Verifiable and Computable ℓ_∞ Performance Evaluation of ℓ_1 Sparse Signal Recovery

Gongguo Tang and Arye Nehorai
Department of Electrical and Systems Engineering
Washington University in St. Louis
St. Louis, MO 63130-1127
Email: {gt2, nehorai}@ese.wustl.edu

***Abstract*—In this paper, we develop verifiable and computable performance analysis of the ℓ_∞ norms of the recovery errors for ℓ_1 minimization algorithms. 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 ℓ_1 recovery in the noise-free case. This implementation performs orders-of-magnitude faster than the state-of-the-art techniques. We derive performance bounds on the ℓ_∞ norms of the recovery errors in terms of these goodness measures. We establish connections between other performance criteria (*e.g.*, the ℓ_2 norm, ℓ_1 norm, and support recovery) and the ℓ_∞ norm in a tight manner. 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.**

***Index Terms*—compressive sensing, computable performance analysis, sparse signal recovery, verifiable sufficient condition**

I. Introduction

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 manner. More importantly, controlling the ℓ_∞ norm within half of the minimal non-zero absolute value of the sparse signal components would guarantee exact recovery of the support. Support recovery is arguably one of the most important and challenging problems in sparse signal recovery. Refer to [1] 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. 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 the restricted isometry constant (RIC) [2]. However, 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. 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 [3], sensor arrays [4], DNA microarrays [5], and MRI [6], 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. Consider the model (1) given in Section II, 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.

The paper is organized as follows. In Section II, 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 III, we derive performance bounds on the ℓ_∞ norms of the recovery errors for several convex relaxation algorithms. In Section IV, we design algorithms to verify a sufficient condition for exact ℓ_1

This work was supported by the Department of Defense under the Air Force Office of Scientific Research MURI Grant FA9550-05-1-0443, ONR Grant N000140810849, and the National Science Foundation, Grant No. CCF-1014908.

978-1-4244-9848-2/11$26.00©2011 IEEE

recovery in the noise-free case, and to compute the goodness measures of arbitrarily given sensing matrices. Section V is devoted to the probabilistic analysis of our ℓ_∞ performance measures. We evaluate the algorithms' performance in Section VI. Section VII summarizes our conclusions.

II. 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, we have $\|\boldsymbol{x}\|_\infty = \|\boldsymbol{x}\|_{1,1}$ and $\|\boldsymbol{x}\|_1 = \|\boldsymbol{x}\|_{n,1}$. We use $\|\cdot\|_\diamond$ to denote a general norm. The support of $\boldsymbol{x}$, $\mathrm{supp}(\boldsymbol{x})$, is the index set of the non-zero components of $\boldsymbol{x}$. The size of the support is the sparsity level of $\boldsymbol{x}$. Signals of sparsity level at most k are called $k-$sparse signals.

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

$$\boldsymbol{y} \quad = \quad A\boldsymbol{x} + \boldsymbol{w}, \tag{1}$$

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

We focus on three recovery algorithms based on ℓ_1 minimization: the Basis Pursuit [7], the Dantzig selector [8], and the LASSO estimator [9].

$$\text{Basis Pursuit: } \min_{\boldsymbol{z}\in\mathbb{R}^n} \|\boldsymbol{z}\|_1 \ \text{ s.t. } \|\boldsymbol{y} - A\boldsymbol{z}\|_\diamond \le \epsilon \tag{2}$$

$$\text{Dantzig: } \min_{\boldsymbol{z}\in\mathbb{R}^n} \|\boldsymbol{z}\|_1 \ \text{ s.t. } \ \|A^T(\boldsymbol{y} - A\boldsymbol{z})\|_\infty \le \mu \tag{3}$$

$$\text{LASSO: } \min_{\boldsymbol{z}\in\mathbb{R}^n} \frac{1}{2}\|\boldsymbol{y} - A\boldsymbol{z}\|_2^2 + \mu\|\boldsymbol{z}\|_1. \tag{4}$$

Here μ is a tuning parameter, and ϵ is the noise level.

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

$$\min_{\boldsymbol{z}\in\mathbb{R}^n} \|\boldsymbol{z}\|_1 \ \text{ s.t. } A\boldsymbol{z} = A\boldsymbol{x}. \tag{5}$$

A minimal requirement on ℓ_1 minimization algorithms is the *uniqueness and exactness* of the solution $\hat{\boldsymbol{x}} \stackrel{\text{def}}{=} \mathrm{argmin}_{\boldsymbol{z}:A\boldsymbol{z}=A\boldsymbol{x}}\|\boldsymbol{x}\|_1$, *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 expressed in terms of $\|\cdot\|_{k,1}$ is [10]–[12]

$$\|\boldsymbol{z}\|_{k,1} < \frac{1}{2}\|\boldsymbol{z}\|_1, \forall \boldsymbol{z} \in \mathrm{Ker}(A), \tag{6}$$

where $\mathrm{Ker}(A) \stackrel{\text{def}}{=} \{\boldsymbol{z} : A\boldsymbol{z} = 0\}$ is the kernel of A.

The approaches in [13] and [14] for verifying the sufficient condition (6) are based on relaxing the following optimization problem in various ways:

$$\alpha_k \quad = \quad \max_{\boldsymbol{z}} \|\boldsymbol{z}\|_{k,1} \text{ s.t. } A\boldsymbol{z} = 0, \|\boldsymbol{z}\|_1 \le 1. \tag{7}$$

Clearly, $\alpha_k < 1/2$ is necessary and sufficient for exact ℓ_1 recovery for $k-$sparse signals. Unfortunately, the direct computation of (7) for general k is extremely difficult: it is the maximization of a norm (convex function) over a polyhedron (convex set) [15]. In [13], in a very rough sense α_1 was computed by solving n linear programs:

$$\min_{\boldsymbol{y}_i\in\mathbb{R}^m} \|\mathbf{e}_i - A^T\boldsymbol{y}_i\|_\infty, i = 1, \cdots, n, \tag{8}$$

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 (6). However, in [16], we found that the primal-dual method of directly solving (6) as the following n linear programs

$$\max \boldsymbol{z}_i \text{ s.t. } \ A\boldsymbol{z} = 0, \|\boldsymbol{z}\|_1 \le 1 \tag{9}$$

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.

III. 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. Due to space limitation, all proofs in this paper are omitted. More details can be found in the full paper [17].

Proposition 1 *Suppose $\boldsymbol{x}$ in* (1) *is $k-$sparse and the noise $\boldsymbol{w}$ satisfies $\|\boldsymbol{w}\|_\diamond \le \epsilon$, $\|A^T\boldsymbol{w}\|_\infty \le \mu$, and $\|A^T\boldsymbol{w}\|_\infty \le \kappa\mu, \kappa \in (0,1)$, 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)*,* (3)*, and* (4)*. Then we have*

$$c\|\boldsymbol{h}\|_{k,1} \ge \|\boldsymbol{h}\|_1, \tag{10}$$

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 1 is to bound the ℓ_1 and ℓ_2 norms of the error vector using the ℓ_∞ norm:

Corollary 1 *Under the assumptions of Proposition 1, we have*

$$\|\boldsymbol{h}\|_1 \le ck\|\boldsymbol{h}\|_\infty, \ \|\boldsymbol{h}\|_2 \le \sqrt{ck}\|\boldsymbol{h}\|_\infty. \tag{11}$$

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

$$\mathrm{supp}(\max(|\hat{\boldsymbol{x}}| - \beta/2, 0) \quad = \quad \mathrm{supp}(\boldsymbol{x}), \tag{12}$$

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

For ease of presentation, we have the following definition:

Definition 1 *For any $s \in [1, n]$ and matrix $A \in \mathbb{R}^{m\times n}$, define*

$$\omega_\diamond(Q, s) \quad = \quad \min_{\boldsymbol{z}:\|\boldsymbol{z}\|_1/\|\boldsymbol{z}\|_\infty \le s} \|Q\boldsymbol{z}\|_\diamond / \|\boldsymbol{z}\|_\infty, \tag{13}$$

where Q is either A or A^TA.

978-1-4244-9848-2/11$26.00©2011 IEEE

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 1 *Under the assumption of Proposition 1, we have*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_\infty \leq \frac{2\epsilon}{\omega_\diamond(A, 2k)} \tag{14}$$

for the Basis Pursuit,

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_\infty \leq \frac{2\mu}{\omega_\infty(A^TA, 2k)} \tag{15}$$

for the Dantzig selector, and

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_\infty \leq \frac{(1+\kappa)\mu}{\omega_\infty(A^TA, 2k/(1-\kappa))} \tag{16}$$

for the LASSO estimator.

One of the primary contributions of this work is the design of algorithms that compute $\omega_\diamond(A, s)$ and $\omega_\infty(A^TA, s)$ 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 1. According to Corollary 1, the correct recovery of signal support is also guaranteed by reducing the ℓ_∞ norm to some threshold.

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 1, $\omega_\diamond(\cdot) > 0$ 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.

IV. 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^TA, and use $\|\cdot\|_\diamond$ to denote a general norm.

A. 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

$$s_* \;=\; \min_{\boldsymbol{z}} \|\boldsymbol{z}\|_1/\|\boldsymbol{z}\|_\infty \text{ s.t. } Q\boldsymbol{z} = 0. \tag{17}$$

Then, when $s < s_*$, we have $\omega_\diamond(Q, s) > 0$. We rewrite the optimization (17) as

$$\max_{\boldsymbol{z}} \|\boldsymbol{z}\|_\infty \text{ s.t. } Q\boldsymbol{z} = 0, \|\boldsymbol{z}\|_1 \leq 1, \tag{18}$$

which is solved using the following n linear programs:

$$\max_{\boldsymbol{z}} \boldsymbol{z}_i \text{ s.t. } Q\boldsymbol{z} = 0, \|\boldsymbol{z}\|_1 \leq 1. \tag{19}$$

Due to the equivalence of $A^TA\boldsymbol{z} = 0$ and $A\boldsymbol{z} = 0$, we always solve (18) for $Q = A$ and avoid $Q = A^TA$. 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.

B. 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:

$$\omega_\diamond(Q, s) = \min_{\boldsymbol{z}} \frac{\|Q\boldsymbol{z}\|_\diamond}{\|\boldsymbol{z}\|_\infty} \text{ s.t. } \frac{\|\boldsymbol{z}\|_1}{\|\boldsymbol{z}\|_\infty} \leq s, \tag{20}$$

or equivalently,

$$\frac{1}{\omega_\diamond(Q, s)} = \max_{\boldsymbol{z}} \|\boldsymbol{z}\|_\infty \text{ s.t. } \|Q\boldsymbol{z}\|_\diamond \leq 1, \frac{\|\boldsymbol{z}\|_1}{\|\boldsymbol{z}\|_\infty} \leq s. \tag{21}$$

We propose a very simple iterative procedure to solve (21). Suppose $\boldsymbol{z}_0$ is any initial feasible point (*e.g.*, the one obtained by solving (18) when $s < s_*$). Denote $t_0 = \|\boldsymbol{z}_0\|_\infty$. For $j = 0, 1, \cdots$, repeatedly solve the following convex optimization sub-problem:

$$t_{j+1} = \max_{\boldsymbol{z}} \|\boldsymbol{z}\|_\infty \text{ s.t. } \|Q\boldsymbol{z}\|_\diamond \leq 1, \|\boldsymbol{z}\|_1 \leq s \cdot t_j. \tag{22}$$

Note that, similar to (19), the sub-problem (22) can be solved by n convex problems. In particular, when $\diamond = 1$ or ∞, the sub-problem (22) can be solved by n linear programs, and when $\diamond = 2$, it can be solved by n second-order cone programs.

The following theorem shows that the simple procedure generates solutions converging to the global maximum of (20). Actually the sequence $\{t_k\}$ increases exponentially fast. More specifically, we have

Theorem 2 *If $s > 1$, then for any $\varepsilon > 0$ small enough, there exists $\gamma = \gamma(\varepsilon) > 1$ such that*

$$t_{j+1} \geq \gamma t_j \geq \gamma^{j+1} t_0 \tag{23}$$

as long as $t_j < (1-\varepsilon)t^*$, where t^* is the global optimal value of (20)*. As a consequence, the procedure* (22) *terminates in finite steps to achieve ε relative accuracy if $t^* < \infty$.*

In Figure 1, we illustrate the convergence behavior of the proposed algorithm (22). 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 $t_0 = \|\boldsymbol{x}_0\|_\infty$ with $\boldsymbol{x}_0 \in \mathcal{S}$, the algorithm solves

$$\max_{\boldsymbol{z}} \|\boldsymbol{z}\|_\infty \text{ s.t. } \|Q\boldsymbol{z}\|_\diamond \leq 1, \|\boldsymbol{z}\|_1 \leq s \cdot t_0 \tag{24}$$

in sub-figure 2. The solution is denoted by the black dot. Although the true domain for the optimization in (24) is the intersection of the distorted ℓ_∞ ball $\{\boldsymbol{z} : \|Q\boldsymbol{z}\|_\infty \leq 1\}$ and the ℓ_1 ball $\{\boldsymbol{z} : \|\boldsymbol{z}\|_1 \leq 1\}$, 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 (24) $t_1 = \|\boldsymbol{x}_1\|_\infty \geq t_0$. As a consequence, for the optimal solution $\boldsymbol{x}_1$, we have $\|\boldsymbol{x}_1\|_1/\|\boldsymbol{x}_1\|_\infty \leq \|\boldsymbol{x}_1\|_1/t_0 \leq s$. In the following sub-figures, at each iteration, we expand the ℓ_1 ball until we

978-1-4244-9848-2/11$26.00©2011 IEEE

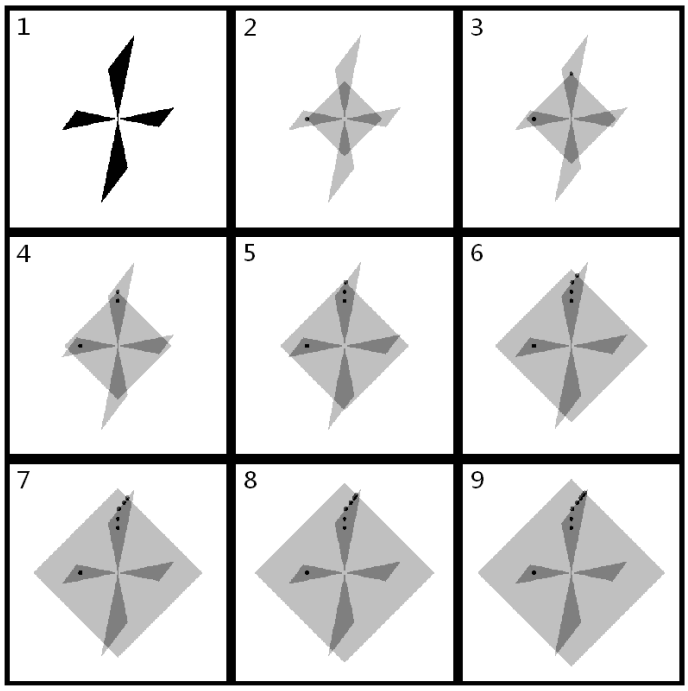

Fig. 1: Illustration of the convergence of procedure (22) when $\diamond = \infty$.

get to the tip point of the star-shaped region $\mathcal{S}$, which is the global optimum.

Note that in equations (20), (21), and (22), if we replace the ℓ_∞ norm with any other norm (with some other minor modifications), especially $\|\cdot\|_{s,1}$ or $\|\cdot\|_2$, then Theorem 2 is still valid. In addition, as we did in Corollary 1, 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 best performance bounds. Unfortunately, the major problem is that in these cases, the subproblems (22) do not admit an obvious polynomial time algorithm. It is very likely the corresponding norm maximization sub-problems (22) for $\|\cdot\|_{s,1}$ and $\|\cdot\|_2$ are NP hard [15].

Although Theorem 2 guarantees the procedure terminates in finite iterations, solving the subproblem (22) exactly at each iteration is still time consuming. We find that exactly solving the subproblem (22) at each iteration is actually not necessary. In the implementation, we use $(1+\eta)t_j$ as a minimal acceptable increase for the $(j+1)$th iteration, where $\eta > 0$ is a parameter. Once $(1+\eta)t_j$ is achieved, the algorithm exits from the loop of solving n convex programs. With this implementation, except for the last iteration and a few intermediate iterations to change the search directions, the procedure solves only *one* (instead of n) convex program at each iteration.

V. Probabilistic Behavior of $\omega_\diamond(Q,s)$

Using a theorem we established in [16] for ℓ_1-constrained minimal singular values (ℓ_1-CMSV), 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* if $\mathbb{E}|\langle \boldsymbol{X}, \boldsymbol{u}\rangle|^2 = \|\boldsymbol{u}\|_2^2$ and $\mathbb{P}(|\langle \boldsymbol{X}, \boldsymbol{u}\rangle| \geq t) \leq 2\exp(-ct^2/\|\boldsymbol{u}\|_2)$ hold for any $\boldsymbol{u} \in \mathbb{R}^n$. Here c is a numerical constant.

Theorem 3 *Let the rows of the sensing matrix A be* i.i.d. *subgaussian and isotropic random vectors. Then there exist constants c_1, c_2, c_3 such that for any $\epsilon > 0$ and $m \geq 1$ satisfying*

$$m \geq c_1 \frac{s^2 \log n}{\epsilon^2}, \tag{25}$$

we have

$$\mathbb{E}\ \omega_2(A,s) \geq 1 - c_2\epsilon, \tag{26}$$

$$\mathbb{P}\{\omega_2(A,s) \geq 1-\epsilon\} \geq 1 - \exp(-c_3\epsilon^2 m), \tag{27}$$

and

$$\mathbb{E}\ \omega_\infty(A^TA,s) \geq \frac{(1-c_2\epsilon)^2}{s}, \tag{28}$$

$$\mathbb{P}\left\{\omega_\infty(A,s) \geq \frac{(1-\epsilon)^2}{s}\right\} \geq 1 - \exp(-c_3\epsilon^2 m). \tag{29}$$

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 $2 \leq p \leq \infty$ [18]. In equations (28) and (29), the extra s in the lower bound of $\omega_\infty(A^TA,s)$ would contribute an s factor in the bounds of Theorem 1. 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 [8], [16].

The measurement bound (25) implies that the algorithms for verifying $\omega_\diamond > 0$ and for computing $\omega_\diamond$ work for s at least up to the order $\sqrt{m/\log n}$.

VI. Numerical Experiments

In this section, we numerically assess the performance of the algorithms for solving (17) and (20). Please refer to [17] for more extensive numerical experiment results.

The linear program (19) are implemented using the primal-dual algorithm outlined in Chapter 11 of [19]. Depending on whether $\diamond = 1, \infty$, or 2, (22) is solved using either n linear programs or n second-order cone programs. We use the log-barrier algorithm described in Chapter 11 of [19] to solve the second-order cone program. Interested readers are encouraged to refer to [20] 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.

The performance of the verification program (18) was previously reported in [16] in another context. For completeness, we reproduce some of the results in Table I. We present the calculated lower bounds for k^*, defined as the maximal sparsity level such that the sufficient and necessary condition (6) holds. We compare the lower bounds and the running times for our

978-1-4244-9848-2/11$26.00©2011 IEEE

implementation of (17), denoted as L_∞, and for the algorithm given in [13], denoted as JN (the authors' initials). Apparently, the lower bound on k^* computed by L_∞ is $k_* \stackrel{\text{def}}{=} \lfloor s^*/2 \rfloor$ with s^* given by (17). Table I is for matrices with a leading dimension $n = 1024$ and $m = \lfloor \rho n \rfloor, \rho = 0.1, 0.2, \ldots, 0.9$. We conclude that L_∞ and JN give comparable lower bounds, and our implementation of L_∞ performs much faster. In addition, it consumes less memory and is very stable.

TABLE I: Comparison of L_∞ and JN for a Gaussian matrix with leading dimension $n = 1024$.

m	lower bounds on k^*		CPU time (s)	
	L_∞	JN	L_∞	JN
102	2	2	136	457
204	4	4	281	1179
307	6	6	510	2235
409	7	7	793	3659
512	10	10	990	5348
614	12	12	1309	7156
716	15	15	1679	9446
819	20	21	2033	12435
921	29	32	2312	13564

In the next set of experiments, we assess the performance of our algorithms for computing $\omega_\infty(A^T A, s)$ and $\omega_2(A, s)$, respectively. We compare our recovery error bounds based on $\omega_\diamond$ with those based on the RIC. Combining Corollary 1 and Theorem 1, we have for the Basis Pursuit

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq \frac{2\sqrt{2k}}{\omega_2(A, 2k)}\epsilon, \tag{30}$$

and for the Dantzig selector

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq \frac{2\sqrt{2k}}{\omega_\infty(A^T A, 2k)}\mu. \tag{31}$$

For comparison, the two RIC bounds are

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq \frac{4\sqrt{1+\delta_{2k}(A)}}{1-(1+\sqrt{2})\delta_{2k}(A)}\epsilon, \tag{32}$$

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

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq \frac{4\sqrt{k}}{1-\delta_{2k}(A)-\delta_{3k}(A)}\mu, \tag{33}$$

for the Dantzig selector, assuming $\delta_{2k}(A) + \delta_{3k}(A) < 1$ [8]. Without loss of generality, we set $\epsilon = 1$ and $\mu = 1$.

The RIC is approximated using Monte Carlo simulations. 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 II, we compare the error bounds (30) and (32) for the Basis Pursuit algorithm. The corresponding s^* and k_* for different m are also included in the tables. Note the blanks mean that the corresponding bounds are not valid. For a Bernoulli matrix, 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 $k = 9$. Our bounds are also smaller than the RIC bounds.

TABLE II: 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 III, we compare the error bounds (31) and (33) for the Dantzig selector. For a Gaussian matrix, our bounds work for wider ranges of (k, m) pairs and are tighter.

TABLE III: 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 a Gaussian matrix.

	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							

In the last set of experiments, we compute $\omega_2(A, 2k)$ for a Gaussian matrix 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 2, the quantity $\omega_2(A, 2k)$ is plotted as a function of k for different m. We can clearly

978-1-4244-9848-2/11$26.00©2011 IEEE

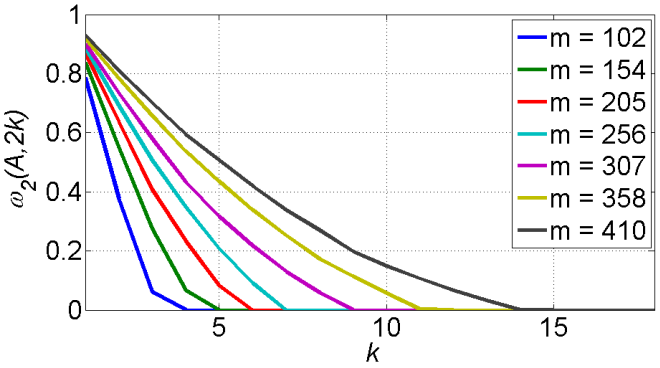

Fig. 2: $\omega_2(A, 2k)$ as a function of k with $n = 512$ and different m for a Gaussian matrix.

see how increasing the number of measurements (namely, the value of m) increases the goodness measure, and how increasing the sparsity level decreases the goodness measure.

In Figure 3, 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.

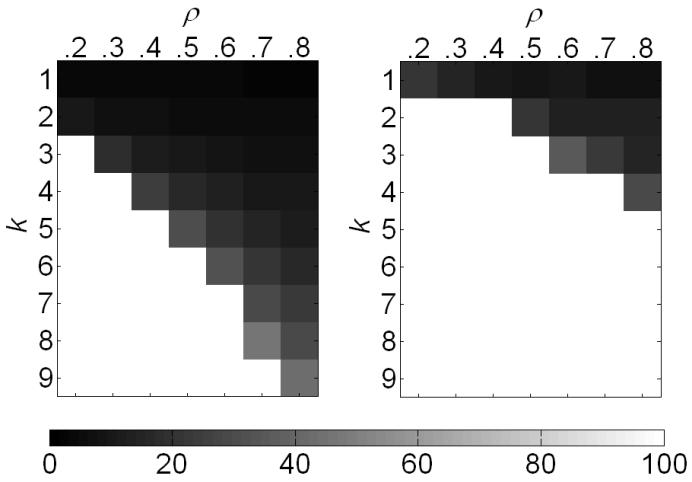

Fig. 3: $\omega_2(A, 2k)$ based bounds v.s. RIC based bounds on the ℓ_2 norms of the errors for a Gaussian matrix with leading dimension $n = 512$. Left: $\omega_2(A, 2k)$ based bounds; Right: RIC based bounds.

For $n = 256$, the algorithm generally takes 1 to 3 minutes to compute either $\omega_2(A, s)$ or $\omega_\infty(A^T A, s)$. The average time for $n = 512$ is around 15 minutes.

VII. 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. 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. 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.

References

[1] G. Tang and A. Nehorai, "Performance analysis for sparse support recovery," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 1383–1399, Mar. 2010.

[2] E. J. Candès, "The restricted isometry property and its implications for compressed sensing," *Compte Rendus de l'Academie des Sciences, Paris, Serie I*, vol. 346, 2008.

[3] S. Sen, G. Tang, and A. Nehorai, "Multi-objective optimized ofdm radar waveform for target detection in multipath scenarios," to appear in *IEEE Trans. Signal Processing*.

[4] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3010–3022, Aug. 2005.

[5] 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.

[6] M. Lustig, D. L Donoho, and J. M Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[7] S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comp.*, vol. 20, no. 1, pp. 33–61, 1998.

[8] E. J. Candès and T. Tao, "The Dantzig selector: Statistical estimation when p is much larger than n," *Ann. Statist.*, vol. 35, pp. 2313–2351, 2007.

[9] R. Tibshirani, "Regression shrinkage and selection via lasso," *J. Roy. Statist. Soc. Ser. B*, vol. 58, pp. 267–288.

[10] Y. Zhang, "A simple proof for recoverability of ℓ_1-minimization: go over or under?," Tech. Rep., Rice CAAM Department, 2005.

[11] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2845–2862, Nov. 2001.

[12] D. Donoho, "High-dimensional centrally-symmetric polytopes with neighborliness proportional to dimension," Technical report, Department of Statistics, Stanford University, 2004.

[13] A. Juditsky and A. S. Nemirovski, "On verifiable sufficient conditions for sparse signal recovery via ℓ_1 minimization," *ArXiv e-prints*, May 2010.

[14] A. d'Aspremont and L. El Ghaoui, "Testing the nullspace property using semidefinite programming," *ArXiv e-prints*, Nov. 2010.

[15] H. Bodlaender, P. Gritzmann, V. Klee, and J. Leeuwen, "Computational complexity of norm-maximization," *Combinatorica*, vol. 10, pp. 203–225, 1990.

[16] G. Tang and A. Nehorai, "Performance analysis of sparse recovery based on constrained minimal singular values," submitted for publication, *ArXiv e-prints* available.

[17] G. Tang and A. Nehorai, "Verifiable and computable ℓ_∞ performance evaluation of ℓ_1 sparse signal recovery," 2011.

[18] S. Mendelson, A. Pajor, and N. Tomczak-Jaegermann, "Reconstruction and subgaussian operators in asymptotic geometric analysis," *Geometric And Functional Analysis*, pp. 1248–1282, Nov. 2007.

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[20] E. Candés and J. Romberg, "ℓ_1-magic: Recovery of sparse signals via convex programming," 2005.

978-1-4244-9848-2/11$26.00©2011 IEEE