. In terms of speed, Robust-ℓ_0 seems to be the most competitive for $\sigma \in \{10^{-3}, 10^{-2}\}$ and $\rho \lessapprox 0.2$. However, for large noise, $\sigma = 10^{-1}$, CGIHT becomes the fastest algorithm of all. We remark that

[1] All the numerical results presented here were performed using a Linux machine with Intel Xeon E5-2643 CPUs 3.30 GHz, NVIDIA Tesla K10 GPUs and executed from Matlab R2016b. The code was added to the GAGA library available at `http://www.gaga4cs.org/`, and described in [28], so as to facilitate large scale benchmarking against the other algorithms presented here which are also contained in the aforementioned library.

while CGIHT performs very well in our numerical tests, the current theory developed for it does not hold in the setting considered here, as it requires zero-mean columns in A.

Figure 4 shows the widths for the Robust-ℓ_0 algorithms. The widths measure how sharp the phase transition of an algorithm is; namely, how thin the boundary between the region of recovery with high-probability and the region of recovery where combinatorial search is needed. It has been shown that the widths of a compressed sensing algorithm tend to zero as $n \to \infty$ when decoding with linear programming [29], and we usually expect the same behaviour for other algorithms [26]. Figure 4 show that the widths for the Robust-ℓ_0 algorithms indeed decrease with n and with δ. The observed smoothness of the phase transition widths signal also suggest that the stopping conditions of the algorithm are consistent for the problem under consideration.



(a) robust-ℓ_0-adaptive

(b) robust-ℓ_0-adaptive-quantised

(c) robust-ℓ_0

(d) robust-ℓ_0-quantised

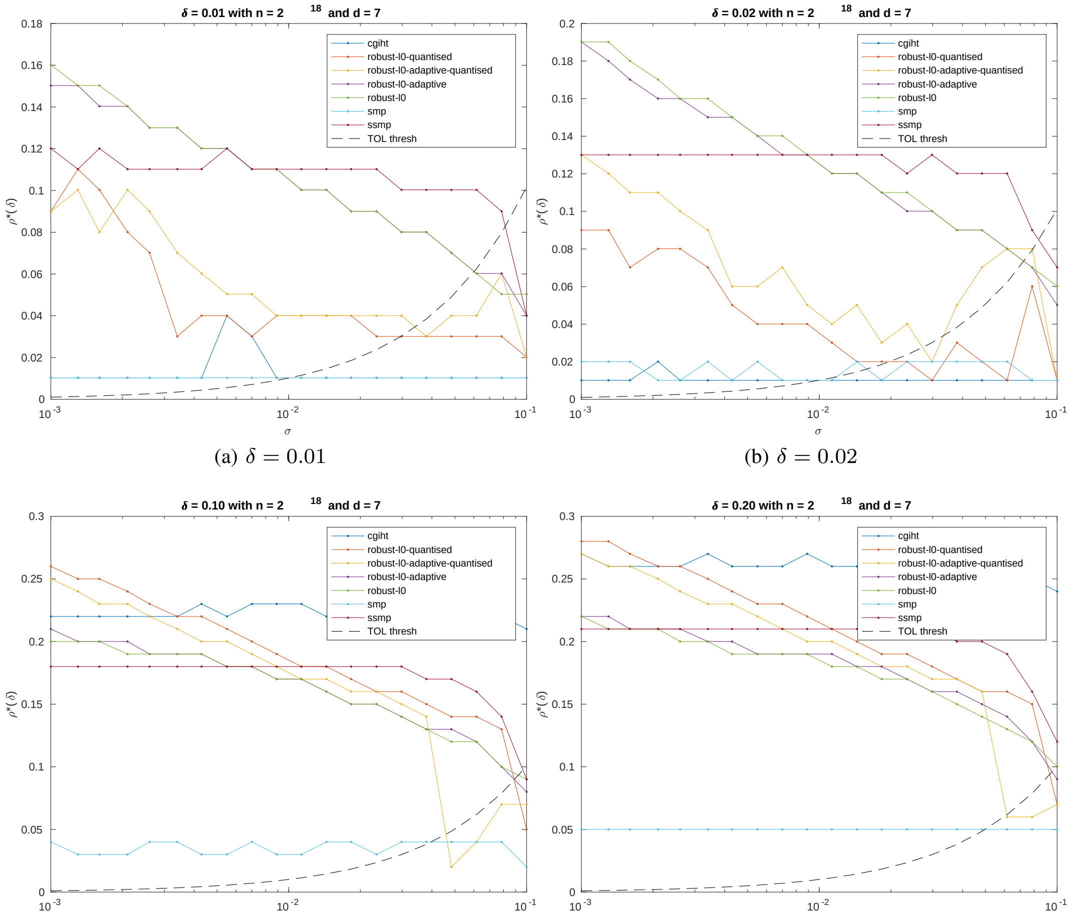
Fig. 4: Widths for Robust-ℓ_0 variants

D. Dependence on noise variance, σ

We now investigate the extent to which the phase transitions of the algorithm decrease as we increase the noise level σ. To do this, we consider $\delta \in \{0.01, 0.02, 0.1, 0.2\}$ and for each value of δ we compute we define the grid

$$\sigma \in \{10^{-3+\frac{i}{10}} : i \in \{0\} \cup [20]\}.$$

Then at each value of σ we let $\rho = 0.01$ and draw ten problem instances from GM with signal distribution $\mathcal{N}(0,1)$ and noise distribution $\mathcal{N}(0,\sigma^2)$. If at least one of the problems was recovered successfully, then we set $\rho \leftarrow \rho + 0.01$ and repeat the experiment. We do this procedure for each of the algorithms considered and record the largest ρ having at least 50% success rate. The results are shown in Figure 5. In order to show where the clipping in (31) becomes active, the figures also show a `TOL`-curve which partitions the space into



(a) $\delta = 0.01$

(b) $\delta = 0.02$

(c) $\delta = 0.10$

(d) $\delta = 0.20$

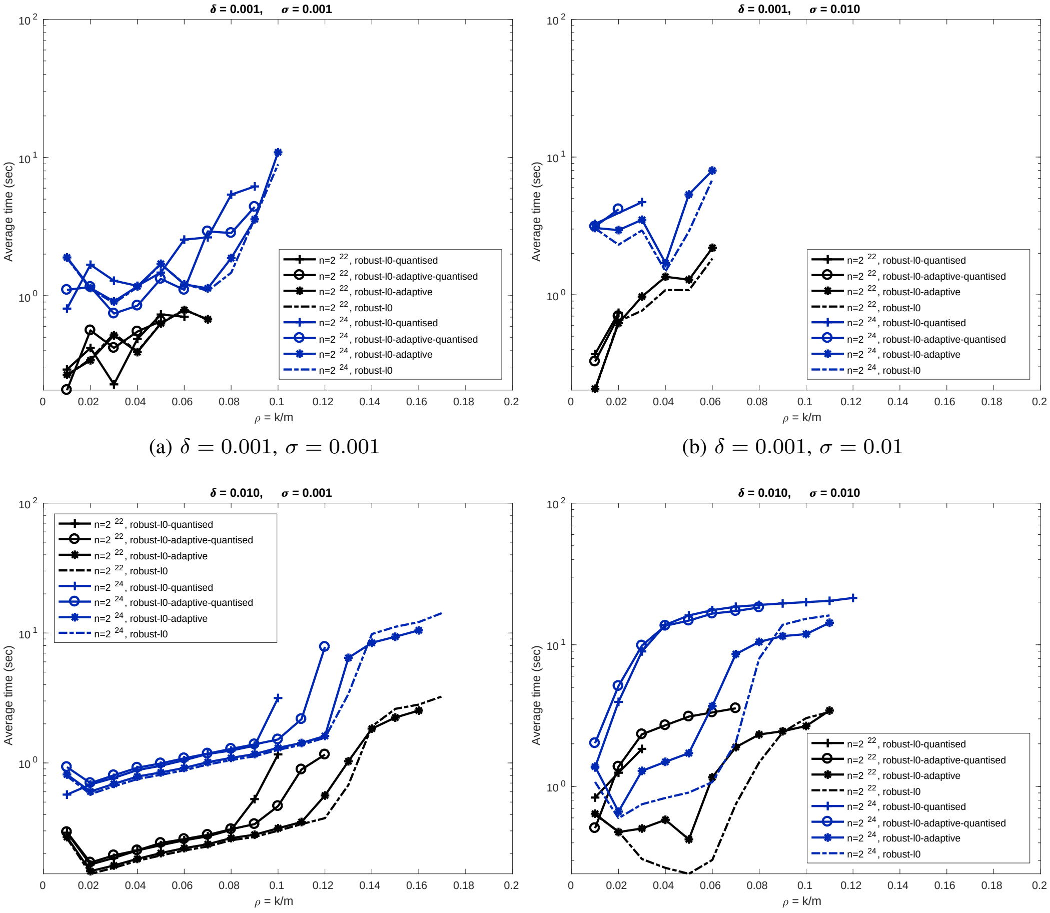
Fig. 5: Decrease in phase transition for varying σ.

the region where the right hand side of (31) equals $\frac{1}{10}$ (bottom region) and the region where it equals the right hand side of (30) (top region). We can appreciate from Figure 5 that for small δ both Robust-ℓ_0 and SSMP have the best recovery capabilities, with Robust-ℓ_0 being preferable for noise levels $\sigma \lessapprox 10^{-2}$ and SSMP being better suited for larger noise levels. For larger δ, CGIHT is preferable except for very low noise levels.

E. Phase transitions for extreme subsampling, $\delta \ll 1$

The numerical experiments of Parallel-ℓ_0 in [1] showed flat phase transitions; that is, it was observed that $\rho^*(\delta)$ remained approximately 0.3 as $\delta \to 0$ provided n was sufficiently large. While Robust-ℓ_0 does not exhibit precisely the same behaviour in the presence of noise, we do observe that $\rho^*(\delta)$ remains nontrivial even for δ as small as 10^{-3}, again provided n is sufficiently large. We provide numerical evidence for this in Figure 6. For each $(\delta, \sigma) \in \{0.001, 0.01\} \times \{0.001, 0.01\}$ we let $\rho = 0.01$ and solve ten problems drawn from GM with nonzero distribution $\mathcal{N}(0,1)$ and noise distribution $\mathcal{N}(0,\sigma^2)$. If at least one problem instance converges, we average the run-time of the problems that converged and repeat the process with $\rho \leftarrow \rho + 0.01$. We plot the timing at each ρ for parameter values for which at which at least 50% of the problems were successfully recovered under the criteria (31).

It can be seen in Figure 6a-6d that the phase transition either remains nearly unchanged or increases as n increases from 2^{22} to 2^{24}. In particular, Figure 6a shows that for $\sigma = 0.001$ and $\delta = 0.001$, the phase transitions for all variants of the Robust-ℓ_0 algorithms in fact increase from $\rho \approx 0.08$ to $\rho \approx 0.1$ as n increases from 2^{22} to 2^{24}. Additionally, contrasting Figures 6a and 6b or 6c and 6d shows that a ten-fold increase in σ has the expected effect of reducing the phase transition and increasing the computational time. Figures 6a and 6b show the phase transition for Robust-ℓ_0 and Robust-ℓ_0-adaptive remain significant even for δ as small as 10^{-3}. Figures 6c and 6d show results for the same set of experiments, but for $\delta = 0.01$ which corresponds to a ten-fold increase in δ over the value used in Figures 6a and 6b. For $\sigma = 0.001$ the phase transition of Robust-ℓ_0 reaches $\rho \approx 0.17$, while for $\sigma = 0.01$ the phase transitions drop to $\rho \approx 0.12$ and there is an increase in the computational time.



(a) $\delta = 0.001$, $\sigma = 0.001$

(b) $\delta = 0.001$, $\sigma = 0.01$

(c) $\delta = 0.01$, $\sigma = 0.001$

(d) $\delta = 0.01$, $\sigma = 0.01$

Fig. 6: Phase transitions and timing Robust-ℓ_0 for $\delta \ll 1$.

V. Conclusions

We have shown that the decoding framework presented in [1] can be extended to the case where the measurements are corrupted by additive noise. This framework is extended by deriving the posterior distribution of an entry in the residual being zero or being equal to another residual entry given the corrupted measurements. This Bayesian approach to decoding was implemented in Robust-ℓ_0 and its four variants. We show that the resulting algorithms inherits some desirable properties from Parallel-ℓ_0 like high phase transitions for low δ and large n and low-latency. However, these qualities are weakened by the corruption of the measurements. Our numerical experiments show that Robust-ℓ_0 should be considered in cases of moderate noise and $\rho \lessapprox 0.3$.

Acknowledgments

The authors would like to thank William Carson for many helpful discussions on imaging applications of this work. This work was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1.

Index Terms

compressed sensing, expander graphs, dissociated signals, robust algorithms.

REFERENCES

[1] R. Mendoza-Smith and J. Tanner, “Expander l0-decoding,” *Applied and Computational Harmonic Analysis*, pp. –, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1063520317300210

[2] E. J. Candes and T. Tao, “Decoding by linear programming,” *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec 2005.

[3] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[4] D. Donoho, V. Stodden, Y. Tsaig *et al.*, “Sparselab,” *SparseLab: Seeking Sparse Solutions to Linear Systems of Equations, SparseLab toolbox shared online, http://sparselab. stanford. edu/, 24th August*, 2007.

[5] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, “An interior-point method for large-scale ℓ_1-regularized least squares,” *IEEE journal of selected topics in signal processing*, vol. 1, no. 4, pp. 606–617, 2007.

[6] M. A. Figueiredo, R. D. Nowak, and S. J. Wright, “Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems,” *IEEE Journal of selected topics in signal processing*, vol. 1, no. 4, pp. 586–597, 2007.

[7] E. Candes and J. Romberg, “l1-magic: Recovery of sparse signals via convex programming,” *URL: www. acm. caltech. edu/l1magic/downloads/l1magic. pdf*, vol. 4, p. 14, 2005.

[8] T. Blumensath and M. E. Davies, “Iterative hard thresholding for compressed sensing,” *CoRR*, vol. abs/0805.0510, 2008.

[9] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition,” pp. 40–44 vol.1, Nov 1993.

[10] D. Needell and J. A. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,” *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[11] T. Blumensath and M. E. Davies, “Normalized iterative hard thresholding: Guaranteed stability and performance,” *IEEE Journal of selected topics in signal processing*, vol. 4, no. 2, pp. 298–309, 2010.

[12] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing signal reconstruction,” *IEEE Trans. Inform. Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[13] J. D. Blanchard, J. Tanner, and K. Wei, “CGIHT: Conjugate gradient iterative hard thresholding for compressed sensing and matrix completion,” *Information and Inference*, vol. 4, no. 4, pp. 289–327, 2015.

[14] V. Cevher, “An ALPS view of sparse recovery,” in *Acoustics Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 2011, pp. 5808 –5811.

[15] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*. Birkhäuser Basel, 2013, vol. 1, no. 3.

[16] S. Sarvotham, D. Baron, and R. G. Baraniuk, “Sudocodes — fast measurement and reconstruction of sparse signals,” in *2006 IEEE International Symposium on Information Theory*, July 2006, pp. 2804–2808.

[17] W. Xu and B. Hassibi, “Efficient compressive sensing with deterministic guarantees using expander graphs,” pp. 414–419, Sept 2007.

[18] R. Berinde, P. Indyk, and M. Ruzic, “Practical near-optimal sparse recovery in the l1 norm,” pp. 198–205, Sept 2008.

[19] R. Berinde and P. Indyk, “Sequential sparse matching pursuit,” pp. 36–43, Sept 2009.

[20] S. Jafarpour, W. Xu, B. Hassibi, and R. Calderbank, “Efficient and robust compressed sensing using optimized expander graphs,” *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 4299–4308, Sept 2009.

[21] Y. Ma, D. Baron, and D. Needell, “Two-part reconstruction with noisy-sudocodes,” *IEEE Trans. Signal Processing*, vol. 62, no. 23, pp. 6323–6334, 2014. [Online]. Available: http://dx.doi.org/10.1109/TSP.2014.2362892

[22] B. Bah and J. Tanner, “Vanishingly sparse matrices and expander graphs, with application to compressed sensing,” *IEEE transactions on information theory*, vol. 59, no. 11, pp. 7491–7508, 2013.

[23] A. Klenke, *Probability theory: a comprehensive course*. Springer Science & Business Media, 2013.

[24] F. Leone, L. Nelson, and R. Nottingham, “The folded normal distribution,” *Technometrics*, vol. 3, no. 4, pp. 543–550, 1961.

[25] D. L. Donoho and J. Tanner, “Precise undersampling theorems,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 913–924, 2010.

[26] J. D. Blanchard, J. Tanner, and K. Wei, “Conjugate gradient iterative hard thresholding: Observed noise stability for compressed sensing,” *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 528–537, Jan 2015.

[27] J. D. Blanchard and J. Tanner, “Performance comparisons of greedy algorithms in compressed sensing,” *Numerical Linear Algebra with Applications*, vol. 22, no. 2, pp. 254–282, 2015.

[28] ——, “GPU accelerated greedy algorithms for compressed sensing,” *Mathematical Programming Computation*, vol. 5, no. 3, pp. 267–304, 2013.

[29] D. L. Donoho and J. Tanner, “Exponential bounds implying construction of compressed sensing matrices, error-correcting codes and neighborly polytopes by random sampling,” *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 2002–2016, 2010.