

Computation and Relaxation of Conditions for Equivalence between ℓ^1 and ℓ^0 Minimization*

Yoav Sharon

John Wright

Yi Ma[†]

April 11, 2008

Abstract

In this paper, we investigate the exact conditions under which the ℓ^1 and ℓ^0 minimizations arising in the context of sparse error correction or sparse signal reconstruction are equivalent. We present a much simplified condition for verifying equivalence, which leads to a provably correct algorithm that computes the exact sparsity of the error or the signal needed to ensure equivalence. Our algorithm is combinatorial in nature, but for moderate-sized matrices it produces the exact result in a reasonably short time. For ℓ^1 - ℓ^0 equivalence problems involving tall encoding matrices (highly robust error correction) and/or wide overcomplete dictionaries (sparse signal reconstruction from few measurements), our algorithm is exponentially faster than the only other algorithm known for this problem. We present an example application that requires such matrices, for which our algorithm can greatly assist with real system design. We also show how, if the encoding matrix is imbalanced, an optimal diagonal rescaling matrix can be computed by linear programming, so that the rescaled system enjoys the widest possible equivalence.

1 Introduction

Recently, there has been an explosion of research in sparse representation, centering around two mutually complementary mathematical problems:

1. **Sparse Signal Reconstruction:** Finding *sparse* solutions to underdetermined systems of linear equations [1, 2, 3].
2. **Sparse Error Correction:** Finding robust solutions to overdetermined systems of linear equations, subject to arbitrary, but *sparse* errors [4, 5].

The conceptual appeal of sparsity is clear: representations involving only a few basis elements are far more amenable to human interpretation than arbitrary linear combinations. Sparsity is also the natural goal of data compression: representations involving only a few basis elements are clearly more compact, and may be encoded using fewer bits. Sparsity also arises in the context of error correcting codes, where the error, rather than the signal, is assumed to be sparse [4]. However, the applications of sparse representations range far beyond such traditional problems in signal processing and information theory. For example, the problem of recognizing human faces from images can be cast as a sparse representation problem in which the pattern of sparsity encodes the identity of the subject [6]. Recognition subject to occlusion can be viewed as a special robust

*This work is partially supported by the following grants: NSF EHS-0509151, NSF CCF-0514955 and NSF ECCS-0701676

[†]All authors are with the Coordinated Science Laboratory, Department of Electrical and Computer Engineering, University of Illinois, Urbana-Champaign, IL, 61801 USA. Email: {ysharon2, jnwright, yima}@uiuc.edu.

decoding problem, in which the sparse errors are concentrated on a fraction of occluded pixels. Further applications exist in image denoising and signal reconstruction [7], as well as in robust state estimation subject to measurement errors.

Much of the promised practical impact of sparse representation is due to a very recently discovered phenomenon, sometimes referred to as ℓ^1 - ℓ^0 equivalence [3]. Equivalence results roughly state that if the representation to be computed (or the error to be corrected) is *sufficiently sparse*, then the NP-hard problem of finding the sparsest linear representation [8] can be solved efficiently and exactly via linear programming, by minimizing an appropriate ℓ^1 -norm. A great deal of effort has been invested into determining, in terms of the properties of the basis or the encoding matrix, how sparse the desired representation must be for equivalence to hold (e.g. [2, 3, 4, 5], amongst others). For example, [9] introduces the so-called *Uniform Uncertainty Principle* for overcomplete bases, which holds if all subsets of bases are approximately orthonormal. Similar hypotheses¹ have proven extremely fruitful in analyzing ℓ^1 - ℓ^0 equivalence in random matrix ensembles. One such result states that in overcomplete bases generated at random from a Gaussian distribution, asymptotically, with overwhelming probability, ℓ^1 -minimization recovers all sparse representations whose fraction of nonzero elements is less than a fixed constant [4].

Despite these successes, the conditions given in the literature are generally sufficient but not necessary², giving very pessimistic indications of the ability of ℓ^1 -minimization to recover sparse representations. Simulations (in e.g. [4]) demonstrate a surprisingly large gap between theory and experiment: ℓ^1 -minimization significantly outperforms expectations. Sharp theoretical results are mostly asymptotic and probabilistic in nature. For example, [10] gives a precise characterization of the equivalence properties of very large Gaussian matrices.

In contrast, this paper considers the problem of guaranteeing ℓ^1 - ℓ^0 equivalence in a given linear system of finite size, whose structure may be dictated by the problem at hand. Our algorithm answers the following question, which is of the utmost importance in designing and analyzing practical systems based on ℓ^1 - ℓ^0 equivalence:

In a given overdetermined linear system of equations, how many arbitrary errors can we guarantee to correct by ℓ^1 -minimization? Equivalently, in an underdetermined linear system, how sparse must a solution be for ℓ^1 -minimization to guarantee to recover it?

Knowing the answer to these questions is essential to analyzing the operating range of the system. Moreover, given several proposed linear bases or encoding matrices, knowing their exact ℓ^1 - ℓ^0 equivalence properties would allow the engineer to choose the one that maximizes the operating range of ℓ^1 -minimization.

It was recently proven that when the signals are constrained to be positive, the above problem is NP-hard [11] and also hard to approximate within a constant factor. As this is a very similar and presumably simpler problem than the one addressed here (where the signals may have arbitrary signs), we believe that our problem is also NP-hard.

Our algorithm is indeed combinatorial. It is also not the first algorithm to address this problem as another algorithm can be easily derived from the results appearing in [3] and in [12, §II]. We will discuss this alternative approach, the only published alternative we are aware of, in more detail at the end of §2. To find the exact answer to the above question, both algorithms need to scan a set of combinatorial size, and apply some verification process on each element in the set. However, we will show that the size of the set that our algorithm needs to scan is *exponentially smaller* than that

¹Stated, for example, in terms of the *Restricted Isometry Constants* [4].

²One noteworthy exception is the necessary and sufficient condition given in [3] in terms of polytope neighborliness, but even there no algorithm is proposed for verifying equivalence.

of the alternative algorithm precisely in circumstances that arguably are the most important and common in practice: in highly robust error correction problems where the encoding matrix is much taller than it is wide; and in sparse signal reconstruction from few measurements where the sensing matrix is much wider than it is tall. Furthermore, our verification process only requires solving a small set of linear equations and then applying a sorting operation, whereas the alternative requires solving a large linear program³. Thus for small matrices, for which our algorithm requires only a few seconds to produce the answer (on a PC), the alternative algorithm will require days or even months. For an engineer designing a solution to an application involving matrices of this size, this difference in running times is crucial. In §5 we give an example of such an application, in robust state estimation from corrupted GPS measurements.

Contributions of this paper. The main contribution of this paper is a simple, novel algorithm for determining when ℓ^1 - ℓ^0 equivalence holds in a given linear system of equations. We prove the algorithm's correctness and analyze its complexity. In situations where the given encoding matrix is imbalanced, we show how an optimal diagonal rescaling matrix can be computed via linear programming, so that the rescaled system enjoys the widest possible equivalence.

Organization of this paper. The paper is organized as follows. §2 formally introduces the sparse error correction and representation problems and discusses their relationship. §3 describes our algorithm for verifying ℓ^1 - ℓ^0 equivalence, and proves its correctness. §4 presents a novel algorithm that optimally rescales a given linear system to maximize equivalence, and proves its correctness. §5 presents an application involving matrices that only our algorithm can analyze in reasonable time. Concluding remarks and open questions for future research are given in §6.

2 ℓ^1 - ℓ^0 equivalence for error correction and sparse representation

Two dual ℓ^0 minimization problems. Consider the following ℓ^0 -norm⁴ minimization problems:

1. **Sparse Error Correction.** Given $\mathbf{y} \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$ ($m > n$),

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \|\mathbf{y} - A\mathbf{x}\|_0. \quad (1)$$

2. **Sparse Signal Reconstruction.** Given $\mathbf{z} \in \mathbb{R}^p$, $B \in \mathbb{R}^{p \times m}$ ($p < m$),

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \|\mathbf{w}\|_0 \quad \text{subject to} \quad \mathbf{z} = B\mathbf{w}. \quad (2)$$

We will assume that these problems have a unique solution. This will be generally be true if there exists \mathbf{x}_0 such that the error vector, $\mathbf{e} = \mathbf{y} - A\mathbf{x}_0$, is sparse enough, or if there exists \mathbf{w}_0 sparse enough such that $\mathbf{z} = B\mathbf{w}_0$. For example, if any set of $2T$ columns of B are linearly independent, then any sparse representation $\mathbf{z} = B\mathbf{w}_0$ with $\|\mathbf{w}_0\| \leq T$ is the unique solution to (2) [2].

The above two problems are mutually *dual*, or complementary, in the sense that one can convert one problem to the other. It has been shown in [4] that the decoding problem (1) can be converted to the sparse representation problem (2). To see how to convert problem (2) to (1), let $n = m - \text{rank}(B)$ and let A be a full-rank $m \times n$ matrix whose columns span the kernel of B : $BA = 0$. Now find any \mathbf{y} so that $\mathbf{z} = B\mathbf{y}$ and define $f(\mathbf{x}) \doteq \|\mathbf{y} - A\mathbf{x}\|$. In this notation (regardless of which norm is used),

$$\arg \min_{\mathbf{w}: \mathbf{z} = B\mathbf{w}} \|\mathbf{w}\| = f\left(\arg \min_{\mathbf{x}} \|\mathbf{y} - A\mathbf{x}\|\right). \quad (3)$$

³A linear programming (LP) solver is required if one needs an exact solution using the alternative algorithm. An alternative approach for the verification process is given in [13]. However, with the approach in [13] there is still need to scan to exponentially larger set; the approach relies on an iterative method, each iteration solving a large set of linear equations; and most importantly, there is no guarantee that the iteration will converge to the true solution.

⁴Here, $\|\mathbf{w}\|_0$ counts the number of nonzero entries in \mathbf{w} .

Equivalence between ℓ^0 and ℓ^1 minimization. Problems (1) and (2) are NP-hard in general [8]. Nevertheless, as shown in [2, 4], if the error \mathbf{e} or the solution \mathbf{w} is sufficiently sparse, the solutions to (1) and (2) are *the same as* the solutions to:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \|\mathbf{y} - A\mathbf{x}\|_1, \quad (4)$$

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \|\mathbf{w}\|_1 \quad \text{subject to} \quad \mathbf{z} = B\mathbf{w}, \quad (5)$$

respectively. Problems (4) and (5) can be efficiently solved via linear programming. In addition to this dramatic decrease in computational cost, these problems can also be shown to be robust to small measurement errors [2, 5].

In this paper, we are interested in determining *for any given matrix A (or B), exactly how sparse the error \mathbf{e} (or the solution \mathbf{w}) must be in order for the above ℓ^1 minimizations to solve problems (1) and (2)*. We therefore define the following threshold for sparse error correction:

$$T^*(A) \doteq \max_{T \in \mathbb{Z}} T \text{ such that } \|\mathbf{y} - A\mathbf{x}_0\|_0 \leq T \Rightarrow \mathbf{x}_0 = \arg \min_{\mathbf{x}} \|\mathbf{y} - A\mathbf{x}\|_1. \quad (6)$$

$T^*(A)$ is the largest number of arbitrary errors in the observation \mathbf{y} that can be corrected by ℓ^1 -minimization. This quantity is closely related to the *Equivalence Breakdown Point* introduced by Donoho [2, 3] for the sparse signal reconstruction problem (2):

$$\text{EBP}(B) \doteq \max_{k \in \mathbb{Z}} k \text{ such that } \|\mathbf{w}_0\|_0 \leq k \Rightarrow \mathbf{w}_0 = \arg \min_{\mathbf{w}: B\mathbf{w}=B\mathbf{w}_0} \|\mathbf{w}\|_1. \quad (7)$$

This is the maximum number of arbitrary nonzeros that can be uniquely recovered by the ℓ^1 -minimization.

For the rest of the paper, we consider only the error correction problem (4). However, the duality between error correction and signal reconstruction implies that if A is a full-rank matrix spanning the kernel of B , $\text{EBP}(B) = T^*(A)$. Thus, the proposed algorithms can be applied to (2) as well.

Previous algorithm for verifying ℓ^1 - ℓ^0 equivalence. The following result was proven in [3] and in [12, §II]:

Theorem 1 *If for some $\mathbf{v} \in \mathbb{R}^n$ we have*

$$\mathbf{v} = \arg \min_{\mathbf{w}} \|\mathbf{w}\|_1 \quad \text{subject to} \quad B\mathbf{v} = B\mathbf{w}, \quad (8)$$

then for all \mathbf{v}' such that $\text{sign}(v'_i) = \text{sign}(v_i)$, $i = 1 \dots n$ we will have

$$\mathbf{v}' = \arg \min_{\mathbf{w}} \|\mathbf{w}\|_1 \quad \text{subject to} \quad B\mathbf{v}' = B\mathbf{w}.$$

In words, this result states that to determine whether ℓ^1 minimization will recover a given signal, one only needs to know the signs of that signal. Thus, $\text{EBP}(B)$ is the maximal s for which $\forall I \subset \{1, \dots, m\}$ of size s , and $\forall \mathbf{w} \in \mathbb{R}^m$ with $|w_i| = 1 \ \forall i \in I$ and $w_i = 0 \ \forall i \notin I$, (8) holds.

This condition can be verified by solving the linear program (8), or by searching for a subgradient to witness the optimality of \mathbf{v} (see [12, §II] for more information on this approach). The second approach, however, also requires solving a linear programming feasibility problem. For each s , (8) needs to be checked on a set of size $2^s \binom{m}{s}$. Thus, the computation time of this algorithm is:

$$\sum_{s=1}^{T^*} 2^s \binom{m}{s} t_{LP}(m), \quad (9)$$

where t_{LP} is the time required to solve a linear program with m constraints. After presenting our Algorithm in §3, we will compare the complexity and running times of the two approaches.

3 Algorithm for verifying ℓ^1 - ℓ^0 equivalence

In this section, we derive an algorithm for computing $T^*(A)$, and hence precisely verifying ℓ^1 - ℓ^0 equivalence. We will need the following definition and propositions:

Definition 2 The “ d -skeleton” is defined to be the collection of all the d -dimensional faces of the standard ℓ^1 -ball $B_1 \doteq \{\mathbf{v} \in \mathbb{R}^m : \|\mathbf{v}\|_1 \leq 1\}$. In particular the 0-skeleton is all the vertices, the 1-skeleton is all the edges, and so on. We will denote it as $SK_d(B_1)$:

$$SK_d(B_1) \doteq \{\mathbf{v} \in \mathbb{R}^m : \|\mathbf{v}\|_1 = 1, \|\mathbf{v}\|_0 \leq d + 1\}.$$

Proposition 3 For every $\mathbf{x}_0 \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ the following implication holds

$$\|\mathbf{y} - A\mathbf{x}_0\|_0 \leq T \quad \Rightarrow \quad \mathbf{x}_0 = \arg \min_{\mathbf{x}} \|\mathbf{y} - A\mathbf{x}\|_1 \quad (10)$$

if and only if

$$\forall \mathbf{v} \in SK_{T-1}(B_1), \quad \forall \mathbf{z} \in \mathbb{R}^n \setminus 0, \quad \|\mathbf{v} + A\mathbf{z}\|_1 > 1. \quad (11)$$

Note that this proposition is true regardless of the assumption that for any \mathbf{y} and \mathbf{x}_0 such that $\|\mathbf{y} - A\mathbf{x}_0\|_0 \leq T$, (1) will have a unique solution. Naturally, if the assumption does not hold, then neither do (10) and (11) hold.

proof: Assume that for some \mathbf{x}_0 and \mathbf{y} with $\|\mathbf{y} - A\mathbf{x}_0\|_0 \leq T$, (10) does not hold. Then $\exists \mathbf{x} \neq \mathbf{x}_0$ such that $\|\mathbf{y} - A\mathbf{x}\|_1 \leq \|\mathbf{y} - A\mathbf{x}_0\|_1 \doteq c$. Choose $\mathbf{v} = \frac{1}{c}(\mathbf{y} - A\mathbf{x}_0)$. Since $\|\mathbf{v}\|_0 \leq T$ and $\|\mathbf{v}\|_1 = 1$, $\mathbf{v} \in SK_{T-1}(B_1)$. Choose $\mathbf{z} = \frac{1}{c}(\mathbf{x}_0 - \mathbf{x}) \neq 0$, then $\|\mathbf{v} + A\mathbf{z}\|_1 = \frac{1}{c}\|\mathbf{y} - A\mathbf{x}\|_1 \leq 1$. Thus (11) does not hold and this proves the sufficiency. Assume now that (11) does not hold, so there exist $\mathbf{v} \in SK_{T-1}(B_1)$ and $\mathbf{z} \in \mathbb{R}^n \setminus 0$ such that $\|\mathbf{v} + A\mathbf{z}\|_1 \leq 1$. Choose \mathbf{x}_0 arbitrarily and set $\mathbf{y} = A\mathbf{x}_0 + \mathbf{v}$. Then $\|\mathbf{y} - A\mathbf{x}_0\|_0 \leq T$, and there exists $\mathbf{x} = \mathbf{x}_0 - \mathbf{z} \neq \mathbf{x}_0$ such that $\|\mathbf{y} - A\mathbf{x}\|_1 = \|\mathbf{v} + A\mathbf{z}\|_1 \leq 1 = \|\mathbf{y} - A\mathbf{x}_0\|_1$. Thus (10) does not hold. This proves the necessity. \square

Note that (10) is exactly the condition we want to verify. Figure 1 left illustrates the relation between (10) and (11), and gives the geometric intuition as to why they are equivalent. Although condition (11) is more geometrically intuitive than condition (10), it is still difficult to verify computationally. Without knowing $T^*(A)$, one needs to check starting from $T = 1, 2, \dots$ until the condition (11) eventually fails. Moreover, condition (11) requires checking that at every point on the d -skeleton, the subspace spanned by A does not penetrate the ℓ^1 ball. This leads to the main result of this paper, an equivalent condition that does not require search over \mathbf{v} , and only involves checking a finite set of points in $\text{span}(A)$.

Proposition 4 Let $A \in \mathbb{R}^{m \times n}$ and $d \in \mathbb{N} \cup 0$ be given and assume the rows of A are in general directions, i.e. any n rows of A are independent⁵. The following holds:

$$\forall \mathbf{v} \in SK_d(B_1), \quad \forall \mathbf{z} \in \mathbb{R}^n \setminus 0 \quad \|\mathbf{v} + A\mathbf{z}\|_1 > 1 \quad (12)$$

if and only if for all subsets $I \subset M \doteq \{1, \dots, m\}$ containing $n - 1$ indices, all subsets $J \subset M \setminus I$ containing $T = d + 1$ indices, and for some $\mathbf{y} \in \mathbb{R}^m$ such that

$$\mathbf{y} \in \text{span}(A) \setminus 0, \quad \forall i \in I \quad y_i = 0, \quad (13)$$

the following holds:

$$\sum_{j \in J} |y_j| < \sum_{j \in M \setminus J} |y_j|. \quad (14)$$

⁵If A is not in general position, the approach can still be applied, by replacing the set of \mathbf{y} in (13) with the directions of the one-dimensional edges of the polyhedral cones given by the intersections of $\text{span}(A)$ with the orthants of \mathbb{R}^m . See the discussions after Theorem 5 for a further geometric interpretation of our results.

proof: First note that due to the assumption that the rows of A are in general directions, for any $I \subset M$ containing $n-1$ indices, (13) defines \mathbf{y} uniquely upto scale ($n-1$ independent equations in n variables). As (14) is invariant to (nonzero) scaling in \mathbf{y} , it holds for some \mathbf{y} satisfying (13) if and only if it holds for all such \mathbf{y} .

We first prove the sufficiency – the “if” direction. Assume (12) does not hold, then there exists $\mathbf{v} \in \text{SK}_d(B_1)$ and $\mathbf{z} \in \mathbb{R}^n \setminus 0$ such that $\|\mathbf{v} + A\mathbf{z}\|_1 \leq 1$. Let A_{i*} denote the i -th row of A . Let $P_z \subset M$ be the subset of indices for which $v_i + A_{i*}\mathbf{z}$ does not vanish. Let $P_v \subset M$ be the subset of indices for which v_i does not vanish: $|P_v| \leq d+1$. Note that $1 \geq \|\mathbf{v} + A\mathbf{z}\|_1 \geq \sum_{i \in P_v} |v_i| - \sum_{i \in P_v} |A_{i*}\mathbf{z}| + \sum_{i \in M \setminus P_v} |A_{i*}\mathbf{z}|$. As $\mathbf{v} \in \text{SK}_d(B_1) \Rightarrow \|\mathbf{v}\|_1 = \sum_{i \in P_v} |v_i| = 1$ we have $\sum_{i \in P_v} |A_{i*}\mathbf{z}| \geq \sum_{i \in M \setminus P_v} |A_{i*}\mathbf{z}|$. We will show that there exists $\tilde{\mathbf{z}} = \mathbf{z} + \mathbf{x}$ such that at least $n-1$ of $A_{i*}\tilde{\mathbf{z}}$, $i \in M \setminus P_v$ vanish and

$$\sum_{i \in P_v} |A_{i*}\tilde{\mathbf{z}}| \geq \sum_{i \in P_v} |A_{i*}\mathbf{z}|, \quad \sum_{i \in M \setminus P_v} |A_{i*}\tilde{\mathbf{z}}| \leq \sum_{i \in M \setminus P_v} |A_{i*}\mathbf{z}|. \quad (15)$$

If such $\tilde{\mathbf{z}}$ exists, then (14) will not hold for $I = \{i \in M \setminus P_v \mid A_{i*}\tilde{\mathbf{z}} = 0\}$, $\mathbf{y} = A\tilde{\mathbf{z}}$ and $J = P_v$.

If $|M \setminus (P_z \cup P_v)| \geq n-1$, then we can take $\tilde{\mathbf{z}} = \mathbf{z}$ and we are done. Otherwise define $w_i = \text{sign}(A_{i*}\mathbf{z})$, and consider the set of $|M \setminus (P_z \cup P_v)| + 1$ equations in \mathbf{x} :

$$\sum_{i \in P_v} w_i A_{i*}\mathbf{x} = 0, \quad \text{and} \quad A_{j*}\mathbf{x} = 0 \quad \forall j \in M \setminus (P_z \cup P_v). \quad (16)$$

Since the number of equations is less than n , there is at least a one dimensional subspace $\{\alpha\mathbf{x} : \alpha \in \mathbb{R}\}$ of solutions. For any solution \mathbf{x} , the first inequality of (15) holds with $\tilde{\mathbf{z}} = \mathbf{z} + \mathbf{x}$, since:

$$\sum_{i \in P_v} |A_{i*}(\mathbf{z} + \mathbf{x})| \geq \sum_{i \in P_v} w_i A_{i*}(\mathbf{z} + \mathbf{x}) = \sum_{i \in P_v} |A_{i*}\mathbf{z}| + 0.$$

Now choose α small enough so that

$$-w_i \alpha A_{i*}\mathbf{x} \leq w_i A_{i*}\mathbf{z} \quad \forall i \in P_z \setminus P_v \quad (17)$$

(note that $w_i A_{i*}\mathbf{z} > 0 \quad \forall i \in P_z \setminus P_v$). We then have $\sum_{i \in M \setminus P_v} |A_{i*}(\mathbf{z} + \alpha\mathbf{x})| = \sum_{i \in P_z \setminus P_v} w_i A_{i*}\mathbf{z} + \alpha \sum_{i \in P_z \setminus P_v} w_i A_{i*}\mathbf{x}$, where we used (16) and the definition of P_z to restrict the summation to be over $P_z \setminus P_v$ instead of over $M \setminus P_v$. By taking the sign of α to be opposite to that of $\sum_{i \in P_z \setminus P_v} w_i A_{i*}\mathbf{x}$, then as long as (17) holds the second inequality of (15) also holds. If we choose $|\alpha|$ such that

$$|\alpha| = \min_{i: \text{sign}(\alpha A_{i*}\mathbf{x}) \neq \text{sign}(A_{i*}\mathbf{z})} \frac{|A_{i*}\mathbf{z}|}{|A_{i*}\mathbf{x}|}, \quad (18)$$

then (17) still holds, but we also have that for some $j \in P_z \setminus P_v$, $A_{j*}\tilde{\mathbf{z}} = A_{j*}(\mathbf{z} + \alpha\mathbf{x}) = 0$. If now at least $n-1$ of $A_{i*}\tilde{\mathbf{z}}$, $i \in M \setminus P_v$ vanish then there exists $I \subset M \setminus P_v$ such that $A_{i*}\tilde{\mathbf{z}} = 0 \quad \forall i \in I$ and with $\mathbf{y} = A\tilde{\mathbf{z}}$ and $J = P_v$ (14) does not hold. If less than $n-1$ of $A_{i*}\tilde{\mathbf{z}}$, $i \in M \setminus P_v$ vanish, then we can replace \mathbf{z} with $\tilde{\mathbf{z}}$, redefine P_z , and repeat the process. Since at each iteration $P_z \setminus P_v$ decreases by at least 1, we are guaranteed to eventually find $\tilde{\mathbf{z}}$ for which $n-1$ of $A_{i*}\tilde{\mathbf{z}}$, $i \in M \setminus P_v$ do vanish. This proves sufficiency.

For the necessity – the “only if” direction, assume there exists $I \subset M$, $|I| = n-1$, $J \subset M \setminus I$, $|J| = d+1$, and some \mathbf{y} for which (13) holds but (14) does not. Set $c = \sum_{j \in J} |y_j|$ and \mathbf{v} such that $\forall i \in J$, $v_i = \frac{1}{c} y_i$ and $\forall i \in M \setminus J$, $v_i = 0$. Let \mathbf{x} such that $\mathbf{y} = A\mathbf{x}$, and set $\mathbf{z} = -\frac{1}{c}\mathbf{x}$. We now have that $\mathbf{v} \in \text{SK}_d(B_1)$ and $\mathbf{z} \in \mathbb{R}^n \setminus 0$. Furthermore,

$$\|\mathbf{v} + A\mathbf{z}\|_1 = \frac{1}{c} \sum_{j \in M \setminus J} |A_{j*}\mathbf{x}| = \frac{\sum_{j \in M \setminus J} |y_j|}{\sum_{j \in J} |y_j|} \leq 1,$$

where we used the assumption that (14) does not hold for the last inequality. Thus (12) does not hold and this proves the necessity. \square

The following algorithm uses Proposition 4 to compute the maximum T for which (10) holds:

Algorithm 1 Computing $T^*(A)$

Input: $A \in \mathbb{R}^{m \times n}$.

- 1: Set $T \leftarrow m$ and let I_1, \dots, I_N , $N = \binom{m}{n-1}$, be all the subsets of $M \doteq \{1, \dots, m\}$ containing $n-1$ indices.
- 2: **for** $k = 1 : N$ **do**
- 3: Find a nontrivial solution $\mathbf{x} \in \mathbb{R}^n$ to $A_{i*}\mathbf{x} = 0 \quad \forall i \in I_k$.
- 4: Set $\mathbf{y} \leftarrow A\mathbf{x}$ and reorder the elements of \mathbf{y} such that $|y_{r_1}| \geq |y_{r_2}| \geq \dots \geq |y_{r_m}|$.
- 5: Find the largest integer, s , such that

$$\sum_{i=1}^s |y_{r_i}| < \sum_{i=s+1}^m |y_{r_i}|. \quad (19)$$

- 6: Set $T \leftarrow \min\{T, s\}$.

7: **end for**

Output: T .

Theorem 5 *Algorithm 1 returns the maximum $T^*(A)$ for which (10) holds.*

proof: Note that due to the sorting, (19) holds for some integer s if and only if (14) holds for all subsets $J \in M \setminus I_k$ containing s elements. Proposition 3, Proposition 4, and the fact that the algorithm returns the minimum integer s for which (19) holds over all subsets $I \subset M$, $|I| = n-1$, and all subsets $J \subset M \setminus I$, $|J| = s$ proves the theorem. \square

Geometric interpretation. Although our proof for Algorithm 1 is mostly algebraic in nature, our results have a strong geometric interpretation. There are 2^n possible sign patterns $\sigma \in \{\pm 1\}^m$ within $\text{span}(A)$. The set of all $\mathbf{w} \in \text{span}(A)$ with a given sign pattern σ is a polyhedral cone $C_\sigma = \{\mathbf{w} : \sigma_i w_i \geq 0 \forall i\}$ whose edges are contained in the one-dimensional intersections of $\text{span}(A)$ with subspaces $S_I = \{\mathbf{w} \in \mathbb{R}^m : w_i = 0 \forall i \in I\}$ spanned by subsets of $n-m+1$ of the Euclidean basis vectors. These one-dimensional intersections are spanned by the vectors $\mathbf{y}(I) \in \text{span}(A) \cap S_I$ in equation (13).

Proposition 3 states that $T(A) \geq s$ iff every translate $\mathbf{v} + \text{span}(A)$ of $\text{span}(A)$ to a s -sparse point \mathbf{v} on the skeleton of the ℓ^1 -ball does not penetrate the ball. Proposition 4 states that, rather than checking that every point in $\mathbf{v} + \text{span}(A)$ does not fall inside B_1 , we need only check a finite set of points $\{\mathbf{v} + \mathbf{y}(I)\}$ given by the $\mathbf{y}(I)$ that generate the edges of the cones C_σ . Furthermore, for a given $\mathbf{y}(I)$, the sorting step of Algorithm 1 identifies the largest s such that for any s -sparse \mathbf{v} , $\mathbf{v} + \mathbf{y}(I)$ does not penetrate the ℓ^1 ball – effectively checking exponentially (in s) many \mathbf{v} simultaneously.

The reason that this finite set of edge directions $\{\mathbf{y}(I)\}$ suffices is that, for a given \mathbf{v} and sign pattern σ , the change in ℓ^1 norm due to a small perturbation \mathbf{y} , $\|\mathbf{v} + \mathbf{y}\|_1 - \|\mathbf{v}\|_1$, is a linear function of \mathbf{y} . Penetration occurs iff one of these functions is negative for some \mathbf{v} . But a linear (and hence concave) functional on the convex cone C_σ is minimized along the boundary, so if it is negative anywhere it is negative on at least one of the edges. So, if any $\mathbf{v} + A\mathbf{x}$ lies inside the ball, $\mathbf{v} + \mathbf{y}(I)$ lies inside for some I .

Complexity and comparison to previous algorithm. The computation time of Algorithm 1 is

$$\binom{m}{n-1} (t_{ls}(n-1) + t_{mv}(m) + t_{sort}(m)), \quad (20)$$

where t_{ls} , t_{mv} and t_{sort} are the times it takes to solve a linear system of equations; to multiply a matrix with a vector; and to sort, respectively. Although both this Algorithm 1 and the algorithm discussed at the end of §2 have exponential complexity, the difference between the two can be quite large. For example, if m, n and $T^*(A)$ grow proportionally with $n = \lfloor \delta m \rfloor$ and $T^* = \lfloor \rho m \rfloor$, the ratio of the computation times of the two algorithms is

$$\frac{t_{\text{Algorithm 1}}}{t_{\text{Algorithm of §2}}} = 2^{m(H(\delta) - \rho - H(\rho) + o(1))}, \quad (21)$$

where $H(x) \doteq -x \log_2 x - (1-x) \log_2 (1-x)$ is the binary entropy function. In this asymptotic setting, Algorithm 1 has exponentially lower complexity than the sign-enumeration approach of §2 when $H(n/m) < T^*/m + H(T^*/m)$. Our algorithm therefore sees its greatest advantage in two scenarios: when n/m is small compared to T^*/m (very tall A), or when n/m is close to one (almost square A).⁶ The first case is of interest for highly robust error correction – here, the number of measurements $A\mathbf{x}$ of the signal \mathbf{x} is quite large, as is the number of errors that can be corrected [12]. In the second case, A is nearly square and any corresponding full-rank B such that $BA = 0$ is very wide. The EBP of such wide B is of interest for finding sparse solutions to highly underdetermined linear systems, and for reconstructing sparse signals from fewest possible measurements [15].

While both algorithms are intractable for large m , the difference can be quite pronounced even for small matrices. For example, for a randomly generated $A \in \mathbb{R}^{40 \times 5}$ we found⁷ using Algorithm 1 that $T^*(A) = 7$ during $\binom{40}{4} 0.2_{ms} = 91,390 \times 0.2_{\text{milliseconds}} = 17_{\text{seconds}}$. To verify (8), with $B \in \mathbb{R}^{35 \times 40}$ such that $BA = 0$, for a single \mathbf{v} with support size $s = 7$ required $25_{\text{milliseconds}}$. Thus we expect that verifying that we can recover signals with support of size 7 using the alternative algorithm will take us $18,643,560 \times 25_{\text{milliseconds}} \approx 1_{\text{year}}$. While most of the literature on ℓ^1 -minimization as a surrogate for ℓ^0 -minimization focuses on the large-matrix limit, where neither algorithms can produce the exact solution, in §5 we show one practical example where ℓ^1 -minimization even with small matrices can be of great benefit. Moreover, when the computational complexity exceeds available resources, a good upper bound for $T(A)$ can still be obtained by randomly sampling supports I_k .

Example 6 (Sparse recovery with random dictionaries) *We apply Algorithm 1 to test the breakdown point of ℓ^1 - ℓ^0 equivalence for matrices B sampled from several random matrix ensembles. The ensembles considered are Gaussian (entries iid normal), Partial DCT (uniformly chosen rows of the DCT matrix), Partial Hadamard (uniformly chosen rows of a Hadamard matrix) [14], and Uniform Sphere (columns sampled uniformly from the sphere). In each case, we generate a $5n \times 6n$ matrix B (the columns of which can be viewed as an overcomplete dictionary for signal representation) and calculate $\text{EBP}(B) = T^*(B^\perp)$. We repeat 1,000 times for each $n = 1, \dots, 5$. Figure 3 plots the histograms for each of these ensembles. Notice that when the Hadamard matrix exists⁸, it exhibits stronger equivalence guarantees than the others. Interestingly, the Partial DCT matrix outperforms Gaussian and Spherical, despite asymptotics that are worse by a log factor [15].*

⁶Comparing the exponent in (21) to numerical results of [10], for Gaussian matrices A , our algorithm's complexity is exponentially smaller than that of the competitor outlined in §2 whenever $n/m < .175$, or $n/m > .95$. Figure 2 gives a graphical explanation of this observation.

⁷We ran both algorithms on an Intel T7200 @ 2.00GHz system with 2GB RAM.

⁸Hadamard matrices do not exist for arbitrary n , so this ensemble is only tested in the 10×12 and 20×24 cases.

4 Computing scaling parameters for higher robustness

We motivate our next problem through an example:

Example 7 Consider the case of $m = 3$, $n = 1$, $A = [1, 0.1, 0.1]^T$. If we translate $\text{span}(A)$ to the vertex $\mathbf{v} = [1, 0, 0]^T$ of the unit ℓ^1 -ball B_1 , it will penetrate B_1 . Thus $T^*(A) = 0$, and ℓ^1 minimization does not correct even a single arbitrary error. However, we can rotate $\text{span}(A)$ so it will not penetrate B_1 , for example by multiplying A on the right with $D \doteq \text{diag}(1, 10, 10)$. Assume we are given \mathbf{y} and \mathbf{x}_0 are such that $\|\mathbf{y} - A\mathbf{x}_0\|_0 \leq 1$. Because D is diagonal, $\|D\mathbf{y} - DA\mathbf{x}_0\|_0 \leq 1$. By construction, (11) holds for DA with $T = 1$. Thus, by Proposition 3, we can recover \mathbf{x}_0 as:

$$\mathbf{x}_0 = \arg \min_{\mathbf{x}} \|D\mathbf{y} - DA\mathbf{x}\|_1. \quad (22)$$

Example 7 shows that rescaling by an appropriate diagonal⁹ matrix can increase T^* , allowing ℓ^1 minimization (4) to correct more errors. The following two results show how to systematically compute such a D , if one exists.

Proposition 8 Let $A \in \mathbb{R}^{m \times n}$ be full-rank, $I \subset \{1, \dots, m\}$, $|I| = n - 1$ and $D \in \mathbb{R}^{m \times m}$ a strictly positive diagonal matrix ($D_{ii} > 0$). For every $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{y} \in \text{span}(A)$ and every $\mathbf{y}' \in \mathbb{R}^m$, $\mathbf{y}' \in \text{span}(DA)$, such that $\forall i \in I$, $y_i = y'_i = 0$, there exists $\alpha \in \mathbb{R}$ such that

$$y'_j = \alpha D_{jj} y_j, \quad \forall j \in M \doteq \{1, \dots, m\}. \quad (23)$$

proof: Let \mathbf{x} and \mathbf{x}' such that $\mathbf{y} = A\mathbf{x}$ and $\mathbf{y}' = DA\mathbf{x}'$. From $\forall i \in I$, $y_i = y'_i = 0$ and the strict positivity of D we have that $\forall i \in I$ $A_{i*}\mathbf{x} = A_{i*}\mathbf{x}' = 0$. Since $|I| = n - 1$ and the rank of A is n , there exists α such that $\alpha A\mathbf{x} = A\mathbf{x}'$. Multiplying both sides by D gives (23). \square

Theorem 9 Let $A \in \mathbb{R}^{m \times n}$ be full-rank. For every $\mathbf{x}_0 \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ such that $\|\mathbf{y} - A\mathbf{x}_0\|_0 \leq T$ one can recover \mathbf{x}_0 using the following ℓ^1 -minimization:

$$\mathbf{x}_0 = \arg \min_{\mathbf{x}} \|D\mathbf{y} - DA\mathbf{x}\|_1 \quad (24)$$

where D is a strictly positive diagonal matrix if and only if for all subsets $I \subset M$ containing $n - 1$ elements, all subsets $J \subset M \setminus I$ containing $T = d + 1$ elements, and for some $\mathbf{y} \in \mathbb{R}^m$ such that

$$\mathbf{y} \in \text{span}(A), \quad \forall i \in I \quad y_i = 0, \quad (25)$$

the following holds:

$$\sum_{j \in J} D_{jj} |y_j| < \sum_{j \in M \setminus J} D_{jj} |y_j| \quad (26)$$

proof: Modify Propositions 3 and 4 by substituting DA for A . Since D is a diagonal matrix $\|D\mathbf{y} - DA\mathbf{x}\|_0 \leq T$ from the modified (10) can be replaced with $\|\mathbf{y} - A\mathbf{x}\|_0 \leq T$. Using Proposition 8 the modified (13) and (14) can be replaced by (25) and (26), completing the proof. \square

Since (26) is linear with respect to the elements of D , we can write (26) for all subsets $I \subset M$ containing $n - 1$ indices, all subsets $J \subset M \setminus I$ containing $T = d + 1$ elements, and some \mathbf{y} for which (25) holds, and then solve for a feasible D^* using linear programming.

⁹Note that if D is not necessarily diagonal, then even if (11) holds for DA with some T , it does not imply that we can use (22) to solve for (1) with the same T since we can find \mathbf{y} and \mathbf{x}_0 such that $\|\mathbf{y} - A\mathbf{x}_0\|_0 = T$ but $\|D\mathbf{y} - DA\mathbf{x}_0\|_0 > T$. While this does not necessarily rule out the existence of a non diagonal matrix that can increase the number of cases in which we can use (22) to solve for (1), in this current paper we consider only diagonal matrices.

Example 10 (Improving robustness of Gaussian codebooks) *We randomly sample 250 Gaussian code books of size 15×3 , A , and compute $T^*(A)$ before and after scaling by the optimal D^* calculated using Theorem 9. The median $T^*(A)$ is 2, while the median $T^*(D^*A)$ is 3. In 95% of cases, $T^*(D^*A) > T^*(A)$, and in 10% of cases $T^*(D^*A) = 4$, which is also an upper bound for T for this matrix size [3]. Thus, the vast majority of Gaussian codebooks are suboptimal, and many cannot even be rescaled to an optimal codebook by any diagonal matrix D !*

5 Application to robust state estimation

Most recent results on ℓ^0 - ℓ^1 equivalence deal with very high-dimensional matrices, giving asymptotic guarantees of equivalence. In the following example, we demonstrate that even with matrices of small dimension, we can benefit from this equivalence.

Consider a vehicle moving on a plane, equipped with inertial sensors that continuously measure its velocity and orientation. The sensors may drift over time. The vehicle is also equipped with external sensors that measure its position at regular time intervals. The position sensors may occasionally be corrupted and produce an erroneous output. The goal is for the vehicle to estimate its current position based on previous sensor readings. If the position sensors were only affected by small white noise, then this would be a classic Kalman filtering problem. However, the Kalman filter, which gives every measurement an equal weight (based on the covariance matrix), is unable to detect and ignore the corrupted readings. By instead applying ℓ^1 -based error correction, we can ignore these erroneous readings. Moreover, the ℓ^1 minimization is also robust to small additive Gaussian noise (see e.g., [5, 1]).

Figure 4 shows the advantage of ℓ^1 minimization over the Kalman filter in this scenario. The state space is 4 dimensional: position (x2), orientation and velocity. Each estimate is based on the last 20 (x2) position readings. Therefore the matrix used in the ℓ^1 minimization is of size 40×4 . For a matrix of this size our algorithm has a running time of only a few seconds. Thus our algorithm enables the engineer to better choose how many position readings to consider for each estimate. Alternatively it enables the engineer to evaluate the expected performance of his implementation. This example also shows us that most of the results in the literature which deals with randomly generated matrices will probably not predict well the behavior of structured matrices like in this example. The matrices generated by our simulation for this application had a T^* between 4 and 5. Random generated matrices of the same size, on the other hand, had on average T^* equal to 7.

6 Discussion

The conditions (and the associated algorithms) given in this paper allow exact analysis of the ℓ^1 - ℓ^0 equivalence properties of small to moderate sized linear systems, allowing the engineer to guarantee optimal robustness or sparse recovery via ℓ^1 -minimization. Small matrix results as in Examples 6 and 10, and in §5, provide a useful counterpart to the asymptotics in the literature, allowing a more informed selection of measurement ensembles for compressed sensing or robust error correction codes. Exact polynomial-time algorithms for the problem considered here seem unlikely, but its NP-hardness is open. Many practical problems demand good bounds on $T^*(A)$ for a very large matrix A (e.g. $5,000 \times 800$ in [6]). Good polynomial-time approximation algorithms are still needed to provide rigorous performance guarantees for such large, structured matrices. A related question for future work is to analyze whether randomly sampled \mathbf{y} from Algorithm 1 can be used to provide probabilistic bounds on T^* , with a certain confidence.

A Figures

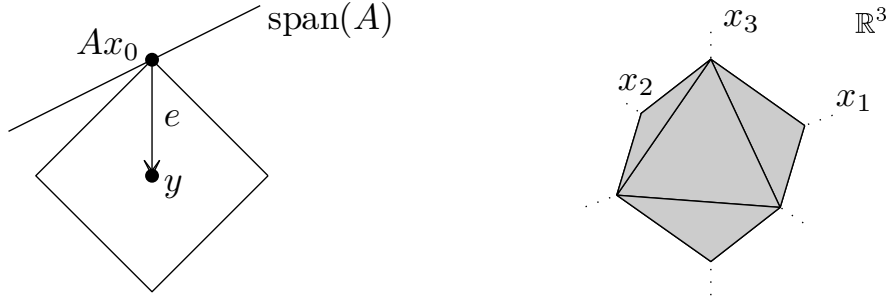


Figure 1: **Left:** Illustration of the relation between (10) and (11) with $m = 2$, $n = 1$, and $T = 1$. Because $T = 1$ the difference \mathbf{e} between $A\mathbf{x}_0$ and \mathbf{y} is parallel to one of the coordinate axes, in this case the vertical axis. The ℓ^1 minimization finds the ℓ^1 ball of smallest radius around \mathbf{y} that touches the subspace spanned by A . The recovered \mathbf{x} is the touching point, and it equals \mathbf{x}_0 iff the subspace spanned by A does not penetrate the ℓ^1 ball. The inequality in (11) verifies that whenever the subspace spanned by A is translated to one of the $(T - 1)$ -faces of the ℓ^1 ball, penetration does not occur. **Right:** For $m = 3$, there exists an A such that (11) and (10) hold: take the one-dimensional span of A to be parallel to the reader's line of sight. Here, the span of A , translated by any difference vector whose 0-norm equals 1, does not penetrate the ℓ_1 ball. This is illustrated by the fact that all the 6 vertices of the ℓ_1 ball are visible to the reader. Since not all edges are visible, for $T = 2$, (11) does not hold (and thus neither does (10)).

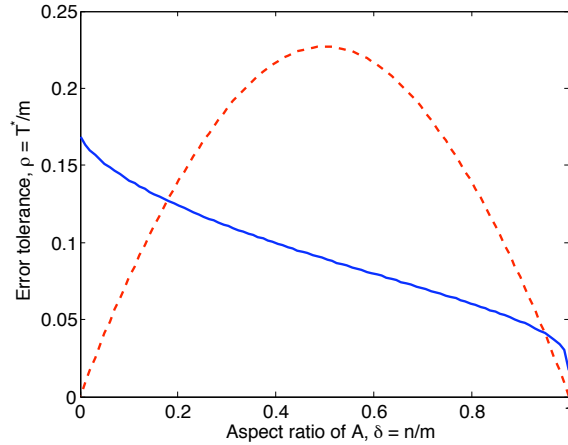


Figure 2: **Red curve:** zero level-set of the exponent $H(\delta) - \rho - H(\rho)$ in (21). For (ρ, δ) pairs above this curve, the complexity of Algorithm 1 is exponentially smaller than that of the competing algorithm outlined in §2. **Blue curve:** asymptotic value of $\rho = T^*(A)/m$ for Gaussian matrices A , computed using the method of [10]. The two curves intersect at $\delta \approx .175$ and $\delta \approx .95$. For $\delta < .175$ or $\delta > .95$, our algorithm has lower complexity. These are arguably the two most interesting cases for equivalence, the first corresponding to highly robust error correction (tall A), and the second to sparse signal reconstruction from few linear measurements (B with $BA = 0$ wide).

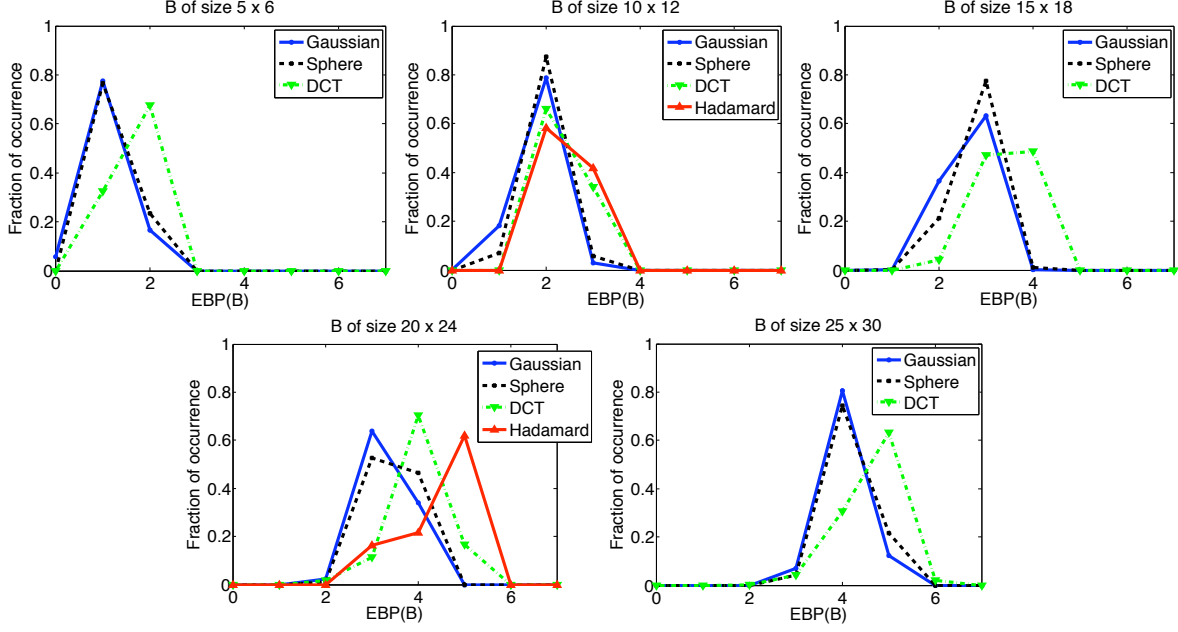


Figure 3: Histogram of EBP for various $5n \times 6n$ random matrices for $n = 1$ (top left) to $n = 5$ (bottom right). Horizontal axis: $EBP(B)$. Vertical axis: fraction of matrices with a given EBP.

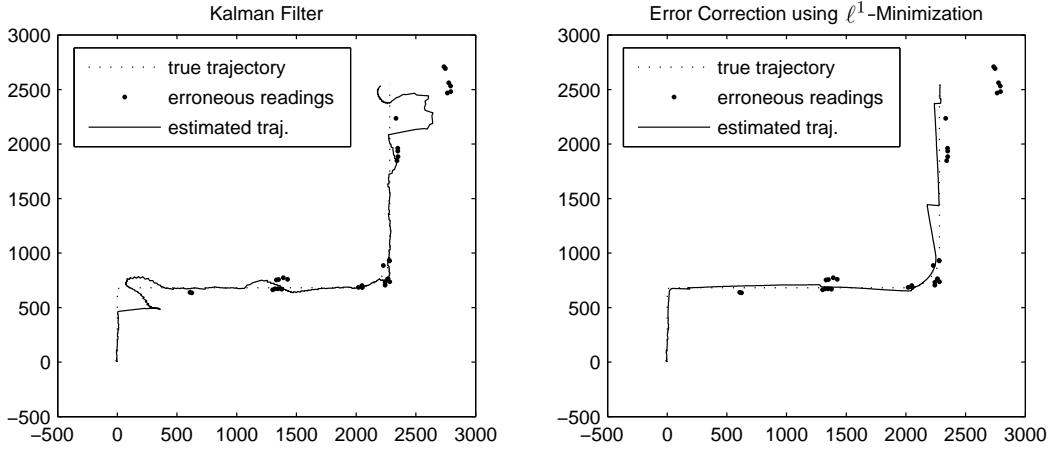


Figure 4: A qualitative comparison between Kalman filter and ℓ^1 filtering. The erratic behavior of the Kalman filter is distinct compared to the ℓ^1 estimate. The axes units are in meters; the vehicle's maximum speed is 30 meters per second; a new position reading is received once every second; the corrupted position readings can be up to 400 meters from the true position (uniform distribution). The maximum position error and the standard deviation of the position error of the ℓ^1 filtering are 105_{meters} and 18_{meters} , respectively. The maximum position error and the standard deviation of the position error of the Kalman filtering are 378_{meters} and 80_{meters} , respectively.

References

- [1] D. Donoho, “For most large underdetermined systems of linear equations the minimal ℓ^1 -norm near solution approximates the sparsest solution,” *preprint*, <http://www-stat.stanford.edu/~donoho/Reports/>, 2004.
- [2] —, “For most large underdetermined systems of linear equations the minimal ℓ^1 -norm solution is also the sparsest solution,” *Comm. Pure and Applied Math.*, vol. 59, no. 6, pp. 797–829, 2006.
- [3] —, “Neighborly polytopes and sparse solution of underdetermined linear equations,” *IEEE Trans. Info. Theory*, 2006.
- [4] E. Candes, M. Rudelson, T. Tao, and R. Vershynin, “Error correction via linear programming,” in *IEEE Symposium on FOCS*, 2005, pp. 295–308.
- [5] E. Candes and P. A. Randall, “Highly robust error correction by convex programming,” *preprint*, <http://arxiv.org/abs/cs.IT/0612124>, 2006.
- [6] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” to appear in *IEEE Trans. Pattern Analysis and Machine Intelligence*. <http://www.dsp.ece.rice.edu/cs/>, 2008.
- [7] E. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 489–509, 2004.
- [8] E. Amaldi and V. Kann, “On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems,” *Theoretical Computer Science*, vol. 209, pp. 237–260, 1998.
- [9] E. Candes and T. Tao, “Near-optimal signal recovery from random projections: universal encoding strategies,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 5406–5425, 2004.
- [10] D. Donoho and J. Tanner, “Counting faces of randomly projected polytopes when projection radically lowers the dimension,” submitted to *Journal of the American Mathematical Society*, 2007.
- [11] J. Wright and Y. Ma, “Hardness of polytope neighborliness and related problems in sparse signal representation,” *CSL Technical Report, University of Illinois at Urbana-Champaign*, 2008.
- [12] E. J. Candes and T. Tao, “Decoding by linear programming,” *IEEE Trans. Inform. Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.
- [13] M. Elad, “Optimized projections for compressed sensing,” *IEEE Transactions on Signal Processing*, vol. 55, no. 12, pp. 5695–5702, Dec. 2007.
- [14] E. W. Weisstein, “Hadamard matrix,” *MathWorld–Wolfram*, <http://mathworld.wolfram.com/HadamardMatrix.html>.
- [15] E. Candes, “Compressive sampling,” in *Proc. International Congress of Mathematicians*, 2006.