VERIFIABLE AND COMPUTABLE PERFORMANCE ANALYSIS OF SPARSITY RECOVERY[*]

GONGGUO TANG[†] AND ARYE NEHORAI[‡]

Abstract. In this paper, we develop verifiable and computable performance analysis of sparsity recovery. We define a family of goodness measures for arbitrary sensing matrices as a set of optimization problems, and design algorithms with a theoretical global convergence guarantee to compute these goodness measures. The proposed algorithms solve a series of second-order cone programs, or linear programs. As a by-product, we implement an efficient algorithm to verify a sufficient condition for exact sparsity recovery in the noise-free case. We derive performance bounds on the recovery errors in terms of these goodness measures. We also analytically demonstrate that the developed goodness measures are non-degenerate for a large class of random sensing matrices, as long as the number of measurements is relatively large. Numerical experiments show that, compared with the restricted isometry based performance bounds, our error bounds apply to a wider range of problems and are tighter, when the sparsity levels of the signals are relatively low.

Key words. compressive sensing, computable performance analysis, fixed point theory, linear programming, second-order cone programming, sparsity recovery

AMS subject classifications. 47H10, 90C05, 90C25, 90C26, 90C90, 94A12

1. Introduction.

Sparse signal recovery (or compressive sensing) has revolutionized the way we think of signal sampling [11]. It goes far beyond sampling and has also been applied to areas as diverse as medical imaging, remote sensing, radar, sensor arrays, image processing, computer vision, and so on. Mathematically, sparse signal recovery aims to reconstruct a sparse signal, namely a signal with only a few non-zero components, from usually noisy linear measurements:

$$\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{w}, \tag{1.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the sparse signal, $\boldsymbol{y} \in \mathbb{R}^m$ is the measurement vector, $A \in \mathbb{R}^{m \times n}$ is the sensing/measurement matrix, and $\boldsymbol{w} \in \mathbb{R}^m$ is the noise. A theoretically justified way to exploit the sparseness in recovering $\boldsymbol{x}$ is to minimize its ℓ_1 norm under certain constraints [9].

In this paper, we investigate the problem of using the ℓ_∞ norm as a performance criterion for sparse signal recovery via ℓ_1 minimization. Although the ℓ_2 norm has been used as the performance criterion by the majority of published research in sparse signal recovery, the adoption of the ℓ_∞ norm is well justified. Other popular performance criteria, such as the ℓ_1 and ℓ_2 norms of the error vectors, can all be expressed in terms of the ℓ_∞ norm in a tight and non-trivial manner. More importantly, the ℓ_∞ norm of the error vector has a direct connection with the support recovery problem. To see this, assuming we know *a priori* the minimal non-zero absolute value of the components of the sparse signal, then controlling the ℓ_∞ norm within half of that

[†]Preston M. Green Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO 63130-1127, (`gt2@ese.wustl.edu`).

[‡]Preston M. Green Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO 63130-1127, (`nehorai@ese.wustl.edu`).

value would guarantee exact recovery of the support. Support recovery is arguably one of the most important and challenging problems in sparse signal recovery. In practical applications, the support is usually physically more significant than the component values. For example, in radar imaging using sparse signal recovery, the sparsity constraints are usually imposed on the discretized time–frequency domain. The distance and velocity of a target have a direct correspondence to the support of the sparse signal. The magnitude determined by coefficients of reflection is of less physical significance [1, 18, 19]. Refer to [27] for more discussions on sparse support recovery.

Another, perhaps more important, reason to use the ℓ_∞ norm as a performance criterion is the verifiability and computability of the resulting performance bounds. A general strategy to study the performance of sparse signal recovery is to define a measure of the goodness of the sensing matrix, and then derive performance bounds in terms of the goodness measure. The most well-known goodness measure is undoubtedly the restricted isometry constant (RIC) [7]. Upper bounds on the ℓ_2 and ℓ_1 norms of the error vectors for various recovery algorithms have been expressed in terms of the RIC. Unfortunately, it is extremely difficult to verify that the RIC of a specific sensing matrix satisfies the conditions for the bounds to be valid, and even more difficult to directly compute the RIC itself. Actually, the only known sensing matrices with nice RICs are certain types of random matrices [20]. By using the ℓ_∞ norm as a performance criterion, we develop a framework in which a family of goodness measures for the sensing matrices are verifiable and computable. The computability further justifies the connection of the ℓ_∞ norm with the support recovery problem, since for the connection described in the previous paragraph to be practically useful, we must be able to compute the error bounds on the ℓ_∞ norm.

The verifiability and computability open doors for wide applications. In many practical applications of sparse signal recovery, *e.g.*, radar imaging [24], sensor arrays [22], DNA microarrays [25], and MRI [21], it is beneficial to know the performance of the sensing system before its implementation and the taking of measurements. In addition, in these application areas, we usually have the freedom to optimally design the sensing matrix. For example, in MRI the sensing matrix is determined by the sampling trajectory in the Fourier domain; in radar systems the optimal sensing matrix design is connected with optimal waveform design, a central topic of radar research. To optimally design the sensing matrix, we need to

1. analyze how the performance of recovering $\boldsymbol{x}$ from $\boldsymbol{y}$ is affected by A, and define a function $\omega(A)$ to accurately quantify the goodness of A in the context of sparse signal reconstruction;
2. develop algorithms to efficiently verify that $\omega(A)$ satisfies the conditions for the bounds to hold, as well as to efficiently compute $\omega(A)$ for arbitrarily given A;
3. design mechanisms to select within a matrix class the sensing matrix that is optimal in the sense of best $\omega(A)$.

In this paper, we successfully address the first two points in the ℓ_∞ performance analysis framework.

We now preview our contributions. First of all, we propose using the ℓ_∞ norm as a performance criterion for sparse signal recovery and establish its connections with other performance criteria. We define a family of goodness measures of the sensing matrix, and use them to derive performance bounds on the ℓ_∞ norm of the recovery error vector. Performance bounds using other norms are expressed using the ℓ_∞

norm. Numerical simulations show that these bounds are tighter than the RIC based bounds when the sparsity levels of the signals are relatively small. Secondly and most importantly, using fixed point theory, we develop algorithms to efficiently compute the goodness measures for given sensing matrices by solving a series of second-order cone programs or linear programs, depending on the specific goodness measure being computed. We analytically demonstrate the algorithms' convergence to the global optima from any initial point. As a by-product, we obtain a fast algorithm to verify the sufficient condition guaranteeing exact sparse recovery via ℓ_1 minimization. Finally, we show that the goodness measures are non-degenerate for subgaussian and isotropic random sensing matrices as long as the number of measurements is relatively large, a result parallel to that of the RIC for random matrices.

Several attempts have been made to address the verifiability and computability of performance analysis for sparse signal recovery, mainly based on the RIC [7,9] and the Null Space Property (NSP) [13]. Due to the difficulty of explicitly computing the RIC and verifying the NSP, researchers use relaxation techniques to approximate these quantities. Examples include semi-definite programming relaxation [14,15] and linear programming relaxation [20]. To the best of the authors' knowledge, the algorithms of [14] and [20] represent state-of-the-art techniques in verifying the sufficient condition of unique ℓ_1 recovery. In this paper, we directly address the computability of the performance bounds. More explicitly, we define the goodness measures of the sensing matrices as optimization problems and design efficient algorithms with theoretical convergence guarantees to solve the optimization problems. An algorithm to verify a sufficient condition for exact ℓ_1 recovery is obtained only as a by-product. Our implementation of the algorithm performs orders of magnitude faster than the state-of-the-art techniques in [14] and [20], consumes much less memory, and produces comparable results.

The paper is organized as follows. In Section 2, we introduce notations, and we present the measurement model, three convex relaxation algorithms, and the sufficient and necessary condition for exact ℓ_1 recovery. In section 3, we derive performance bounds on the ℓ_∞ norms of the recovery errors for several convex relaxation algorithms. In Section 4, we design algorithms to verify a sufficient condition for exact ℓ_1 recovery in the noise-free case, and to compute the goodness measures of arbitrarily given sensing matrices. Section 5 is devoted to the probabilistic analysis of our ℓ_∞ performance measures. We evaluate the algorithms' performance in Section 6. Section 7 summarizes our conclusions.

2. Notations, Measurement Model, and Recovery Algorithms. In this section, we introduce notations and the measurement model, and review recovery algorithms based on ℓ_1 minimization.

For any vector $\boldsymbol{x} \in \mathbb{R}^n$, the norm $\|\boldsymbol{x}\|_{k,1}$ is the summation of the absolute values of the k (absolutely) largest components of $\boldsymbol{x}$. In particular, the ℓ_∞ norm $\|\boldsymbol{x}\|_\infty = \|\boldsymbol{x}\|_{1,1}$ and the ℓ_1 norm $\|\boldsymbol{x}\|_1 = \|\boldsymbol{x}\|_{n,1}$. The classical inner product in $\mathbb{R}^n$ is denoted by $\langle \cdot, \cdot \rangle$, and the ℓ_2 (or Euclidean) norm is $\|\boldsymbol{x}\|_2 = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$. We use $\| \cdot \|_\diamond$ to denote a general norm.

The support of $\boldsymbol{x}$, $\text{supp}(\boldsymbol{x})$, is the index set of the non-zero components of $\boldsymbol{x}$. The size of the support, usually denoted by the ℓ_0 "norm" $\|\boldsymbol{x}\|_0$, is the sparsity level of $\boldsymbol{x}$. Signals of sparsity level at most k are called $k-$sparse signals. If $S \subset \{1, \cdots, n\}$ is an index set, then $|S|$ is the cardinality of S, and $\boldsymbol{x}_S \in \mathbb{R}^{|S|}$ is the vector formed by the components of $\boldsymbol{x}$ with indices in S.

We use $\mathbf{e}_i$, $\mathbf{0}$, $\boldsymbol{O}$, and $\mathbf{1}$ to denote respectively the ith canonical basis vector, the

zero column vector, the zero matrix, and the column vector with all ones.

Suppose $\boldsymbol{x}$ is a $k-$sparse signal. In this paper, we observe $\boldsymbol{x}$ through the following linear model:

$$\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{w}, \tag{2.1}$$

where $A \in \mathbb{R}^{m \times n}$ is the measurement/sensing matrix, $\boldsymbol{y}$ is the measurement vector, and $\boldsymbol{w}$ is noise.

Many algorithms have been proposed to recover $\boldsymbol{x}$ from $\boldsymbol{y}$ by exploiting the sparseness of $\boldsymbol{x}$. We focus on three algorithms based on ℓ_1 minimization: the Basis Pursuit [12], the Dantzig selector [10], and the LASSO estimator [29].

$$\text{Basis Pursuit:} \min_{\boldsymbol{z} \in \mathbb{R}^n} \|\boldsymbol{z}\|_1 \ \text{ s.t. } \|\boldsymbol{y} - A\boldsymbol{z}\|_\diamond \leq \varepsilon \tag{2.2}$$

$$\text{Dantzig:} \min_{\boldsymbol{z} \in \mathbb{R}^n} \|\boldsymbol{z}\|_1 \ \text{ s.t. } \ \|A^T(\boldsymbol{y} - A\boldsymbol{z})\|_\infty \leq \mu \tag{2.3}$$

$$\text{LASSO:} \min_{\boldsymbol{z} \in \mathbb{R}^n} \frac{1}{2}\|\boldsymbol{y} - A\boldsymbol{z}\|_2^2 + \mu\|\boldsymbol{z}\|_1. \tag{2.4}$$

Here μ is a tuning parameter, and ε is a measure of the noise level. All three optimization problems have efficient implementations using convex programming or even linear programming.

In the noise-free case where $\boldsymbol{w} = 0$, roughly speaking all the three algorithms reduce to

$$\min_{\boldsymbol{z} \in \mathbb{R}^n} \|\boldsymbol{z}\|_1 \ \text{ s.t. } A\boldsymbol{z} = A\boldsymbol{x}, \tag{2.5}$$

which is the ℓ_1 relaxation of the NP hard ℓ_0 minimization problem:

$$\min_{\boldsymbol{z} \in \mathbb{R}^n} \|\boldsymbol{z}\|_0 \ \text{ s.t. } A\boldsymbol{z} = A\boldsymbol{x}. \tag{2.6}$$

A minimal requirement on ℓ_1 minimization algorithms is the *uniqueness and exactness* of the solution $\hat{\boldsymbol{x}} \overset{\text{def}}{=} \mathrm{argmin}_{\boldsymbol{z}:A\boldsymbol{z}=A\boldsymbol{x}}\|\boldsymbol{x}\|_1$, *i.e.*, $\hat{\boldsymbol{x}} = \boldsymbol{x}$. When the true signal $\boldsymbol{x}$ is $k-$sparse, the sufficient and necessary condition for exact ℓ_1 recovery is [16, 17, 30]

$$\sum_{i \in S} |z_i| < \sum_{i \notin S} |z_i|, \forall \boldsymbol{z} \in \mathrm{Ker}(A), |S| \leq k, \tag{2.7}$$

where $\mathrm{Ker}(A) \overset{\text{def}}{=} \{\boldsymbol{z} : A\boldsymbol{z} = 0\}$ is the kernel of A, and $S \subset \{1, \dots, n\}$ is an index set. Expressed in terms of $\|\cdot\|_{k,1}$, the necessary and sufficient condition becomes

$$\|\boldsymbol{z}\|_{k,1} < \frac{1}{2}\|\boldsymbol{z}\|_1, \forall \boldsymbol{z} \in \mathrm{Ker}(A). \tag{2.8}$$

The approaches in [20] and [14] for verifying the sufficient condition (2.8) are based on relaxing the following optimization problem in various ways:

$$\alpha_k = \max_{\boldsymbol{z}} \|\boldsymbol{z}\|_{k,1} \ \text{ s.t. } A\boldsymbol{z} = 0, \|\boldsymbol{z}\|_1 \leq 1. \tag{2.9}$$

Clearly, $\alpha_k < 1/2$ is necessary and sufficient for exact ℓ_1 recovery for $k-$sparse signals. Unfortunately, the direct computation of (2.9) for general k is extremely difficult: it

is the maximization of a norm (convex function) over a polyhedron (convex set) [4]. In [20], in a very rough sense α_1 was computed by solving n linear programs:

$$\min_{\boldsymbol{y}_i \in \mathbb{R}^m} \|\mathbf{e}_i - A^T \boldsymbol{y}_i\|_\infty, i = 1, \cdots, n, \tag{2.10}$$

where $\mathbf{e}_i$ is the ith canonical basis in $\mathbb{R}^n$. This, together with the observation that $\alpha_k < k\alpha_1$, yields an efficient algorithm to verify (2.8). However, in [26], we found that the primal-dual method of directly solving (2.8) as the following n linear programs

$$\max \boldsymbol{z}_i \text{ s.t. } A\boldsymbol{z} = 0, \|\boldsymbol{z}\|_1 \leq 1 \tag{2.11}$$

gives rise to an algorithm orders of magnitude faster. In the next section, we will see how the computation of α_1 arises naturally in the context of ℓ_∞ performance evaluation.

3. Performance Bounds on the ℓ_∞ Norms of the Recovery Errors. In this section, we derive performance bounds on the ℓ_∞ norms of the error vectors. We first establish a theorem characterizing the error vectors for the ℓ_1 recovery algorithms, whose proof is given in Appendix 8.1

PROPOSITION 3.1. *Suppose $\boldsymbol{x}$ in (2.1) is $k-$sparse and the noise $\boldsymbol{w}$ satisfies $\|\boldsymbol{w}\|_\diamond \leq \varepsilon$, $\|A^T \boldsymbol{w}\|_\infty \leq \mu$, and $\|A^T \boldsymbol{w}\|_\infty \leq \kappa\mu, \kappa \in (0,1)$, for the Basis Pursuit, the Dantzig selector, and the LASSO estimator, respectively. Define $\boldsymbol{h} = \hat{\boldsymbol{x}} - \boldsymbol{x}$ as the error vector for any of the three ℓ_1 recovery algorithms (2.2), (2.3), and (2.4). Then we have*

$$c\|\boldsymbol{h}\|_{k,1} \geq \|\boldsymbol{h}\|_1, \tag{3.1}$$

where $c = 2$ for the Basis Pursuit and the Dantzig selector, and $c = 2/(1-\kappa)$ for the LASSO estimator.

An immediate corollary of Proposition 3.1 is to bound the ℓ_1 and ℓ_2 norms of the error vector using the ℓ_∞ norm:

COROLLARY 3.2. *Under the assumptions of Proposition 3.1, we have*

$$\|\boldsymbol{h}\|_1 \leq ck\|\boldsymbol{h}\|_\infty, \tag{3.2}$$

$$\|\boldsymbol{h}\|_2 \leq \sqrt{ck}\|\boldsymbol{h}\|_\infty. \tag{3.3}$$

Furthermore, if $S = \text{supp}(\boldsymbol{x})$ and $\beta = \min_{i \in S} |\boldsymbol{x}_i|$, then $\|\boldsymbol{h}\|_\infty < \beta/2$ implies

$$\text{supp}(\max(|\hat{\boldsymbol{x}}| - \beta/2, 0) = \text{supp}(\boldsymbol{x}), \tag{3.4}$$

i.e., a thresholding operator recovers the signal support.

For ease of presentation, we have the following definition:

DEFINITION 3.3. *For any real number $s \in [1, n]$ and matrix $A \in \mathbb{R}^{m \times n}$, define*

$$\omega_\diamond(Q, s) = \min_{\boldsymbol{z}:\|\boldsymbol{z}\|_1/\|\boldsymbol{z}\|_\infty \leq s} \frac{\|Q\boldsymbol{z}\|_\diamond}{\|\boldsymbol{z}\|_\infty}, \tag{3.5}$$

where Q is either A or $A^T A$.

Now we present the error bounds on the ℓ_∞ norm of the error vectors for the Basis Pursuit, the Dantzig selector, and the LASSO estimator.

THEOREM 3.4. *Under the assumption of Proposition 3.1, we have*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_\infty \leq \frac{2\varepsilon}{\omega_\diamond(A, 2k)} \tag{3.6}$$

for the Basis Pursuit,

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_\infty \leq \frac{2\mu}{\omega_\infty(A^T A, 2k)} \tag{3.7}$$

for the Dantzig selector, and

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_\infty \leq \frac{(1 + \kappa)\mu}{\omega_\infty(A^T A, 2k/(1 - \kappa))} \tag{3.8}$$

for the LASSO estimator.

Proof. Observe that for the Basis Pursuit

$$\begin{aligned}
\|A(\hat{\boldsymbol{x}} - \boldsymbol{x})\|_2 &\leq \|\boldsymbol{y} - A\hat{\boldsymbol{x}}\|_2 + \|\boldsymbol{y} - A\boldsymbol{x}\|_2 \\
&\leq \varepsilon + \|A\boldsymbol{w}\|_2 \\
&\leq 2\varepsilon,
\end{aligned} \tag{3.9}$$

and similarly,

$$\|A^T A(\hat{\boldsymbol{x}} - \boldsymbol{x})\|_\infty \leq 2\mu \tag{3.10}$$

for the Dantzig selector, and

$$\|A^T A(\hat{\boldsymbol{x}} - \boldsymbol{x})\|_\infty \leq (1 + \kappa)\mu \tag{3.11}$$

for the LASSO estimator. The conclusions of Theorem 3.4 follow from equations (3.2), (3.3), and Definition 3.3. □

One of the primary contributions of this work is the design of algorithms that compute $\omega_\diamond(A, s)$ and $\omega_\infty(A^T A, s)$ efficiently. The algorithms provide a way to numerically assess the performance of the Basis Pursuit, the Dantzig selector, and the LASSO estimator according to the bounds given in Theorem 3.4. According to Corollary 3.2, the correct recovery of signal support is also guaranteed by reducing the ℓ_∞ norm to some threshold. In Section 5, we also demonstrate that the bounds in Theorem 3.4 are non-trivial for a large class of random sensing matrices, as long as m is relatively large. Numerical simulations in Section 6 show that in many cases the error bounds on the ℓ_2 norms based on Corollary 3.2 and Theorem 3.4 are tighter than the RIC based bounds. We expect the bounds on the ℓ_∞ norms in Theorem 3.4 are even tighter, as we do not need the relaxation in Corollary 3.2.

We note that a prerequisite for these bounds to be valid is the positiveness of the involved $\omega_\diamond(\cdot)$. We call the validation of $\omega_\diamond(\cdot) > 0$ the verification problem. Note that from Theorem 3.4, $\omega_\diamond(\cdot) > 0$ implies the exact recovery of the true signal $\boldsymbol{x}$ in the noise-free case. Therefore, verifying $\omega_\diamond(\cdot) > 0$ is equivalent to verifying a sufficient condition for exact ℓ_1 recovery.

4. Verification and Computation of $\omega_\diamond$. In this section, we present algorithms for verification and computation of $\omega_\diamond(\cdot)$. We will present a very general algorithm and make it specific only when necessary. For this purpose, we use Q to denote either A or $A^T A$, and use $\| \cdot \|_\diamond$ to denote a general norm.

4.1. Verification of $\omega_\diamond > 0$. Verifying $\omega_\diamond(Q, s) > 0$ amounts to making sure $\|\boldsymbol{z}\|_1/\|\boldsymbol{z}\|_\infty \leq s$ for all $\boldsymbol{z}$ such that $Q\boldsymbol{z} = 0$. Equivalently, we can compute

$$s_* = \min_{\boldsymbol{z}} \frac{\|\boldsymbol{z}\|_1}{\|\boldsymbol{z}\|_\infty} \text{ s.t. } Q\boldsymbol{z} = 0. \tag{4.1}$$

Then, when $s < s_*$, we have $\omega_\diamond(Q, s) > 0$. We rewrite the optimization (4.1) as

$$\frac{1}{s_*} = \max_{\boldsymbol{z}} \|\boldsymbol{z}\|_\infty \text{ s.t. } Q\boldsymbol{z} = 0, \|\boldsymbol{z}\|_1 \leq 1, \tag{4.2}$$

which is solved using the following n linear programs:

$$\max_{\boldsymbol{z}} \boldsymbol{z}_i \text{ s.t. } Q\boldsymbol{z} = 0, \|\boldsymbol{z}\|_1 \leq 1. \tag{4.3}$$

The dual problem for (4.3) is

$$\min_{\boldsymbol{\lambda}} \|\mathbf{e}_i - Q^T \boldsymbol{\lambda}\|_\infty, \tag{4.4}$$

where $\mathbf{e}_i$ is the ith canonical basis vector.

We solve (4.3) using the primal-dual algorithm expounded in Chapter 11 of [5], which gives an implementation much more efficient than the one for solving its dual (4.4) in [20]. This method is also used to implement the ℓ_1 MAGIC for sparse signal recovery [6]. Due to the equivalence of $A^T A \boldsymbol{z} = 0$ and $A\boldsymbol{z} = 0$, we always solve (4.2) for $Q = A$ and avoid $Q = A^T A$. The former apparently involves solving linear programs of smaller size. In practice, we usually replace A with the matrix with orthogonal rows obtained from the economy-size QR decomposition of A^T.

As a dual of (4.4), (4.3) (and hence (4.2) and (4.1)) shares the same limitation as (4.4), namely, it verifies $\omega_\diamond > 0$ only for s up to $2\sqrt{2m}$. We now reformulate Proposition 4 of [20] in our framework:

PROPOSITION 4.1. [20, Proposition 4] *For any $m \times n$ matrix A with $n \geq 32m$, one has*

$$s_* = \min \left\{ \frac{\|\boldsymbol{z}\|_1}{\|\boldsymbol{z}\|_\infty} : Q\boldsymbol{z} = 0 \right\} < 2\sqrt{2m}. \tag{4.5}$$

4.2. Computation of $\omega_\diamond$. Now we turn to one of the primary contributions of this work, the computation of $\omega_\diamond$. The optimization problem is as follows:

$$\omega_\diamond(Q, s) = \min_{\boldsymbol{z}} \frac{\|Q\boldsymbol{z}\|_\diamond}{\|\boldsymbol{z}\|_\infty} \text{ s.t. } \frac{\|\boldsymbol{z}\|_1}{\|\boldsymbol{z}\|_\infty} \leq s, \tag{4.6}$$

or equivalently,

$$\frac{1}{\omega_\diamond(Q, s)} = \max_{\boldsymbol{z}} \|\boldsymbol{z}\|_\infty \text{ s.t. } \|Q\boldsymbol{z}\|_\diamond \leq 1, \frac{\|\boldsymbol{z}\|_1}{\|\boldsymbol{z}\|_\infty} \leq s. \tag{4.7}$$

We will show that $1/\omega_\diamond(Q, s)$ is the unique fixed point of certain scalar function. To this end, we define functions $f_{s,i}(\eta), i = 1, \ldots, n$ and $f_s(\eta)$ over $[0, \infty)$ parameterized by $s \in (1, s_*)$:

$$f_{s,i}(\eta) \stackrel{\text{def}}{=} \max_{\boldsymbol{z}} \{\boldsymbol{z}_i : \|Q\boldsymbol{z}\|_\diamond \leq 1, \|\boldsymbol{z}\|_1 \leq s\eta\}$$
$$= \max_{\boldsymbol{z}} \{|\boldsymbol{z}_i| : \|Q\boldsymbol{z}\|_\diamond \leq 1, \|\boldsymbol{z}\|_1 \leq s\eta\}, \tag{4.8}$$

since the domain for the maximization is symmetric to the origin, and

$$f_s(\eta) \stackrel{\text{def}}{=} \max_{\boldsymbol{z}} \{\|\boldsymbol{z}\|_\infty : \|Q\boldsymbol{z}\|_\diamond \leq 1, \|\boldsymbol{z}\|_1 \leq s\eta\}$$

$$= \max_{\substack{\boldsymbol{z}:\|Q\boldsymbol{z}\|_\diamond \leq 1 \\ \|\boldsymbol{z}\|_1 \leq s\eta}} \max_i |\boldsymbol{z}_i|$$

$$= \max_i \max_{\substack{\boldsymbol{z}:\|Q\boldsymbol{z}\|_\diamond \leq 1 \\ \|\boldsymbol{z}\|_1 \leq s\eta}} |\boldsymbol{z}_i|$$

$$= \max_i f_{s,i}(\eta), \tag{4.9}$$

where for the last but one equality we have exchanged the two maximizations. For $\eta > 0$, it is easy to show that strong duality holds for the optimization problem defining $f_{s,i}(\eta)$. As a consequence, we have the dual form of $f_{s,i}(\eta)$:

$$f_{s,i}(\eta) = \min_{\boldsymbol{\lambda}} s\eta\|\mathbf{e}_i - Q^T\boldsymbol{\lambda}\|_\infty + \|\boldsymbol{\lambda}\|_\diamond^*, \tag{4.10}$$

where $\|\cdot\|_\diamond^*$ is the dual norm of $\|\cdot\|_\diamond$.

In the definition of $f_s(\eta)$, we basically replaced the $\|\boldsymbol{z}\|_\infty$ in the denominator of the fractional constraint in (4.7) with η. The following theorem states that the unique positive fixed point of $f_s(\eta)$ is exactly $1/\omega_\diamond(Q, s)$. See Appendix 8.2 for the proof.

THEOREM 4.2. *The functions $f_{s,i}(\eta)$ and $f_s(\eta)$ have the following properties:*
1. *$f_{s,i}(\eta)$ and $f_s(\eta)$ are continuous in η;*
2. *$f_{s,i}(\eta)$ and $f_s(\eta)$ are strictly increasing in η;*
3. *$f_{s,i}(\eta)$ is concave for every i;*
4. *$f_s(0) = 0$, $f_s(\eta) \geq s\eta > \eta$ for sufficiently small $\eta > 0$, and there exists $\rho < 1$ such that $f_s(\eta) < \rho\eta$ for sufficiently large η; the same holds for $f_{s,i}(\eta)$;*
5. *$f_{s,i}$ and $f_s(\eta)$ have unique positive fixed points $\eta_i^* = f_{s,i}(\eta_i^*)$ and $\eta^* = f_s(\eta^*)$, respectively; and $\eta^* = \max_i \eta_i^*$;*
6. *The unique positive fixed point of $f_s(\eta)$, η^*, is equal to $1/\omega_\diamond(Q, s)$;*
7. *For $\eta \in (0, \eta^*)$, we have $f_s(\eta) > \eta$; and for $\eta \in (\eta^*, \infty)$, we have $f_s(\eta) < \eta$; the same statement holds also for $f_{s,i}(\eta)$.*
8. *For any $\epsilon > 0$, there exists $\rho_1(\epsilon) > 1$ such that $f_s(\eta) > \rho_1(\epsilon)\eta$ as long as $0 < \eta \leq (1-\epsilon)\eta^*$; and there exists $\rho_2(\epsilon) < 1$ such that $f_s(\eta) < \rho_2(\epsilon)\eta$ as long as $\eta > (1+\epsilon)\eta^*$.*

Theorem 4.2 implies three ways to compute the fixed point of $\eta^* = 1/\omega_\diamond(Q, s)$ for $f_s(\eta)$.
1. **Naive Fixed Point Iteration:** Property 8) of Theorem 4.2 suggests that the fixed point iteration

$$\eta_{t+1} = f_s(\eta_t), t = 0, 1, \ldots \tag{4.11}$$

starting from any initial point $\eta_0 > 0$ converges to η^*, no matter $\eta_0 < \eta^*$ or $\eta_0 > \eta^*$. The algorithm can be made more efficient in the case $\eta_0 < \eta^*$. More specifically, since $f_s(\eta) = \max_i f_{s,i}(\eta)$, at each fixed point iteration, we set η_{t+1} to be the first $f_{s,i}(\eta_t)$ that is greater than $\eta_t + \epsilon$ with ϵ some tolerance parameter. If for all i, $f_{s,i}(\eta_t) < \eta_t + \epsilon$, then $f_s(\eta_t) = \max_i f_{s,i}(\eta_t) < \eta_t + \epsilon$, which indicates the optimal function value can not be improved greatly and the algorithm should terminate. In most cases, to get η_{t+1}, we need to solve only one optimization problem $\max_{\boldsymbol{z}} \boldsymbol{z}_i : \|Q\boldsymbol{z}\|_\diamond \leq 1, \|\boldsymbol{z}\|_1 \leq s\eta_t$ instead of n. This is in contrast to the case where $\eta_0 > \eta^*$, because in the later case we
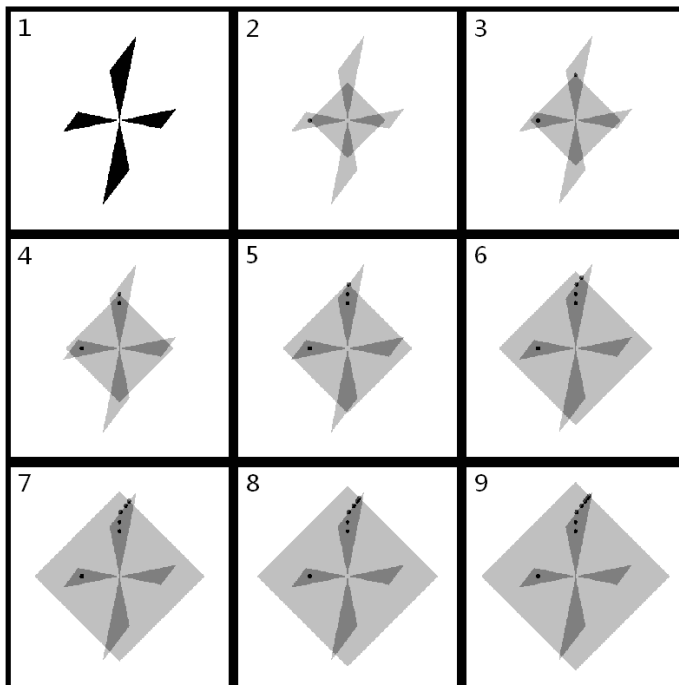
Fig. 4.1: Illustration of the naive fixed point iteration (4.11) when $\diamond = \infty$.

must compute all $f_{s,i}(\eta_t)$ to update $\eta_{t+1} = \max_i f_{s,i}(\eta_t)$. An update based on a single $f_{s,i}(\eta_t)$ might generate a value smaller than η^*.

In Figure 4.1, we illustrate the behavior of the naive fixed point iteration algorithm (4.11). These figures are generated by Matlab for a two dimensional problem. We index the sub-figures from left to right and from top to bottom. The first (upper left) sub-figure shows the star-shaped region $\mathcal{S} = \{\boldsymbol{z} : \|Q\boldsymbol{z}\|_\infty \leq 1, \|\boldsymbol{z}\|_1 / \|\boldsymbol{z}\|_\infty \leq s\}$. Starting from an initial $\eta_0 < \eta^*$, the algorithm solves

$$\max_{\boldsymbol{z}} \|\boldsymbol{z}\|_\infty \text{ s.t. } \|Q\boldsymbol{z}\|_\diamond \leq 1, \|\boldsymbol{z}\|_1 \leq s\eta_0 \tag{4.12}$$

in sub-figure 2. The solution is denoted by the black dot. Although the true domain for the optimization in (4.12) is the intersection of the distorted ℓ_∞ ball $\{\boldsymbol{z} : \|Q\boldsymbol{z}\|_\infty \leq 1\}$ and the ℓ_1 ball $\{\boldsymbol{z} : \|\boldsymbol{z}\|_1 \leq s\eta_0\}$, the intersection of the ℓ_1 ball (light gray diamond) and the star-shaped region $\mathcal{S}$ forms the effective domain, which is the dark grey region in the sub-figures. To see this, we note the optimal value of the optimization (4.12) $\eta_1 = \|\boldsymbol{x}_1^*\|_\infty = f_s(\eta_0) > \eta_0$ according to 7) of Theorem 4.2, implying that, for the optimal solution $\boldsymbol{x}_1^*$, we have $\|\boldsymbol{x}_1^*\|_1 / \|\boldsymbol{x}_1^*\|_\infty \leq \|\boldsymbol{x}_1^*\|_1 / \eta_0 \leq s$. Therefore, the optimal solution $\boldsymbol{x}_1^*$ can always be found in the dark grey region. In the following sub-figures, at each iteration, we expand the ℓ_1 ball until we get to the tip point of the star-shaped region $\mathcal{S}$, which is the global optimum.

Despite of its simplicity, the naive fixed point iteration has two major disadvantages. Firstly, the stopping criterion based on successive improvement

is not accurate as it does not reflect the gap between η_t and η^*. This disadvantage can be remedied by starting from both below and above η^*. The distance between corresponding terms in the two generated sequences is an indication of the gap to the fixed point η^*. However, the resulting algorithm is generally slow, especially when updating η_{t+1} from above η^*. Secondly, the iteration process is slow when close to the fixed point η^*. This is because $\rho_1(\epsilon)$ and $\rho_2(\epsilon)$ in 8) of Theorem 4.2 are close to 1 for small $\epsilon > 0$.

2. **Bisection:** The bisection approach is motivated by property 7) of Theorem 4.2. Starting from an initial interval (η_L, η_U) that contains η^*, we compute $f_s(\eta_M)$ with $\eta_M = (\eta_L + \eta_U)/2$. As a consequence of property 7), $f_s(\eta_M) > \eta_M$ implies $f_s(\eta_M) < \eta^*$, and we set $\eta_L = f_s(\eta_M)$; $f_s(\eta_M) < \eta_M$ implies $f_s(\eta_M) > \eta^*$, and we set $\eta_U = f_s(\eta_M)$. The bisection process can also be accelerated by setting $\eta_L = f_{s,i}(\eta_M)$ for the first $f_{s,i}(\eta_M)$ greater than η_M. The convergence of the bisection approach is much faster than the naive fixed point iteration because each iteration reduces the interval length at least by half. In addition, half the length of the interval is an upper bound on the gap between η_M and η^*, resulting an accurate stopping criterion. However, if the initial η_U is too larger than η^*, the majority of $f_s(\eta_M)$ would turn out to be less than η^*. The verification of $f_s(\eta_M) < \eta_M$ needs solving n linear programs or second-order cone programs, greatly degrading the algorithm's performance.

3. **Fixed Point Iteration + Bisection:** The third approach combines the advantages of the bisection method and the fixed point iteration method, at the level of $f_{s,i}(\eta)$. This method relies on the representation $f_s(\eta) = \max_i f_{s,i}(\eta)$ and $\eta^* = \max_i \eta_i^*$.
Starting from an initial interval (η_{L0}, η_U) and the index set $\mathcal{I}_0 = \{1, \ldots, n\}$, we pick any $i_0 \in \mathcal{I}_0$ and use the (accelerated) bisection method with starting interval (η_{L0}, η_U) to find the positive fixed point $\eta_{i_0}^*$ of $f_{s,i_0}(\eta)$. For any $i \in \mathcal{I}_0/i_0$, $f_{s,i}(\eta_{i_0}^*) \leq \eta_{i_0}^*$ implies that the fixed point η_i^* of $f_{s,i}(\eta)$ is less than or equal to $\eta_{i_0}^*$ according to the continuity of $f_{s,i}(\eta)$ and the uniqueness of its positive fixed point. As a consequence, we remove this i from the index set $\mathcal{I}_0$. We denote $\mathcal{I}_1$ as the index set after all such is removed, *i.e.,* $\mathcal{I}_1 = \mathcal{I}_0/\{i : f_{s,i}(\eta_{i_0}^*) \leq \eta_{i_0}^*\}$. We then set $\eta_{L1} = \eta_{i_0}^*$ as $\eta^* \geq \eta_{i_0}^*$. Next we test the $i_1 \in \mathcal{I}_1$ with the *largest* $f_{s,i}(\eta_{i_0}^*)$ and construct $\mathcal{I}_2$ and η_{L2} in a similar manner. We repeat the process until the index set $\mathcal{I}_t$ is empty. The η_i^* found at the last step is the maximal η_i^*, which is equal to η^*.

Note that in equations (4.6), (4.7), and (4.9), if we replace the ℓ_∞ norm with any other norm (with some other minor modifications), especially $\|\cdot\|_{s,1}$ or $\|\cdot\|_2$, then a naive fixed point iteration algorithm still exists. In addition, as we did in Corollary 3.2, we can express other norms on the error vector in terms of $\|\cdot\|_{s,1}$ and $\|\cdot\|_2$. We expect the norm $\|\cdot\|_{s,1}$ would yield the tightest performance bounds. Unfortunately, the major problem is that in these cases, the function $f_s(\eta)$ do not admit an obvious polynomial time algorithm to compute. It is very likely the corresponding norm maximization defining $f_s(\eta)$ for $\|\cdot\|_{s,1}$ and $\|\cdot\|_2$ are NP hard [4].

5. Probabilistic Behavior of $\omega_\diamond(Q, s)$. In [26], we defined the ℓ_1-constrained minimal singular value (ℓ_1-CMSV) as a goodness measure of the sensing matrix and established performance bounds using ℓ_1−CMSV. For comparison, we include the definition below:

DEFINITION 1. *For any $s \in [1, n]$ and matrix $A \in \mathbb{R}^{m \times n}$, define the ℓ_1-constrained minimal singular value (abbreviated as ℓ_1-CMSV) of A by*

$$\rho_s(A) = \min_{\boldsymbol{z}:\ \|\boldsymbol{z}\|_1^2/\|\boldsymbol{z}\|_2^2 \leq s} \frac{\|A\boldsymbol{z}\|_2}{\|\boldsymbol{z}\|_2}. \tag{5.1}$$

Despite the seeming resemblance of the definitions between $\omega_\diamond(Q, s)$, especially $\omega_2(A, s)$, and $\rho_s(A)$, the difference in the ℓ_∞ norm and the ℓ_2 norm has important implications. As shown in Theorem 4.2, the ℓ_∞ norm enables the design of optimization procedures with nice convergence properties to efficiently compute $\omega_\diamond(Q, s)$. On the other hand, the ℓ_1-CMSV yields tight performance bounds at least for a large class of random sensing matrices, as we will see in Theorem 5.2.

However, there are some interesting connections among these quantities, as shown in the following proposition. These connections allow us the analyze the probabilistic behavior of $\omega_\diamond(Q, s)$ using the results for $\rho_s(A)$ established in [26].

PROPOSITION 5.1.

$$\sqrt{s}\sqrt{\omega_\infty(A^T A, s)} \geq \omega_2(A, s) \geq \rho_{s^2}(A). \tag{5.2}$$

Proof. For any $\boldsymbol{z}$ such that $\|\boldsymbol{z}\|_\infty = 1$ and $\|\boldsymbol{z}\|_1 \leq s$, we have

$$\boldsymbol{z}A^T A\boldsymbol{z} \leq \sum_i |\boldsymbol{z}_i||(A^T A\boldsymbol{z})_i|$$
$$\leq \|\boldsymbol{z}\|_1 \|A^T A\boldsymbol{z}\|_\infty$$
$$\leq s\|A^T A\boldsymbol{z}\|_\infty. \tag{5.3}$$

Taking the minimum over $\{\boldsymbol{z} : \|\boldsymbol{z}\|_\infty = 1, \|\boldsymbol{z}\|_1 \leq s\}$ yields

$$\omega_2^2(A, s) \leq s\omega_\infty(A^T A, s). \tag{5.4}$$

Note that $\|\boldsymbol{z}\|_1/\|\boldsymbol{z}\|_\infty \leq s$ implies $\|\boldsymbol{z}\|_1 \leq s\|\boldsymbol{z}\|_\infty \leq s\|\boldsymbol{z}\|_2$, or equivalently,

$$\{\boldsymbol{z} : \|\boldsymbol{z}\|_1/\|\boldsymbol{z}\|_\infty \leq s\} \subseteq \{\boldsymbol{z} : \|\boldsymbol{z}\|_1/\|\boldsymbol{z}\|_2 \leq s\}. \tag{5.5}$$

As a consequence, we have

$$\begin{aligned}
\omega_2(A, s) &= \min_{\|\boldsymbol{z}\|_1/\|\boldsymbol{z}\|_\infty \leq s} \frac{\|A\boldsymbol{z}\|_2}{\|\boldsymbol{z}\|_2} \frac{\|\boldsymbol{z}\|_2}{\|\boldsymbol{z}\|_\infty} \\
&\geq \min_{\|\boldsymbol{z}\|_1/\|\boldsymbol{z}\|_\infty \leq s} \frac{\|A\boldsymbol{z}\|_2}{\|\boldsymbol{z}\|_2} \\
&\geq \min_{\|\boldsymbol{z}\|_1/\|\boldsymbol{z}\|_2 \leq s} \frac{\|A\boldsymbol{z}\|_2}{\|\boldsymbol{z}\|_2} \\
&= \rho_{s^2}(A), \tag{5.6}
\end{aligned}$$

where the first inequality is due to $\|\boldsymbol{z}\|_2 \geq \|\boldsymbol{z}\|_\infty$, and the second inequality is because the minimization is taken over a larger set. $\square$

As a consequence of the theorem we established in [26] and include below, we derive a condition on the number of measurements to get $\omega_\diamond(Q, s)$ bounded away from zero with high probability for sensing matrices with *i.i.d.* subgaussian and isotropic

rows. Note that a random vector $\boldsymbol{X} \in \mathbb{R}^n$ is called *isotropic and subgaussian* with constant L if $\mathbb{E}|\langle \boldsymbol{X}, \boldsymbol{u} \rangle|^2 = \|\boldsymbol{u}\|_2^2$ and $\mathbb{P}(|\langle \boldsymbol{X}, \boldsymbol{u} \rangle| \geq t) \leq 2\exp(-t^2/(L\|\boldsymbol{u}\|_2))$ hold for any $\boldsymbol{u} \in \mathbb{R}^n$.

THEOREM 5.2. [26] *Let the rows of the scaled sensing matrix $\sqrt{m}A$ be i.i.d. subgaussian and isotropic random vectors with numerical constant L. Then there exist constants c_1 and c_2 such that for any $\epsilon > 0$ and $m \geq 1$ satisfying*

$$m \geq c_1 \frac{L^2 s \log n}{\epsilon^2}, \tag{5.7}$$

we have

$$\mathbb{E}|1 - \rho_s(A)| \leq \epsilon, \tag{5.8}$$

and

$$\mathbb{P}\{1 - \epsilon \leq \rho_s(A) \leq 1 + \epsilon\} \geq 1 - \exp(-c_2\epsilon^2 m/L^4). \tag{5.9}$$

THEOREM 5.3. *Under the assumptions and notations of Theorem 5.2, there exist constants c_1 and c_2 such that for any $\epsilon > 0$ and $m \geq 1$ satisfying*

$$m \geq c_1 \frac{L^2 s^2 \log n}{\epsilon^2}, \tag{5.10}$$

we have

$$\mathbb{E}\,\omega_2(A, s) \geq 1 - \epsilon, \tag{5.11}$$
$$\mathbb{P}\{\omega_2(A, s) \geq 1 - \epsilon\} \geq 1 - \exp(-c_2\epsilon^2 m), \tag{5.12}$$

and

$$\mathbb{E}\,\omega_\infty(A^T A, s) \geq \frac{(1 - \epsilon)^2}{s}, \tag{5.13}$$
$$\mathbb{P}\left\{\omega_\infty(A, s) \geq \frac{(1 - \epsilon)^2}{s}\right\} \geq 1 - \exp(-c_2\epsilon^2 m). \tag{5.14}$$

Sensing matrices with *i.i.d.* subgaussian and isotropic rows include the Gaussian ensemble, and the Bernoulli ensemble, as well as the normalized volume measure on various convex symmetric bodies, for example, the unit balls of ℓ_p^n for $2 \leq p \leq \infty$ [23]. In equations (5.13) and (5.14), the extra s in the lower bound of $\omega_\infty(A^T A, s)$ would contribute an s factor in the bounds of Theorem 3.4. It plays the same role as the extra $\sqrt{k}$ factor in the error bounds for the Dantzig selector and the LASSO estimator in terms of the RIC and the ℓ_1-CMSV [10, 26].

The measurement bound (5.10) implies that the algorithms for verifying $\omega_\diamond > 0$ and for computing $\omega_\diamond$ work for s at least up to the order $\sqrt{m/\log n}$. The order $\sqrt{m/\log n}$ is complementary to the $\sqrt{m}$ upper bound in Proposition 4.1.

Note that Theorem 5.2 implies that the following program:

$$\max_{\boldsymbol{z}} \|\boldsymbol{z}\|_2 \text{ s.t. } A\boldsymbol{z} = 0, \|\boldsymbol{z}\|_1 \leq 1, \tag{5.15}$$

verifies the sufficient condition for exact ℓ_1 recovery for s up to the order $m/\log n$, at least for subgaussian and isotropic random sensing matrices. Unfortunately, this program is NP hard and hence not tractable.

6. Numerical Experiments. In this section, we provide implementation details and numerically assess the performance of the algorithms for solving (4.6) using the naive fixed point iteration. The numerical implementation and performance of (4.1) were previously reported in [26] and hence are omitted here. All the numerical experiments in this section were conducted on a desktop computer with a Pentium D CPU@3.40GHz, 2GB RAM, and Windows XP operating system, and the computations were running single-core.

Recall that the optimization defining $f_{s,i}(\eta)$ is

$$\min z_i \text{ s.t. } \|Qz\|_\diamond \leq 1, \|z\|_1 \leq s\eta. \tag{6.1}$$

Depending on whether $\diamond = 1, \infty$, or 2, (6.1) is solved using either linear programs or second-order cone programs. For example, when $\diamond = \infty$, we have the following corresponding linear programs:

$$\min \begin{bmatrix} \mathbf{e}_i^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} z \\ u \end{bmatrix}$$

$$\text{s.t. } \begin{bmatrix} Q & O \\ -Q & O \\ I & -I \\ -I & -I \\ \mathbf{0}^T & \mathbf{1}^T \end{bmatrix} \begin{bmatrix} z \\ u \end{bmatrix} \leq \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ s\eta \end{bmatrix}, i = 1, \ldots, n, \tag{6.2}$$

These linear programs are implemented using the primal-dual algorithm outlined in Chapter 11 of [5]. The algorithm finds the optimal solution together with optimal dual vectors by solving the Karush-Kuhn-Tucker condition using linearization. The major computation is spent in solving linear systems of equations with positive definite coefficient matrices. When $\diamond = 2$, we rewrite (6.1) as the following second-order cone programs

$$\min \begin{bmatrix} \mathbf{e}_i^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} z \\ u \end{bmatrix}$$

$$\text{s.t. } \frac{1}{2} \left(\left\| \begin{bmatrix} Q & O \end{bmatrix} \begin{bmatrix} z \\ u \end{bmatrix} \right\|_2^2 - 1 \right) \leq 0$$

$$\begin{bmatrix} I & -I \\ -I & -I \\ \mathbf{0}^T & \mathbf{1}^T \end{bmatrix} \begin{bmatrix} z \\ u \end{bmatrix} \leq \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ s\eta \end{bmatrix}. \tag{6.3}$$

We use the log-barrier algorithm described in Chapter 11 of [5] to solve (6.3). Interested readers are encouraged to refer to [6] for a concise exposition of the general primal-dual and log-barrier algorithms and implementation details for similar linear programs and second-order cone programs.

We test the algorithms on Bernoulli, Gaussian, and Hadamard matrices of different sizes. The entries of Bernoulli and Gaussian matrices are randomly generated from the classical Bernoulli distribution with equal probability and the standard Gaussian distribution, respectively. For Hadamard matrices, first a square Hadamard matrix of size n (n is a power of 2) is generated, then its rows are randomly permuted and its first m rows are taken as an $m \times n$ sensing matrix. All $m \times n$ matrices are normalized to have columns of unit length.

We compare our recovery error bounds based on $\omega_\diamond$ with those based on the RIC. Combining Corollary 3.2 and Theorem 3.4, we have for the Basis Pursuit

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq \frac{2\sqrt{2k}}{\omega_2(A, 2k)}\varepsilon, \tag{6.4}$$

and for the Dantzig selector

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq \frac{2\sqrt{2k}}{\omega_\infty(A^T A, 2k)}\mu. \tag{6.5}$$

For comparison, the two RIC bounds are

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq \frac{4\sqrt{1 + \delta_{2k}(A)}}{1 - (1 + \sqrt{2})\delta_{2k}(A)}\varepsilon, \tag{6.6}$$

for the Basis Pursuit, assuming $\delta_{2k}(A) < \sqrt{2} - 1$ [7], and

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq \frac{4\sqrt{k}}{1 - \delta_{2k}(A) - \delta_{3k}(A)}\mu, \tag{6.7}$$

for the Dantzig selector, assuming $\delta_{2k}(A) + \delta_{3k}(A) < 1$ [10]. Without loss of generality, we set $\varepsilon = 1$ and $\mu = 1$.

The RIC is computed using Monte Carlo simulations. More explicitly, for $\delta_{2k}(A)$, we randomly take 1000 sub-matrices of $A \in \mathbb{R}^{m \times n}$ of size $m \times 2k$, compute the maximal and minimal singular values σ_1 and σ_{2k}, and approximate $\delta_{2k}(A)$ using the maximum of $\max(\sigma_1^2 - 1, 1 - \sigma_{2k}^2)$ among all sampled sub-matrices. Obviously, the approximated RIC is always smaller than or equal to the exact RIC. As a consequence, the performance bounds based on the exact RIC are *worse* than those based on the approximated RIC. Therefore, in cases where our $\omega_\diamond$ based bounds are better (tighter, smaller) than the approximated RIC bounds, they are even better than the exact RIC bounds.

In Tables 6.1, 6.2, and 6.3, we compare the error bounds (6.4) and (6.6) for the Basis Pursuit algorithm. In the tables, we also include s_* computed by (4.2), and $k_* = \lfloor s_*/2 \rfloor$, *i.e.*, the maximal sparsity level such that the sufficient and necessary condition (2.7) holds. The number of measurements is taken as $m = \lfloor \rho n \rfloor, \rho = 0.2, 0.3, \ldots, 0.8$. Note the blanks mean that the corresponding bounds are not valid. For the Bernoulli and Gaussian matrices, the RIC bounds work only for $k \leq 2$, even with $m = \lfloor 0.8n \rfloor$, while the $\omega_2(A, 2k)$ bounds work up until $k = 9$. Both bounds are better for Hadamard matrices. For example, when $m = 0.5n$, the RIC bounds are valid for $k \leq 3$, and our bounds hold for $k \leq 5$. In all cases for $n = 256$, our bounds are smaller than the RIC bounds.

We next compare the error bounds (6.5) and (6.7) for the Dantzig selector. For the Bernoulli and Gaussian matrices, our bounds work for wider ranges of (k, m) pairs and are tighter in all tested cases. For the Hadamard matrices, the RIC bounds are better, starting from $k \geq 5$ or 6. We expect that this indicates a general trend, namely, when k is relatively small, the ω based bounds are better, while when k is large, the RIC bounds are tighter. This was suggested by the probabilistic analysis of ω in Section 5. The reason is that when k is relatively small, both the relaxation $\|\boldsymbol{x}\|_1 \leq 2k\|\boldsymbol{x}\|_\infty$ on the sufficient and necessary condition (2.7) and the relaxation $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq \sqrt{2k}\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_\infty$ are sufficiently tight.

Table 6.1: Comparison of the ω_2 based bounds and the RIC based bounds on the ℓ_2 norms of the errors of the Basis Pursuit algorithm for a Bernoulli matrix with leading dimension $n = 256$.

	m	51	77	102	128	154	179	205
	s_*	4.6	6.1	7.4	9.6	12.1	15.2	19.3
k	k_*	2	3	3	4	6	7	9
1	ω bd	4.2	3.8	3.5	3.4	3.3	3.2	3.2
	ric bd			23.7	16.1	13.2	10.6	11.9
2	ω bd	31.4	12.2	9.0	7.4	6.5	6.0	5.6
	ric bd						72.1	192.2
3	ω bd		252.0	30.9	16.8	12.0	10.1	8.9
	ric bd							
4	ω bd				52.3	23.4	16.5	13.6
	ric bd							
5	ω bd					57.0	28.6	20.1
	ric bd							
6	ω bd					1256.6	53.6	30.8
	ric bd							
7	ω bd						161.6	50.6
	ric bd							
8	ω bd							93.1
	ric bd							
9	ω bd							258.7
	ric bd							

In Table 6.7 we present the execution times for computing different ω. For random matrices with leading dimension $n = 256$, the algorithm generally takes 1 to 3 minutes to compute either $\omega_2(A, s)$ or $\omega_\infty(A^T A, s)$.

In the last set of experiments, we compute $\omega_2(A, 2k)$ and $\omega_\infty(A^T A, 2k)$ for a Gaussian matrix and a Hadamard matrix, respectively, with leading dimension $n = 512$. The row dimensions of the sensing matrices range over $m = \lfloor \rho n \rfloor$ with $\rho = 0.2, 0.3, \ldots, 0.8$. In Figure 6.1, we compare the ℓ_2 norm error bounds of the Basis Pursuit using $\omega_2(A, 2k)$ and the RIC. The color indicates the values of the error bounds. We remove all bounds that are greater than 50 or are not valid. Hence, all white areas indicate that the bounds corresponding to (k, m) pairs that are too large or not valid. The left sub-figure is based on $\omega_2(A, 2k)$ and the right sub-figure is based on the RIC. We observe that the $\omega_2(A, 2k)$ based bounds apply to a wider range of (k, m) pairs.

In Figure 6.2, we conduct the same experiment as in Figure 6.1 for a Hadamard matrix and the Dantzig selector. We observe that for the Hadamard matrix, the RIC gives better performance bounds. This result coincides with the one we obtained in Table 6.5.

The average time for computing each $\omega_2(A, 2k)$ and $\omega_\infty(A^T A, 2k)$ was around 15 minutes.

7. Conclusions. In this paper, we analyzed the performance of ℓ_1 sparse signal recovery algorithms using the ℓ_∞ norm of the errors as a performance criterion.

Fig. 6.1: $\omega_2(A, 2k)$ based bounds v.s. RIC based bounds on the ℓ_2 norms of the errors for a Gaussian matrix with leading dimension $n = 512$. Left: $\omega_2(A, 2k)$ based bounds; Right: RIC based bounds.
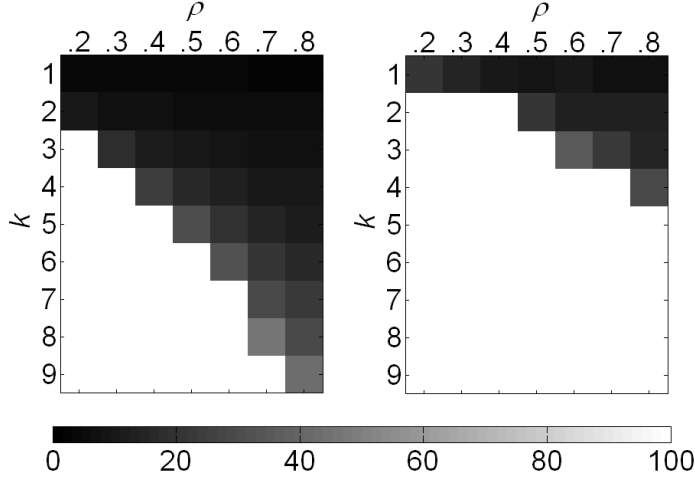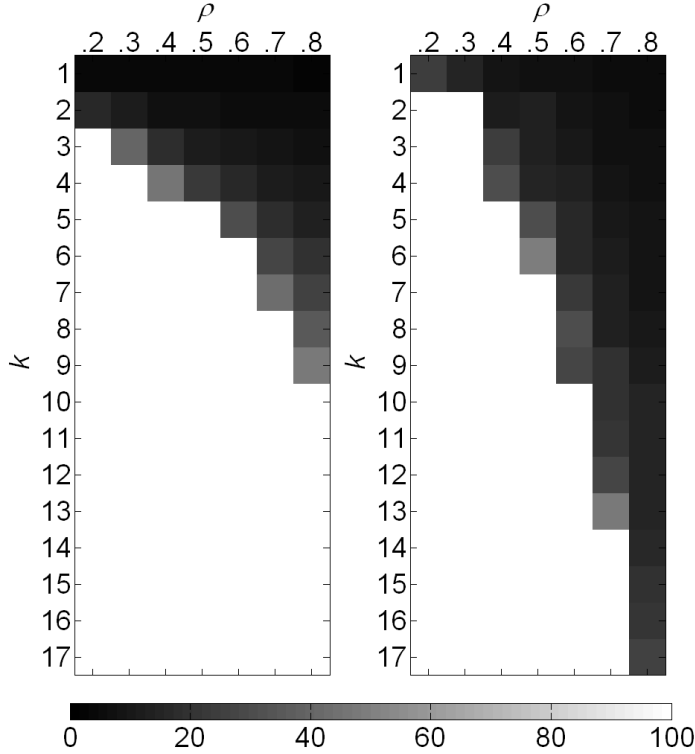

Fig. 6.2: $\omega_\infty(A^T A, 2k)$ based bounds v.s. RIC based bounds on the ℓ_2 norms of the errors for a Hadamard matrix with leading dimension $n = 512$. Left: $\omega_2(A, 2k)$ based bounds; Right: RIC based bounds

Table 6.2: Comparison of the ω_2 based bounds and the RIC based bounds on the ℓ_2 norms of the errors of the Basis Pursuit algorithm for a Hadamard matrix with leading dimension $n = 256$.

k		51	77	102	128	154	179	205
	m	51	77	102	128	154	179	205
	s_*	5.4	7.1	9.1	11.4	14.0	18.4	25.3
k	k_*	2	3	4	5	6	9	12
1	ω bd	3.8	3.5	3.3	3.2	3.1	3.0	3.0
	ric bd	46.6	13.2	9.2	9.4	8.3	6.2	5.2
2	ω bd	13.7	8.4	6.7	5.9	5.4	4.9	4.6
	ric bd			46.6	24.2	15.3	8.6	7.1
3	ω bd		30.9	14.0	10.1	8.4	7.1	6.3
	ric bd				1356.6	25.4	10.3	8.8
4	ω bd			47.4	18.9	13.2	9.9	8.1
	ric bd					40.0	14.0	10.2
5	ω bd				51.5	22.6	13.8	10.3
	ric bd						18.8	11.6
6	ω bd					50.8	20.1	13.1
	ric bd						42.5	15.9
7	ω bd						31.8	16.7
	ric bd						94.2	19.7
8	ω bd						63.5	21.7
	ric bd						1000.0	24.6
9	ω bd						449.8	29.4
	ric bd							39.1
10	ω bd							42.8
	ric bd							35.6
11	ω bd							72.7
	ric bd							134.1
12	ω bd							195.1
	ric bd							

We expressed other popular performance criteria in terms of the ℓ_∞ norm. A family of goodness measures of the sensing matrices was defined using optimization procedures. We used these goodness measures to derive upper bounds on the ℓ_∞ norms of the reconstruction errors for the Basis Pursuit, the Dantzig selector, and the LASSO estimator. Polynomial-time algorithms with established convergence properties were implemented to efficiently solve the optimization procedures defining the goodness measures. We expect that these goodness measures will be useful in comparing different sensing systems and recovery algorithms, as well as in designing optimal sensing matrices. In future work, we will use these computable performance bounds to optimally design $k-$space sample trajectories for MRI and to optimally design transmitting waveforms for compressive sensing radar.

8. Appendix: Proofs.

8.1. Proof of Proposition 3.1. *Proof.* [Proof of Proposition 3.1] Suppose $S = \mathrm{supp}(\boldsymbol{x})$ and $|S| = \|\boldsymbol{x}\|_0 = k$. Define the error vector $\boldsymbol{h} = \hat{\boldsymbol{x}} - \boldsymbol{x}$. For any vector $\boldsymbol{z} \in \mathbb{R}^n$ and any index set $S \subseteq \{1, \ldots, n\}$, we use $\boldsymbol{z}_S \in \mathbb{R}^{|S|}$ to represent the vector

Table 6.3: Comparison of the ω_2 based bounds and the RIC based bounds on the ℓ_2 norms of the errors of the Basis Pursuit algorithm for a Gaussian matrix with leading dimension $n = 256$.

	m	51	77	102	128	154	179	205
	s_*	4.6	6.2	8.1	9.9	12.5	15.6	20.0
k	k_*	2	3	4	4	6	7	10
1	ω bd	4.3	3.7	3.5	3.4	3.3	3.2	3.2
	ric bd			26.0	14.2	10.0	10.9	12.1
2	ω bd	34.3	12.3	8.3	7.0	6.4	5.9	5.6
	ric bd						47.1	27.6
3	ω bd		197.4	23.4	14.5	11.6	9.8	8.9
	ric bd							
4	ω bd			1036.6	39.6	21.7	15.9	13.4
	ric bd							
5	ω bd					49.3	26.4	20.0
	ric bd							
6	ω bd					284.2	48.8	31.2
	ric bd							
7	ω bd						129.1	48.1
	ric bd							
8	ω bd							185.5
	ric bd							
9	ω bd							9640.3
	ric bd							

whose elements are those of $\boldsymbol{z}$ indicated by S.

We first deal with the Basis Pursuit and the Dantzig selector. As observed by Candés in [7], the fact that $\|\hat{\boldsymbol{x}}\|_1 = \|\boldsymbol{x} + \boldsymbol{h}\|_1$ is the minimum among all $\boldsymbol{z}$s satisfying the constraints in (2.2) and (2.3), together with the fact that the true signal $\boldsymbol{x}$ satisfies the constraints as required by the conditions imposed on the noise in Proposition 3.1, imply that $\|\boldsymbol{h}_{S^c}\|_1$ cannot be very large. To see this, note that

$$\begin{aligned}
\|\boldsymbol{x}\|_1 &\geq \|\boldsymbol{x} + \boldsymbol{h}\|_1 \\
&= \sum_{i \in S} |\boldsymbol{x}_i + \boldsymbol{h}_i| + \sum_{i \in S^c} |\boldsymbol{x}_i + \boldsymbol{h}_i| \\
&\geq \|\boldsymbol{x}_S\|_1 - \|\boldsymbol{h}_S\|_1 + \|\boldsymbol{h}_{S^c}\|_1 \\
&= \|\boldsymbol{x}\|_1 - \|\boldsymbol{h}_S\|_1 + \|\boldsymbol{h}_{S^c}\|_1.
\end{aligned} \tag{8.1}$$

Therefore, we obtain $\|\boldsymbol{h}_S\|_1 \geq \|\boldsymbol{h}_{S^c}\|_1$, which leads to

$$2\|\boldsymbol{h}_S\|_1 \geq \|\boldsymbol{h}_S\|_1 + \|\boldsymbol{h}_{S^c}\|_1 = \|\boldsymbol{h}\|_1. \tag{8.2}$$

We now turn to the LASSO estimator(2.4). We use the proof technique in [8] (see also [3]). Since the noise $\boldsymbol{w}$ satisfies $\|A^T \boldsymbol{w}\|_\infty \leq \kappa\mu$ for some small $\kappa > 0$, and $\hat{\boldsymbol{x}}$ is a

Table 6.4: Comparison of the ω_∞ based bounds and the RIC based bounds on the ℓ_2 norms of the errors of the Dantzig selector algorithm for the Bernoulli matrix used in Table 6.1.

	m	51	77	102	128	154	179	205
	s_*	4.6	6.1	7.4	9.6	12.1	15.2	19.3
k	k_*	2	3	3	4	6	7	9
1	ω bd	6.0	5.4	4.8	4.4	4.2	4.1	4.1
	ric bd		46.3	17.4	12.1	11.2	10.3	8.6
2	ω bd	102.8	38.4	29.0	18.5	14.1	12.8	11.9
	ric bd						47.2	22.5
3	ω bd		1477.2	170.2	81.2	57.0	41.1	32.6
	ric bd							
4	ω bd				522.7	194.6	128.9	89.0
	ric bd							
5	ω bd					768.7	323.6	203.2
	ric bd							
6	ω bd					24974.0	888.7	489.0
	ric bd							
7	ω bd						3417.3	1006.9
	ric bd							
8	ω bd							2740.0
	ric bd							
9	ω bd							10196.9
	ric bd							

solution to (2.4), we have

$$\frac{1}{2}\|A\hat{\boldsymbol{x}} - \boldsymbol{y}\|_2^2 + \mu\|\hat{\boldsymbol{x}}\|_1 \leq \frac{1}{2}\|A\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mu\|\boldsymbol{x}\|_1.$$

Consequently, substituting $\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{w}$ yields

$$\begin{aligned}
\mu\|\hat{\boldsymbol{x}}\|_1 &\leq \frac{1}{2}\|A\boldsymbol{x} - \boldsymbol{y}\|_2^2 - \frac{1}{2}\|A\hat{\boldsymbol{x}} - \boldsymbol{y}\|_2^2 + \mu\|\boldsymbol{x}\|_1 \\
&= \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \frac{1}{2}\|A(\hat{\boldsymbol{x}} - \boldsymbol{x}) - \boldsymbol{w}\|_2^2 + \mu\|\boldsymbol{x}\|_1 \\
&= \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \frac{1}{2}\|A(\hat{\boldsymbol{x}} - \boldsymbol{x})\|_2^2 \\
&\quad + \langle A(\hat{\boldsymbol{x}} - \boldsymbol{x}), \boldsymbol{w}\rangle - \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \mu\|\boldsymbol{x}\|_1 \\
&\leq \langle A(\hat{\boldsymbol{x}} - \boldsymbol{x}), \boldsymbol{w}\rangle + \mu\|\boldsymbol{x}\|_1 \\
&= \langle \hat{\boldsymbol{x}} - \boldsymbol{x}, A^T\boldsymbol{w}\rangle + \mu\|\boldsymbol{x}\|_1.
\end{aligned}$$

Using the Cauchy-Swcharz type inequality, we get

$$\begin{aligned}
\mu\|\hat{\boldsymbol{x}}\|_1 &\leq \|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_1\|A^T\boldsymbol{w}\|_\infty + \mu\|\boldsymbol{x}\|_1 \\
&= \kappa\mu\|\boldsymbol{h}\|_1 + \mu\|\boldsymbol{x}\|_1,
\end{aligned}$$

Table 6.5: Comparison of the ω_∞ based bounds and the RIC based bounds on the ℓ_2 norms of the errors of the Dantzig selector algorithm for the Hadamard matrix used in Table 6.2.

	m	51	77	102	128	154	179	205
	s_*	5.2	6.9	9.1	12.1	14.4	18.3	25.2
k	k_*	2	3	4	6	7	9	12
1	ω bd	4.8	4.0	3.8	3.4	3.4	3.2	3.1
	ric bd		15.6	9.3	7.0	6.3	5.8	5.1
2	ω bd	50.9	16.2	10.1	7.1	7.0	6.1	5.3
	ric bd			45.3	16.6	13.7	10.6	8.8
3	ω bd		108.2	30.7	14.3	13.9	10.0	8.0
	ric bd			1016.4	29.9	24.9	15.8	12.5
4	ω bd			150.7	35.3	29.3	16.8	11.7
	ric bd				126.4	38.7	24.2	16.6
5	ω bd				108.5	64.2	31.4	17.3
	ric bd					187.3	30.0	22.1
6	ω bd				3168.9	171.5	59.7	25.3
	ric bd					112.0	53.1	26.8
7	ω bd					1499.5	116.3	38.8
	ric bd					411.7	71.3	34.7
8	ω bd						265.3	61.4
	ric bd						95.4	47.6
9	ω bd						2394.0	96.0
	ric bd						198.7	61.9
10	ω bd							157.4
	ric bd							82.9
11	ω bd							296.4
	ric bd							130.3
12	ω bd							898.2
	ric bd							201.2

which leads to

$$\|\hat{\boldsymbol{x}}\|_1 \leq \kappa\|\boldsymbol{h}\|_1 + \|\boldsymbol{x}\|_1.$$

Therefore, similar to the argument in (8.1), we have

$$
\begin{aligned}
&\|\boldsymbol{x}\|_1 \\
&\geq \|\hat{\boldsymbol{x}}\|_1 - \kappa\|\boldsymbol{h}\|_1 \\
&= \|\boldsymbol{x} + \boldsymbol{h}_{S^c} + \boldsymbol{h}_S\|_1 - \kappa\left(\|\boldsymbol{h}_{S^c} + \boldsymbol{h}_S\|_1\right) \\
&\geq \|\boldsymbol{x} + \boldsymbol{h}_{S^c}\|_1 - \|\boldsymbol{h}_S\|_1 - \kappa\left(\|\boldsymbol{h}_{S^c}\|_1 + \|\boldsymbol{h}_S\|_1\right) \\
&= \|\boldsymbol{x}\|_1 + (1 - \kappa)\|\boldsymbol{h}_{S^c}\|_1 - (1 + \kappa)\|\boldsymbol{h}_S\|_1,
\end{aligned}
$$

where $S = \text{supp}(\boldsymbol{x})$. Consequently, we have

$$\|\boldsymbol{h}_S\|_1 \geq \frac{1 - \kappa}{1 + \kappa}\|\boldsymbol{h}_{S^c}\|_1.$$

Table 6.6: Comparison of the ω_∞ based bounds and the RIC based bounds on the ℓ_2 norms of the errors of the Dantzig selector algorithm for the Gaussian matrix used in Table 6.3.

	m	51	77	102	128	154	179	205
	s_*	4.6	6.2	8.1	9.9	12.5	15.6	20.0
k	k_*	2	3	4	4	6	7	10
1	ω bd	6.5	5.1	4.8	4.3	4.2	4.0	3.9
	ric bd		30.0	18.0	14.6	9.7	9.3	9.1
2	ω bd	119.4	37.8	22.5	17.6	14.1	12.7	11.4
	ric bd					91.5	44.4	23.5
3	ω bd		1216.7	120.7	67.3	53.6	38.7	36.4
	ric bd							2546.6
4	ω bd			4515.9	318.2	168.4	115.8	109.0
	ric bd							
5	ω bd					663.6	292.4	247.8
	ric bd							
6	ω bd					5231.4	764.3	453.5
	ric bd							
7	ω bd						2646.4	1087.7
	ric bd							
8	ω bd							2450.5
	ric bd							
9	ω bd							6759.0
	ric bd							

Therefore, similar to (8.2), we obtain

$$
\begin{aligned}
\frac{2}{1-\kappa}\|\boldsymbol{h}_S\|_1 &\geq \frac{1+\kappa}{1-\kappa}\|\boldsymbol{h}_S\|_1 + \frac{1-\kappa}{1-\kappa}\|\boldsymbol{h}_S\|_1 \\
&\geq \frac{1+\kappa}{1-\kappa}\frac{1-\kappa}{1+\kappa}\|\boldsymbol{h}_{S^c}\|_1 + \frac{1-\kappa}{1-\kappa}\|\boldsymbol{h}_S\|_1 \\
&= \|\boldsymbol{h}\|_1.
\end{aligned}
\tag{8.3}
$$

□

8.2. Proof of Theorem 4.2. *Proof.*

1. Since in the optimization problem defining $f_{s,i}(\eta)$, the objective function $\boldsymbol{z}_i$ is continuous, and the constraint correspondence

$$
\begin{aligned}
C(\eta) &: [0,\infty) \twoheadrightarrow \mathbb{R}^n \\
\eta &\mapsto \{\boldsymbol{z} : \|Q\boldsymbol{z}\|_\diamond \leq 1, \|\boldsymbol{z}\|_1 \leq s\eta\}
\end{aligned}
\tag{8.4}
$$

   is compact-valued and continuous (both upper and lower hemicontinuous), according to Berge's Maximum Theorem [2], the optimal value function $f_{s,i}(\eta)$ is continuous. The continuity of $f_s(\eta)$ follows from that finite maximization preserves the continuity.

2. To show the strict increasing property, suppose $0 < \eta_1 < \eta_2$ and the dual

Table 6.7: Time in seconds taken to compute $\omega_2(A, \cdot)$ and $\omega_\infty(A^T A, \cdot)$ for Bernoulli, Hadamard, and Gaussian matrices

k	type	m	51	77	102	128	154	179	205
1	Bernoulli	ω_2	118	84	133	87	133	174	128
		ω_∞	75	81	84	65	63	144	151
	Hadamard	ω_2	84	82	82	82	80	79	79
		ω_∞	57	55	58	58	58	58	57
	Gaussian	ω_2	82	84	212	106	156	185	104
		ω_∞	69	65	72	102	81	104	72
3	Bernoulli	ω_2		155	96	95	97	97	131
		ω_∞		300	228	190	125	135	196
	Hadamard	ω_2		91	88	87	88	74	72
		ω_∞		84	83	77	92	102	70
	Gaussian	ω_2		134	168	115	95	96	100
		ω_∞		137	142	125	165	145	105
5	Bernoulli	ω_2					97	111	97
		ω_∞					156	81	107
	Hadamard	ω_2				87	85	85	81
		ω_∞				75	74	75	75
	Gaussian	ω_2					98	105	96
		ω_∞							193
7	Bernoulli	ω_2						164	104
		ω_∞						178	85
	Hadamard	ω_2						82	77
		ω_∞						134	71	65
	Gaussian	ω_2						106	105
		ω_∞							193

variable $\boldsymbol{\lambda}_2^*$ achieves $f_{s,i}(\eta_2)$ in (4.10). Then we have

$$
\begin{aligned}
f_{s,i}(\eta_1) &\leq s\eta_1 \|\mathbf{e}_i - Q^T \boldsymbol{\lambda}_2^*\|_\infty + \|\boldsymbol{\lambda}_2\|_\diamond^* \\
&< s\eta_2 \|\mathbf{e}_i - Q^T \boldsymbol{\lambda}_2^*\|_\infty + \|\boldsymbol{\lambda}_2\|_\diamond^* \\
&= f_{s,i}(\eta_2).
\end{aligned}
\tag{8.5}
$$

The case for $\eta_1 = 0$ is proved by continuity, and the strict increasing of $f_s(\eta)$ follows immediately.

3. The concavity of $f_{s,i}(\eta)$ follows from the dual representation (4.10) and the fact that $f_{s,i}(\eta)$ is the minimization of a function of variables η and $\boldsymbol{\lambda}$, and when $\boldsymbol{\lambda}$, the variable to be minimized, is fixed, the function is linear in η.

4. Next we show that when $\eta > 0$ is sufficiently small $f_s(\eta) \geq s\eta$. Taking $\boldsymbol{z} = s\eta \mathbf{e}_i$, we have $\|\boldsymbol{z}\|_1 = s\eta$ and $\boldsymbol{z}_i = s\eta > \eta$ (recall $s \in (1, \infty)$). In addition, when $0 < \eta \leq 1/(s\|Q_i\|_\diamond)$, we also have $\|Q\boldsymbol{z}\|_\diamond \leq 1$. Therefore, for sufficiently small η, we have $f_{s,i}(\eta) \geq s\eta > \eta$. Clearly, $f_s(\eta) = \max_i f_{s,i}(\eta) \geq s\eta > \eta$ for such η.

Recall that

$$
\frac{1}{s_*} = \max_i \min_{\boldsymbol{\lambda}_i} \|\mathbf{e}_i - Q^T \boldsymbol{\lambda}_i\|_\infty.
\tag{8.6}
$$

Suppose $\boldsymbol{\lambda}_i^*$ is the optimal solution for each $\min_{\boldsymbol{\lambda}_i} \|\mathbf{e}_i - Q^T\boldsymbol{\lambda}_i\|_\infty$. For each i, we then have

$$\frac{1}{s_*} \geq \|\mathbf{e}_i - Q^T\boldsymbol{\lambda}_i^*\|_\infty, \tag{8.7}$$

which implies

$$\begin{aligned}
f_{s,i}(\eta) &= \min_{\boldsymbol{\lambda}_i} s\eta\|\mathbf{e}_i - Q^T\boldsymbol{\lambda}_i\|_\infty + \|\boldsymbol{\lambda}_i\|_\diamond^* \\
&\leq s\eta\|\mathbf{e}_i - Q^T\boldsymbol{\lambda}_i^*\|_\infty + \|\boldsymbol{\lambda}_i^*\|_\diamond^* \\
&\leq \frac{s}{s_*}\eta + \|\boldsymbol{\lambda}_i^*\|_\diamond^*. \tag{8.8}
\end{aligned}$$

As a consequence, we obtain

$$f_s(\eta) = \max_i f_{s,i}(\eta) \leq \frac{s}{s_*}\eta + \max_i \|\boldsymbol{\lambda}_i^*\|_\diamond^*. \tag{8.9}$$

Pick $\rho \in (s/s_*, 1)$. Then, we have the following when $\eta > \max_i \|\boldsymbol{\lambda}_i^*\|_\diamond^*/(\rho - s/s_*)$:

$$\begin{aligned}
f_{s,i}(\eta) &\leq \rho\eta, i = 1, \ldots, n, \text{ and} \\
f_s(\eta) &\leq \rho\eta. \tag{8.10}
\end{aligned}$$

5. We first show the existence and uniqueness of the positive fixed points for $f_{s,i}(\eta)$. The properties 1) and 4) imply that $f_{s,i}(\eta)$ has at least one positive fixed point. (Interestingly, 2) and 4) also imply the existence of a positive fixed point, see [28].) To prove uniqueness, suppose there are two fixed points $0 < \eta_1^* < \eta_2^*$. Pick η_0 small enough such that $f_{s,i}(\eta_0) > \eta_0 > 0$ and $\eta_0 < \eta_1^*$. Then $\eta_1^* = \lambda\eta_0 + (1-\lambda)\eta_2^*$ for some $\lambda \in (0,1)$, which implies that $f_{s,i}(\eta_1^*) \geq \lambda f_{s,i}(\eta_0) + (1-\lambda)f_{s,i}(\eta_2^*) > \lambda\eta_0 + (1-\lambda)\eta_2^* = \eta_1^*$ due to the concavity, contradicting with $\eta_1^* = f_{s,i}(\eta_1^*)$.
The set of positive fixed point for $f_s(\eta)$, $\{\eta \in (0,\infty) : \eta = f_s(\eta) = \max_i f_{s,i}(\eta)\}$, is a subset of $\bigcup_{i=1}^p \{\eta \in (0,\infty) : \eta = f_{s,i}(\eta)\} = \{\eta_i^*\}_{i=1}^n$. We argue that

$$\eta^* = \max_i \eta_i^* \tag{8.11}$$

is the unique positive fixed point for $f_s(\eta)$.
We proceed to show that η^* is a fixed point of $f_s(\eta)$. Suppose η^* is a fixed point of $f_{s,i_0}(\eta)$, then it suffices to show that $f_s(\eta^*) = \max_i f_{s,i}(\eta^*) = f_{s,i_0}(\eta^*)$. If this is not the case, there exists $i_1 \neq i_0$ such that $f_{s,i_1}(\eta^*) > f_{s,i_0}(\eta^*) = \eta^*$. The continuity of $f_{s,i_1}(\eta)$ and the property 4) imply that there exists $\eta > \eta^*$ with $f_{s,i_1}(\eta) = \eta$, contradicting with the definition of η^*.
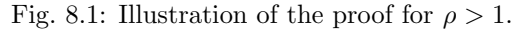To show the uniqueness, suppose η_1^* is fixed point of $f_{s,i_1}(\eta)$ satisfying $\eta_1^* < \eta^*$. Then, we must have $f_{s,i_0}(\eta_1^*) > f_{s,i_1}(\eta_1^*)$ because otherwise the continuity implies the existence of another fixed point of $f_{s,i_0}(\eta)$. As a consequence, $f_s(\eta_1^*) > f_{s,i_1}(\eta_1^*) = \eta_1^*$ and η_1^* is not a fixed point of $f_s(\eta)$.

6. Next we show $\eta^* = \gamma^* \overset{\text{def}}{=} 1/\omega_\diamond(Q, s)$. We first prove $\gamma^* \geq \eta^*$ for the fixed point $\eta^* = f_s(\eta^*)$. Suppose $\boldsymbol{z}^*$ achieves the optimization problem defining $f_s(\eta^*)$, then we have

$$\eta^* = f_s(\eta^*) = \|\boldsymbol{z}^*\|_\infty, \|Q\boldsymbol{z}^*\|_\diamond \leq 1, \text{and } \|\boldsymbol{z}^*\|_1 \leq s\eta^*. \tag{8.12}$$

Since $\|z^*\|_1/\|z^*\|_\infty \le s\eta^*/\eta^* \le s$, we have

$$\gamma^* \ge \frac{\|z^*\|_\infty}{\|Qz^*\|_\diamond} \ge \eta^*. \tag{8.13}$$

If $\eta^* < \gamma^*$, we define $\eta_0 = (\eta^* + \gamma^*)/2$ and

$$z^c = \operatorname{argmax}_z \frac{s\|z\|_\infty}{\|z\|_1} \text{ s.t. } \|Qz\|_\diamond \le 1, \|z\|_\infty \ge \eta_0, \tag{8.14}$$

$$\rho = \frac{s\|z^c\|_\infty}{\|z^c\|_1}. \tag{8.15}$$

Suppose z^{**} with $\|Qz^{**}\|_\diamond = 1$ achieves the optimum of the optimization (4.6) defining $\gamma^* = 1/\omega_\diamond(Q, s)$. Clearly, $\|z^{**}\|_\infty = \gamma^* > \eta_0$, which implies z^{**} is a feasible point of the optimization problem (8.14) defining z^c and ρ. As a consequence, we have

$$\rho \ge \frac{s\|z^{**}\|_\infty}{\|z^{**}\|_1} \ge 1. \tag{8.16}$$

Fig. 8.1: Illustration of the proof for $\rho > 1$.

Actually we will show that $\rho > 1$. If $\|z^{**}\|_1 < s\|z^{**}\|_\infty$, we are done. If not (*i.e.*, $\|z^{**}\|_1 = s\|z^{**}\|_\infty$), as illustrated in Figure 8.1, we consider $\boldsymbol{\xi} = \frac{\eta_0}{\gamma^*} z^{**}$, which satisfies

$$\|Q\boldsymbol{\xi}\|_\diamond = \frac{\eta_0}{\gamma^*} < 1, \tag{8.17}$$

$$\|\boldsymbol{\xi}\|_\infty = \eta_0, \text{ and} \tag{8.18}$$

$$\|\boldsymbol{\xi}\|_1 = s\eta_0. \tag{8.19}$$

To get $\boldsymbol{\xi}^n$ as shown in Figure 8.1, pick the component of $\boldsymbol{\xi}$ with the smallest non-zero absolute value, and scale that component by a small positive constant less than 1. Because $s > 1$, $\boldsymbol{\xi}$ has more than one non-zero components,

implying $\|\boldsymbol{\xi}^{\mathrm{n}}\|_\infty$ will remain the same. If the scaling constant is close enough to 1, $\|Q\boldsymbol{\xi}^{\mathrm{n}}\|_\diamond$ will remain less than 1 due to continuity. But the good news is that $\|\boldsymbol{\xi}^{\mathrm{n}}\|_1$ decreases, and hence $\rho \geq \frac{s\|\boldsymbol{\xi}^{\mathrm{n}}\|_\infty}{\|\boldsymbol{\xi}^{\mathrm{n}}\|_1}$ becomes greater than 1.

Now we proceed to obtain a contradiction that $f_s(\eta^*) > \eta^*$. If $\|\boldsymbol{z}^{\mathrm{c}}\|_1 \leq s \cdot \eta^*$, then it is a feasible point of

$$\max_{\boldsymbol{z}} \|\boldsymbol{z}\|_\infty \text{ s.t. } \|Q\boldsymbol{z}\|_\diamond \leq 1, \|\boldsymbol{z}\|_1 \leq s \cdot \eta^*. \tag{8.20}$$

As a consequence, $f_s(\eta^*) \geq \|\boldsymbol{z}^{\mathrm{c}}\|_\infty \geq \eta_0 > \eta^*$, contradicting with η^* is a fixed point and we are done. If this is not the case, $i.e.$, $\|\boldsymbol{z}^{\mathrm{c}}\|_1 > s \cdot \eta^*$, we define a new point

$$\boldsymbol{z}^{\mathrm{n}} = \tau \boldsymbol{z}^{\mathrm{c}} \tag{8.21}$$

with

$$\tau = \frac{s \cdot \eta^*}{\|\boldsymbol{z}^{\mathrm{c}}\|_1} < 1. \tag{8.22}$$

Note that $\boldsymbol{z}^{\mathrm{n}}$ is a feasible point of the optimization problem defining $f_s(\eta^*)$ since

$$\|Q\boldsymbol{z}^{\mathrm{n}}\|_\diamond = \tau\|Q\boldsymbol{z}^{\mathrm{c}}\|_\diamond < 1, \text{ and} \tag{8.23}$$
$$\|\boldsymbol{z}^{\mathrm{n}}\|_1 = \tau\|\boldsymbol{z}^{\mathrm{c}}\|_1 = s \cdot \eta^*. \tag{8.24}$$

Furthermore, we have

$$\|\boldsymbol{z}^{\mathrm{n}}\|_\infty = \tau\|\boldsymbol{z}^{\mathrm{c}}\|_\infty = \rho\eta^*. \tag{8.25}$$

As a consequence, we obtain a contradiction
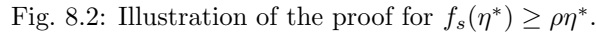
$$f_s(\eta^*) \geq \rho\eta^* > \eta^*. \tag{8.26}$$

Therefore, for the fixed point η^*, we have $\eta^* = \gamma^* = 1/\omega_\diamond(Q, s)$.

7. This property simply follows from the continuity, the uniqueness, and property 4).

8. We use contradiction to show the existence of $\rho_1(\epsilon)$ in 8). In view of 4), we need only to show the existence of such a $\rho_1(\epsilon)$ that works for $\eta_L \leq \eta \leq (1 - \epsilon)\eta^*$ where $\eta_L = \sup\{\eta : f_s(\xi) \geq s\xi, \forall 0 < \xi \leq \eta\}$. Suppose otherwise, we then construct sequences $\{\eta^{(k)}\}_{k=1}^\infty \subset [\eta_L, (1 - \epsilon)\eta^*]$ and $\{\rho_1^{(k)}\}_{k=1}^\infty \subset (1, \infty)$ with

$$\lim_{k\to\infty} \rho_1^{(k)} = 1,$$
$$f_s(\eta^{(k)}) \leq \rho^{(k)}\eta^{(k)}. \tag{8.27}$$

Due to the compactness of $[\eta_L, (1 - \epsilon)\eta^*]$, there must exist a subsequence $\{\eta^{(k_l)}\}_{l=1}^\infty$ of $\{\eta^{(k)}\}$ such that $\lim_{l\to\infty} \eta^{(k_l)} = \eta_{\lim}$ for some $\eta_{\lim} \in [\eta_L, (1 - \epsilon)\eta^*]$. As a consequence of the continuity of $f_s(\eta)$, we have

$$f_s(\eta_{\lim}) = \lim_{l\to\infty} f_s(\eta^{(k_l)}) \leq \lim_{l\to\infty} \rho_1^{(k_l)}\eta^{(k_l)} = \eta_{\lim}. \tag{8.28}$$

Fig. 8.2: Illustration of the proof for $f_s(\eta^*) \geq \rho\eta^*$.

Again due to the continuity of $f_s(\eta)$ and the fact that $f_s(\eta) < \eta$ for $\eta < \eta_L$, there exists $\eta_c \in [\eta_L, \eta_{\lim}]$ such that

$$f_s(\eta_c) = \eta_c, \tag{8.29}$$

contradicting with the uniqueness of the fixed point for $f_s(\eta)$. The existence of $\rho_2(\epsilon)$ can be proved in a similar manner.

□

REFERENCES

[1] R. BARANIUK AND P. STEEGHS, *Compressive radar imaging*, in IEEE Radar Conference, apr 2007, pp. 128–133.

[2] C. BERGE, *Topological Spaces*, Dover Publications, Mineola, NY, reprinted 1997 in paperback.

[3] P. BICKEL, Y. RITOV, AND A. TSYBAKOV, *Simultaneous analysis of Lasso and Dantzig selector*, Annals of Statistics, 37 (2009), pp. 1705–1732.

[4] H. BODLAENDER, P. GRITZMANN, V. KLEE, AND J. LEEUWEN, *Computational complexity of norm-maximization*, Combinatorica, 10 (1990), pp. 203–225.

[5] S. BOYD AND L. VANDENBERGHE, *Convex Optimization*, Cambridge University Press, 2004.

[6] E. CANDÉS AND J. ROMBERG, *ℓ₁-magic: Recovery of sparse signals via convex programming*, oct 2005.

[7] E. J. CANDÈS, *The restricted isometry property and its implications for compressed sensing*, Compte Rendus de l'Academie des Sciences, Paris, Serie I, 346 (2008), pp. 589–592.

[8] E. J. CANDÈS AND Y. PLAN, *Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements*, IEEE Trans. Inf. Theory, 57 (2011), pp. 2342–2359.

[9] E. J. CANDÈS, J. ROMBERG, AND T. TAO, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inf. Theory, 52 (2006), pp. 489–509.

[10] E. J. CANDÈS AND T. TAO, *The Dantzig selector: Statistical estimation when p is much larger than n*, Ann. Statist., 35 (2007), pp. 2313–2351.

[11] E. J. CANDÈS AND M. B. WAKIN, *An introduction to compressive sampling*, IEEE Signal Process. Mag., 25 (2008), pp. 21–30.

[12] S. CHEN, D. L. DONOHO, AND M. A. SAUNDERS, *Atomic decomposition by basis pursuit*, SIAM J. Sci. Comp., 20 (1998), pp. 33–61.

[13] A. COHEN, W. DAHMEN, AND R. DEVORE, *Compressed sensing and best k-term approximation*, J. Amer. Math. Soc., 22 (2009), pp. 211–231.

[14] A. D'ASPREMONT AND L. EL GHAOUI, *Testing the nullspace property using semidefinite programming*, Mathematical Programming, 127 (2011), pp. 123–144.

[15] A. D'ASPREMONT, L. EL GHAOUI, M. JORDAN, AND G. R. G. LANCKRIET, *A direct formulation for sparse PCA using semidefinite programming*, SIAM Review, 49 (2007), pp. 434–448.

[16] D. DONOHO, *High-dimensional centrally-symmetric polytopes with neighborliness proportional to dimension*, technical report, Department of Statistics, Stanford University, 2004.

[17] D. L. DONOHO AND X. HUO, *Uncertainty principles and ideal atomic decomposition*, IEEE Trans. Inf. Theory, 47 (2001), pp. 2845–2862.

[18] M. HERMAN AND T. STROHMER, *Compressed sensing radar*, in IEEE Radar Conference, may 2008, pp. 1–6.

[19] M. A. HERMAN AND T. STROHMER, *High-resolution radar via compressed sensing*, IEEE Trans. Signal Process., 57 (2009), pp. 2275–2284.

[20] A. JUDITSKY AND A. NEMIROVSKI, *On verifiable sufficient conditions for sparse signal recovery via ℓ_1 minimization*, Mathematical Programming, 127 (2011), pp. 57–88.

[21] M. LUSTIG, D. L DONOHO, AND J. M PAULY, *Sparse MRI: The application of compressed sensing for rapid MR imaging*, Magnetic Resonance in Medicine, 58 (2007), pp. 1182–1195.

[22] D. MALIOUTOV, M. CETIN, AND A. S. WILLSKY, *A sparse signal reconstruction perspective for source localization with sensor arrays*, IEEE Trans. Signal Process., 53 (2005), pp. 3010–3022.

[23] S. MENDELSON, A. PAJOR, AND N. TOMCZAK-JAEGERMANN, *Reconstruction and subgaussian operators in asymptotic geometric analysis*, Geometric And Functional Analysis, (2007), pp. 1248–1282.

[24] S. SEN, G. TANG, AND A. NEHORAI, *Multiobjective optimization of OFDM radar waveform for target detection*, IEEE Trans. Signal Process., 59 (2011), pp. 639–652.

[25] M. A. SHEIKH, S. SARVOTHAM, O. MILENKOVIC, AND R. G. BARANIUK, *DNA array decoding from nonlinear measurements by belief propagation*, in Proc. IEEE Workshop Statistical Signal Processing (SSP 2007), Madison, WI, aug 2007, pp. 215–219.

[26] G. TANG AND A. NEHORAI, *Performance analysis of sparse recovery based on constrained minimal singular values*, to appear in IEEE Trans. Signal Processing.

[27] ———, *Performance analysis for sparse support recovery*, IEEE Trans. Inf. Theory, 56 (2010), pp. 1383–1399.

[28] A. TARSKI, *A lattice-theoretical fix point theorem and its applications*, Pacific Journal of Mathematics, 5 (1955), pp. 285–309.

[29] R. TIBSHIRANI, *Regression shrinkage and selection via LASSO*, J. Roy. Statist. Soc. Ser. B, 58 (1996), pp. 267–288.

[30] Y. ZHANG, *A simple proof for recoverability of ℓ_1-minimization: go over or under?*, tech. report, Rice CAAM Department, 2005.