Compressed Sensing via Iterative Support Detection*

Yilun Wang† Wotao Yin‡

Abstract

We present a new compressive sensing reconstruction method ISD, aiming to achieve fast reconstruction and a reduced requirement on the number of measurements compared to the classical ℓ_1 minimization approach. ISD addresses failed cases of ℓ_1–based construction due to insufficient measurements, in which the returned signals are not equal or even close to the true signals. ISD will learn from such signals and solve new minimization problems that return a perfect or a better signal. Specifically, given an incorrect signal, ISD detects an index set I that includes components most likely to be true nonzeros and solves $\min\{\sum_{i\notin I}|x_i| : Ax = b\}$ for a new signal x, and it iterates these two steps alternatively using latest x and I from one another until convergence. ISD differs from the orthogonal matching pursuit (OMP) method, as well as its variants, in two aspects. First, although both methods generate index sets iteratively, the index set I in ISD is not necessarily nested or increasing over the iterations. Second, the OMP–family methods at each of their iterations fix x_i, $i \in I$, and update the remaining components of x, but the ISD minimization problem above updates all the components of x.

To analyze the performance of ISD, we generalize the *Null Space Property* to *Truncated Null Space Property* and present our analysis on the latter.

We introduce an efficient implementation of ISD, called threshold–ISD, for recovering signals with fast decaying distributions of nonzeros from compressive measurements. Numerical experiments show that threshold–ISD has significant overall advantages over the classical ℓ_1 minimization approach, as well as two other state–of–the–art algorithms such as the iterative reweighted ℓ_1 minimization algorithm (IRL1) and the iterative reweighted least–squares algorithm (IRLS).

Contents

*This research was supported in part by NSF CAREER Award DMS-07-48839, ONR Grant N00014-08-1-1101, and an Alfred P. Sloan Research Fellowship.

†School of Civil and Environmental Engineering, Cornell University, Ithaca, New York, 14853, U.S.A. (yilun.wang@gmail.com). The author's part of work was done when he was a doctoral student at Rice University.

‡Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 77005, U.S.A. (wotao.yin@rice.edu).

1 Introduction and Contributions

Brought to the research forefront by Donoho [17] and Candes, Romberg, and Tao [6], compressive sensing (CS) reconstructs a sparse unknown signal from a small set of linear projections. Let $\bar{x} \in \mathbb{R}^n$ denote an unknown signal and $b := A\bar{x} \in \mathbb{R}^m$ represent a set of m linear projections of $\bar{x}$. The optimization problem

$$(\mathrm{P}_{\ell_0}) \quad \min_x \|x\|_0 \quad \text{s.t.} \quad Ax = b, \tag{1}$$

where $\|x\|_0$ is defined as the number of nonzero components of x, can exactly reconstruct a k-sparse[1] vector $\bar{x}$ from $O(k)$ random projections. (Throughout this paper, $\bar{x}$ is used to denote the true signal to reconstruct.) However, because $\|x\|_0$ is non-convex and combinatorial, (P_{ℓ_0}) is impractical for real applications. A practical alternative is the basis pursuit (BP) problem

$$(\mathrm{BP}) \qquad \min_x \|x\|_1 \quad \text{s.t.} \quad Ax = b, \tag{2}$$

$$\text{or} \qquad \min_x \ \|x\|_1 + \frac{1}{2\rho}\|Ax - b\|_2^2, \tag{3}$$

where (2) is used when b contains little or no noise and, otherwise, (3) (also called the basis pursuit denoising problem) is used with a proper parameter $\rho > 0$. The BP problems have been known to yield sparse solutions under certain conditions (see [18, 15, 22] for explanations) and also have efficient algorithms such as [3, 26, 24, 21, 19, 30, 33, 32]. It is shown in [5, 28] that, when A is a Gaussian random or partial Fourier ensemble, BP returns a solution equal to $\bar{x}$ with high probability from $m = O(k \log(n/k))$ and $O(k \log(n)^4)$ linear measurements, respectively, which are much smaller than n. Requirements for general matrices A are studied in [4, 14, 20, 28, 34, 37]. Compared to (P_{ℓ_0}), BP is much easier to solve but requires significantly more measurements. In this paper, we propose an iterative support detection method (abbreviated as ISD) that runs as fast as those for BP but requires significantly fewer measurements.

ISD enhances sparse and compressible signal reconstruction by learning from failed BP–based reconstructions that arise when the number of linear measurements is not sufficient. In a failed reconstruction, the solution of BP is not equal or even close to the true signal $\bar{x}$. One would normally give up, but ISD learns from the incorrect signal and solves a new optimization problem that returns a perfect or a better solution, requiring no additional measurements. Given an incorrect solution x, a set I of components that are likely to be *nonzeros* in the true signal $\bar{x}$ are detected from x, and the new problem

$$(\text{Truncated } \ell_1 \text{ minimization}) \quad \min_x \ \|x_T\|_1 \quad \text{s.t.} \quad Ax = b \tag{4}$$

(or the one corresponding to (3)) is solved for $T := I^C = \{1, \ldots, n\} \setminus I$ where $\|x_T\|_1 := \sum_{i \in T} |x_i|$. The solution of (4), if not sparse or compressible (i.e., not resembling $\bar{x}$), is learned again to update I and another instance of (4) with the new $T = I^C$ is solved. The framework of ISD is described in Section 2, followed by a simple demo to illustrate that successful reconstruction of a sparse Gaussian signal requires very few measurements. This demo also shows its robustness to *false detections* defined as the elements of I that are not true nonzeros. In this paper, we call the zero and nonzero components of $\bar{x}$ the *true zero* and *true nonzeros*, respectively.

[1] A k-sparse vector has no more than k nonzero components.

To provide theoretical justifications for ISD, we study a so-called *truncated null space property* of A, an extension of the null space property originally studied in [11] and later in [35, 36, 17, 13] which is more general than the widely used restricted isometry property [7]. Based upon this study, we establish sufficient conditions for (4) to return the sparse true signal $\bar{x}$. When $\bar{x}$ is not exactly sparse, its exact reconstruction is generally impossible. However, we show an error bound between the solution of (4) and $\bar{x}$, which is proportional to the tail size of $\bar{x}_T$. Built upon these results for a single instance of (4), the following result for ISD is obtained: the chances for (4) to return a sparse true signal $\bar{x}$ improve over the iterations as I improves; specifically, among the new detections in each I, the true nonzeros must be more than the false nonzeros by a certain factor depending on the sparsity of $\bar{x}$, certain properties of A, as well as the previous I. We note that these result are independent of specific support detection methods used to generate I.

We implemented a "first significant jump" thresholding rule in *threshold–ISD* and tested it on synthetic and real signals. This rule is studied in Subsection 4.1 below for sparse or compressible signals with components having a fast decaying distribution of nonzeros.

To reconstruct a signal, one or more instances of (4) need to be solved. Solving one instance can benefit the solution of the next one by warm-starting, namely, starting solving the latter from the returned solution of the former rather than from scratch. We applied the solver YALL1 [33] for solving (4) with warm starts, which significantly reduced the number of iterations of YALL1. Although YALL1 can return highly accurate solutions, thresholding does not require an accuracy much higher than the size of the threshold. Therefore, dynamic stopping tolerances were set according to the thresholds used in YALL1.

Our numerical experiments compared threshold–ISD to BP, the iteratively reweighted least squares algorithm [10] (IRLS) and the iteratively reweighted ℓ_1 minimization algorithm [9] (IRL1). IRLS and IRL1 are known for their state-of-the-art reconstruction rates, on both noiseless and noisy measurements. With the same number of measurements, threshold–ISD and IRLS returned better signals than IRL1, which is further better than BP. Comparing between threadhold–ISD and IRLS, the former ran much faster. However, unless the true signal had a fast decaying distribution of nonzeros, none of threshold–ISD, IRLS, and IRL1 was significantly better than BP in term of chance of successful reconstruction.

The remaining of this paper is organized as follows. In Section 2, the algorithmic framework of ISD is given along with a simple demo. Sections 3 presents our preliminary theoretical results on ISD. Section 4 and 5 study the details of threshold–ISD and present our numerical results, respectively. Section 6 is devoted to conclusions and discussions on future research.

2 Algorithmic Framework

We first present the algorithmic framework of ISD.

Input: A and b

1. Set the iteration number $s \leftarrow 0$ and initialize the set of detected entries $I^{(s)} \leftarrow \emptyset$;
2. While the stopping condition is not met, do
   (a) $T^{(s)} \leftarrow (I^{(s)})^C := \{1, 2, \ldots, n\} \setminus I^{(s)}$;
   (b) $x^{(s)} \leftarrow$ solve (4) for $T = T^{(s)}$;
   (c) $I^{(s+1)} \leftarrow$ support detection using $x^{(s)}$ as the reference;
   (d) $s \leftarrow s + 1$.

Since $T^{(0)} = \{0, 1, \ldots, n\}$, the first instance of (4) solved in Step (b) is simply an instance of BP (2). Although support detection can be done in a greedy way similar to those in the *greedy algorithms* such as OMP [29], StOMP [16], and CoSaMP [27], ISD is significantly different from any greedy algorithm for the two reasons given in the abstract. Furthermore, iterative support detection can even be applied with a greedy algorithm by calling the greedy algorithm iteratively and, at iteration s, always including $I^{(s)}$ within its so–called greedy index sets.

A demo: We generated a sparse signal $\bar{x}$ of length $n = 200$ with $k = 25$ nonzero numbers independently sampled from the standard normal distribution and assigned to randomly chosen components of $\bar{x}$. We let $m = 60$, created a Gaussian random $m \times n$ matrix A, and set $b := A\bar{x}$. ISD was applied to reconstruct $\bar{x}$ from A and b, where Step (c) was implemented as

$$I^{(s+1)} \leftarrow \{i : |x_i^{(s)}| > \epsilon^{(s)}\}, \quad \epsilon^{(s)} := \|x^{(s)}\|_\infty / 5^{(s+1)}. \tag{5}$$

The true signal $\bar{x}$ was returned in merely four iterations. The solutions at the end of the four iterations are depicted in the four subplots of Figure 1, respectively, where the components of $\bar{x}$ are marked as $\bullet$ and the nonzero components of $x^{(s)}$ are marked separately as $\circ$ and $\diamond$, standing for true and false nonzeros, respectively. The thresholds $\epsilon^{(s)}$ are shown as green lines. To measure the solution qualities, we give the quadruplet "(total, det, c–det, w–det)" and "Err" in the title of each subplot, which are defined as follows:

- (total, det, c-det, w-det):
  - total: the number of total nonzero components of the true signal $\bar{x}$.
  - det: the number of detected nonzero components, equal to $|I^{(s+1)}|$ = (c–det)+(w–det).
  - c-det: the number of *correctly* detected nonzero components, i.e, $|I^{(s+1)} \cap \{i : \bar{x}_i \neq 0\}|$.
  - w-det: the number of *falsely* detected nonzero components, i.e, $|I^{(s+1)} \cap \{i : \bar{x}_i = 0\}|$.
- Err: the relative error $\|x^{(s)} - \bar{x}\|_2 / \|\bar{x}\|_2$.

From the upper left subplot, it is clear to see that $x^{(0)}$, which was the BP solution, contained a large number of false nonzeros and had a large relative error. However, most of its correct nonzero components were relatively large in magnitude (as a consequence of $\bar{x}$ having a relatively fast decaying distribution of nonzeros), the thresholding method (5) with the threshold $\epsilon^{(0)} = \|x^{(0)}\|_\infty / 5$ detected 12 nonzeros, among which 10 were true nonzeros and 2 were not. In spite of the 2 false detections, the detection yielded $T^{(1)}$ that was good to let (4) return a much better solution $x^{(1)}$, depicted in the upper right subplot. This solution further allowed (5), now having the tighter threshold $\epsilon^{(1)} = \|x^{(1)}\|_\infty / 5^2$, to yield 19 detected true nonzeros with 8 false detections. Noticeably, most of true nonzeros with large magnitude had been correctly detected. The next solution $x^{(2)}$, depicted in the bottom left subplot, became even better, which well matched the true signal $\bar{x}$ except for tiny false nonzero components. (5) detected 24 true nonzeros of $\bar{x}$ from $x^{(2)}$ and 7 false nonzeros, and $x^{(3)}$ had exactly the same nonzero components as $\bar{x}$, as well as an error almost as low as the double precision.

Obviously, ISD is insensitive to a small number of false detections and has an attractive self-correction capacity. Critical to a final successful reconstruction is not to fix any of the detected components but to let all the components $x_1, \ldots, x_n$ be free variables in (4). It is also important to look for candidate nonzero components among all the components of $x^{(s)}$, whether or not one has previously been detected. In the above test, the results would be much worse if components had been only allowed to enter $I^{(s)}$ but not allowed to leave. In addition, the performance of ISD is invariant to small variations in the thresholding tolerance in (5). On the same set of data, we also tried $\epsilon^{(s)} = \|x^{(s)}\|_\infty / \beta^{(s+1)}$ for $\beta = 3$ and $\beta = 1.5$, and obtained $\bar{x}$ with high accuracies in 4 and 6 iterations, respectively. More numerical behaviors of ISD are reported in Sections 4 and 5 below.

3 Preliminary Theoretical Analysis

The preliminary theoretical results in this section explain under what conditions ISD can successfully reconstruct $\bar{x}$, especially, from measurements that are not enough for BP. Most of the results are based on a property of the sensing matrix A defined in Subsection 3.1. Focusing on the minimization problem (4), Subsections 3.2 and 3.3 study exact reconstruction conditions for sparse signals and reconstruction errors for compressible signals, respectively. Finally, Subsection 3.4 gives a sufficient condition for ISD to improve the chance of perfect reconstruction over its iteration.

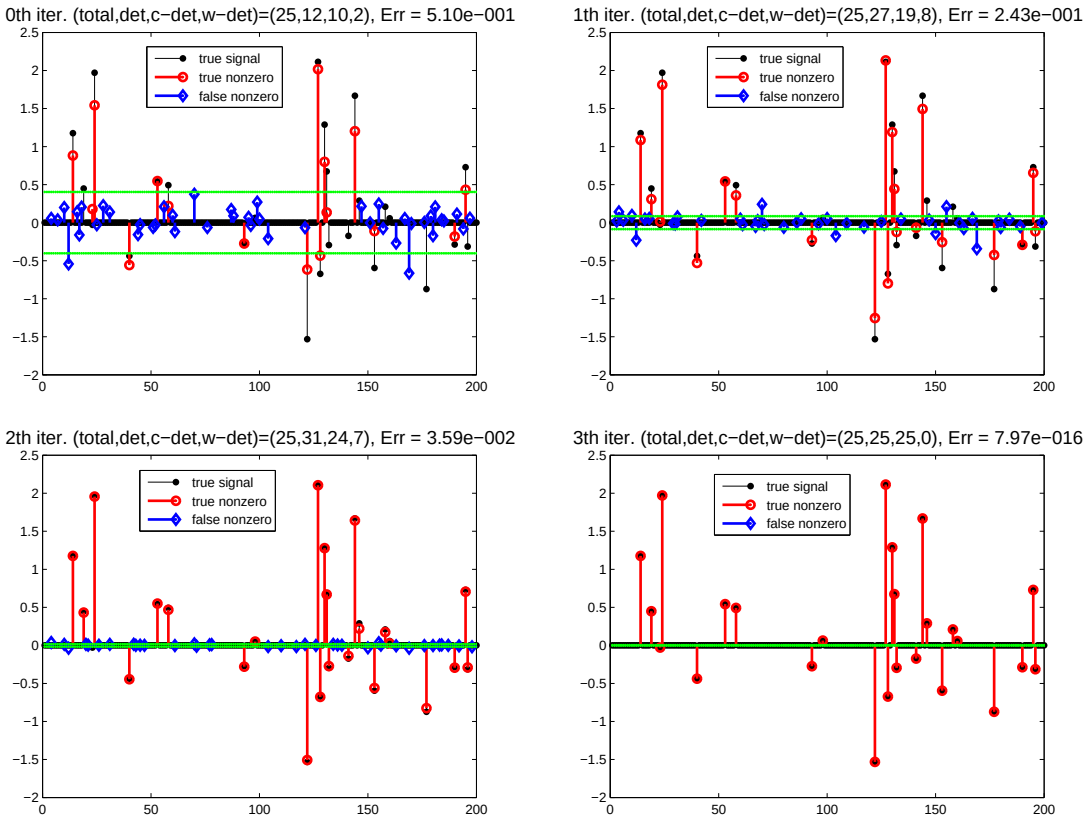


Figure 1: The 0th iteration (upper-left), 1st ISD iteration (upper-right), 2nd ISD iteration (bottom-left), and 3rd ISD iteration (bottom-right).

3.1 The Truncated Null Space Property

We start with introducing the *truncated null space property* (t-NSP), a generalization of the *null space property* (NSP) studied in [35, 36, 17, 11, 13]. The NSP is used in slightly different forms and difference names in these papers. We adopt the definition in [13]: a matrix $A \in \mathbb{R}^{m\times n}$ satisfies the NSP of order L for $\gamma > 0$ if

$$\|\eta_S\|_1 \leq \gamma \|\eta_{S^C}\|_1 \tag{6}$$

holds for all index sets S with $|S| \leq L$ and all $\eta \in \mathcal{N}(A)$, the null space of A. In (6) and the rest of this paper, $\eta_S \in \mathbb{R}^n$ denotes the subvector of η consisting of η_i for $i \in S \subset \{1, 2, \ldots, n\}$, and S^C denotes the complement of S with respect to $\{1, \ldots, n\}$.

With $\gamma < 1$, the NSP says that any nonzero vector η in the null space of A cannot have an ℓ_1-mass concentrated on any set of L or fewer elements. [12] proposes a semidefinite relaxation to test the NSP for general matrices. A sufficient exact reconstruction condition for BP is given in [13] based on the NSP: the true k–sparse signal $\bar{x}$ is the unique solution of BP if A has the NSP of order $L \geq k$ and $0 < \gamma < 1$.

The more widely known *restricted isometry property* (RIP) is more restricted than the NSP for establishing recoverability and stability results for BP [8, 7]. A matrix $A \in \mathbb{R}^{m\times n}$ satisfies the RIP of order k for a constant $\delta_k \in (0, 1)$ if

$$(1 - \delta_k)\|x\|_2 \leq \|Ax\|_2 \leq (1 + \delta_k)\|x\|_2,$$

holds for all k-sparse vectors x. An RIP-based sufficient reconstruction condition is given in [8] as follows: the true k–sparse signal is the unique solution of BP if A satisfies the RIP with $\delta_{2k} + \delta_{3k} < 1$.

The NSP is more relaxed than the RIP. It is shown in [11] that if A satisfies the RIP of order $k := j + j'$ for a given $\delta \in (0, 1)$, where $j, j' \geq 1$ are integers, then A satisfies the NSP of order j for $\gamma := \frac{1+\delta}{1-\delta}\sqrt{\frac{j}{j'}}$. Note that the condition $\gamma < 1$ among the above NSP-based sufficient reconstruction conditions holds as long as j is sufficiently smaller than j'.

In order to analyze the minimization problem (4) with a truncated ℓ_1–norm objective, we now generalize the NSP to the t-NSP.

Definition 1. *A matrix A satisfies the t-NSP of order L for $\gamma > 0$ and $0 < t \leq n$ if*

$$\|\eta_S\|_1 \leq \gamma \|\eta_{(T\cap S^C)}\|_1 \tag{7}$$

holds for all sets $T \subset \{1, \ldots, n\}$ with $|T| = t$, all subsets $S \subset T$ with $|S| \leq L$, and all $\eta \in \mathcal{N}(A)$ — the null space of A.

For simplicity, we use t-NSP(t, L, γ) to denote the t-NSP of order L for γ and t, and use $\bar{\gamma}$ to replace γ and write t-NSP$(t, L, \bar{\gamma})$ if $\bar{\gamma}$ is the infimum of all the feasible γ satisfying (7).

Notice that when $t = n$, the t-NSP reduces to the NSP. Compared to the NSP, the inequality (7) in the t-NSP has an extra set limiter T of size t applied to η_S and η_{S^C} in (6). It is introduced for $\|x_T\|_1$.

Clearly, for a given A and t, $\bar{\gamma}$ is monotonic increasing in L. On the other hand, fixing L, $\bar{\gamma}$ is monotonically decreasing in t. If γ is fixed, then the largest possible L is monotonically increasing in t.

3.2 Sufficient Recovery Conditions of Truncated ℓ_1 Minimization

We first analyze the model (4) and explain why its may require significantly fewer measurements than BP. Below we present a sufficient exact reconstruction condition for L–sparse signals.

Theorem 3.1. *Let $\bar{x}$ be a given vector, $\bar{S} = \{i : \bar{x}_i \neq 0\}$, and T be given such that $T \cap \bar{S} \neq \emptyset$. Assume that a matrix A satisfies t-NSP$(t, L, \bar{\gamma})$ for $t = |T|$. If $\|\bar{x}_T\|_0 \leq L$ and $\bar{\gamma} < 1$, then $\bar{x}$ is the unique minimizer of* (4) *for $b := A\bar{x}$.*

Proof. The true signal $\bar{x}$ uniquely solves (4) if and only if

$$\|\bar{x}_T + v_T\|_1 > \|\bar{x}_T\|_1, \ \forall v \in \mathcal{N}(A), v \neq \mathbf{0}. \tag{8}$$

Let $S := T \cap \bar{S}$. Since $\|\bar{x}_S\|_1 = \|\bar{x}_T\|_1$, we have

$$\begin{aligned}\|\bar{x}_T + v_T\|_1 &= \|\bar{x}_S + v_S\|_1 + \|\mathbf{0} + v_{T\cap S^C}\|_1 \\ &= \underbrace{(\|\bar{x}_S + v_S\|_1 - \|\bar{x}_S\|_1 + \|v_S\|_1)}_{\geq \mathbf{0}} + \|\bar{x}_T\|_1 \\ &+ (\|v_{T\cap S^C}\|_1 - \|v_S\|_1).\end{aligned}$$

Therefore, having $\|v_S\|_1 < \|v_{T\cap S^C}\|_1$ is sufficient for (8).

If $\|\bar{x}_T\|_0 \leq L$, then $|S| \leq L$. According to the definition of t-NSP$(|T|, L, \bar{\gamma})$, it holds that $\|v_S\|_1 \leq \bar{\gamma}\|v_{T\cap S^C}\|_1 < \|v_{T\cap S^C}\|_1$. □

We note that t-NSP$(|T|, L, \bar{\gamma})$ is more strict than what is needed when T is given because (7) is required to hold for all T with $|T| = t$ rather than a specific T. The assumption $T \cap \bar{S} \neq \emptyset$ in Theorem 3.1 is not essential because $T \cap \bar{S} = \emptyset$ is not a case of interest. If $T \cap \bar{S} = \emptyset$, then $\bar{x}$ is a solution of (4). If in addition A_{T^C} has independent columns, then $\bar{x}$ is the unique solution.

The following lemma states that Gaussian matrices of appropriate sizes have the t-NSP. Our proof is inspired by the work of Zhang [38].

Lemma 3.1. *Let $m < n$. Assume that $A \in \mathbb{R}^{m\times n}$ is either a standard Gaussian matrix (i.e., one with i.i.d. standard normal entries) or a rank-m matrix with its m rows all orthogonal to an $(n-m)$–dimensional standard Gaussian linear subspace (i.e., existing a standard Gaussian matrix $B \in \mathbb{R}^{n\times(n-m)}$ such that $AB = \mathbf{0}$). Given an index set T, with probability greater than $1 - e^{-c_0(n-m)}$, the matrix A satisfies t-NSP(t, L, γ) for $t = |T|$ and $\gamma = \frac{\sqrt{L}}{2\sqrt{k(d)-\sqrt{L}}}$, where*

$$k(d) := c\frac{m-d}{1+\log(\frac{n-d}{m-d})}, \tag{9}$$

$d = n - |T|$, and $c_0, c > 0$ are absolute constants independent of the dimensions m, n and d.

This lemma is used to prove Theorem 3.2 below. According to its definition, d equals the number $|I|$ of detected entries (possibly including both correct and wrong detections) in ISD. d determines the parameter $k(d)$. Jointly with the parameter L, d further determines the t-NSP parameter γ. We note that the γ given in Lemma 3.1 is not necessarily tight, namely, the same matrix A may satisfy t-NSP(t, L, γ_1) for some $\gamma_1 < \gamma$ with the same probability. For exact reconstruction, the sufficient condition in Theorem 3.1 requires $\bar{\gamma} < 1$ (i.e., exists $\gamma < 1$) and $\|\bar{x}_T\|_0 \leq L$. Therefore, $k(d)$ plays a pivoting role. Therefore, we analyze its formula (9) at the end of this subsection.

Proof. Let the columns of B span $\mathcal{N}(A)$, i.e., $B \in \mathbb{R}^{(n-m)\times m}$ and $AB = \mathbf{0}$, and P_T refer to projection to the coordinates T. Then, $\Lambda = \{v_T : v \in \mathcal{N}(A)\} = \{(P_T B)w : w \in \mathbb{R}^{n-m}\}$ is a randomly drawn subspace in $\mathbb{R}^{|T|}$ with dimensions up to $(n-m)$. Kashi-Garnaev-Gluskin's result states that for any $p < q$, with

$$\text{probability} \geq 1 - e^{-c_0 p},$$

a randomly drawn p-dimensional subspace $V_p \in \mathbb{R}^q$ satisfies

$$\frac{\|z\|_1}{\|z\|_2} \geq \frac{c_1\sqrt{q-p}}{\sqrt{1+\log(q/(q-p))}},\ \forall z \in V_p, z \neq \mathbf{0},$$

where c_0 and c_1 are independent of the dimensions. Applying this result with $q := |T| = n-d$ and $p := n-m$, we obtain

$$\frac{\|v_T\|_1}{\|v_T\|_2} \geq \frac{c_1\sqrt{(n-d)-(n-m)}}{\sqrt{1+\log((n-d)/((n-d)-(n-m)))}},\ \forall v \in \mathcal{N}(A),\ v \neq \mathbf{0}.$$

or

$$\frac{\|v_T\|_1}{\|v_T\|_2} \geq \frac{c_1\sqrt{m-d}}{\sqrt{1+\log\frac{n-d}{m-d}}},\ \forall v \in \mathcal{N}(A),\ v \neq \mathbf{0}.$$

Let $k(d)$ be defined in (9) where $c = \frac{c_1^2}{4}$. For all $S \subset T$ with $|S| \leq L$ we have $\sqrt{k(d)}\|v_T\|_2 \leq \frac{1}{2}\|v_T\|_1$ and thus

$$\|v_S\|_1 \leq \sqrt{|S|}\|v_S\|_2 \leq \frac{\sqrt{L}}{\sqrt{k(d)}}\sqrt{k(d)}\|v_S\|_2 \leq \frac{\sqrt{L}}{\sqrt{k(d)}}\sqrt{k(d)}\|v_T\|_2 \leq \frac{\sqrt{L}}{2\sqrt{k(d)}}\|v_T\|_1,$$

or equivalently

$$\|v_S\|_1 \leq r\|v_{T\cap S^C}\|_1,$$

where $\gamma = \frac{\sqrt{L}}{2\sqrt{k(d)}-\sqrt{L}}$. The lemma follows from the definition of the t-NSP.

□

Theorem 3.1 and Lemma 3.1 lead to the following theorem.

Theorem 3.2. *Let $\bar{x} \in \mathbb{R}^n$, $\bar{S} = \{i : \bar{x}_i \neq 0\}$ and T be given such that $T \cap \bar{S} \neq \emptyset$. Let $m < n$ and $A \in \mathbb{R}^{m\times n}$ be given as in Lemma 3.1. Then, with probability greater than $1 - e^{-c_0(n-m)}$, the true signal $\bar{x}$ is the unique solution of (4) for $b := A\bar{x}$ if*

$$\|\bar{x}_T\|_0 < k(d), \tag{10}$$

where $k(d)$ is defined in (9), $d = n - t = n - |T|$, and $c_0, c > 0$ are absolute constants independent of the dimensions m, n, and d.

Proof. In Lemma 3.1, let $L = \|\bar{x}_T\|_0$ and if $L < k(d)$, then $\gamma < 1$. Then, the result follows from Theorem 3.1. □

Note that when $d = 0$, condition (10) reduces to the existing result for BP: for the same vector $\bar{x}$ and A given in Theorem 3.2 above, with probability greater than $1 - e^{-c_0(n-m)}$, $\bar{x}$ is the unique solution of (4) for b set to $A\bar{x}$ if $\|\bar{x}\|_0 \leq cm(1 + \log(n/m))^{-1}$, namely, the inequality (9) holds for $d = 0$.

In the following parts, we study (9). Recall that in the algorithm, I and T denote the sets of detected and remaining entries, respectively. Let $d = |I| = n - |T| = d_c + d_w$ where d_c and d_w are the numbers of the correct and wrong detections, respectively. It is well-known that for stable exact reconstruction by ℓ_0-minimization, the largest possible number of nonzeros in $\bar{x}$ is $m/2$. Therefore, we only consider the case $\|\bar{x}\|_0 < m/2$. Suppose BP fails to recover $\bar{x}$. Then, according to Theorem 3.2, $k(d) \leq \|\bar{x}\|_0$ with high probability. In the view of (9) with $d = 0$, we have

$$\frac{c}{1 + \log(\frac{n}{m})} < \frac{1}{2}. \tag{11}$$

Keeping this inequality in mind, we get the derivative of $k(d)$

$$k'(d) := -c\left(\frac{1}{1 + \log\left(\frac{n-d}{m-d}\right)} + \frac{n-m}{\left(1 + \log\left(\frac{n-d}{m-d}\right)\right)^2 (n-d)}\right) \tag{12}$$

and see that $k'(d) < 0$, which means that as the number of detections d (possibly including both correct and wrong detections) increases, the number of nonzeros within $\bar{x}_T$ must also decrease in order to satisfy the sufficient condition (10). However, we argue that when most of the detections are correct, $k(d)$ decreases more slowly than d increases, meaning that the left-hand side of (10) decreases faster than its left-hand side, or in other words, (10) becomes more likely to hold.

Assuming $d < m$ (which is a natural assumption, meaning that the number of detections is no more than twice the number of true nonzeros), (11) leads to

$$|k'(d)| < 1. \tag{13}$$

Hence, if the number of correction detections d satisfies

$$d_c > \int_0^d |k'(d)|, \tag{14}$$

it holds that

$$k(d) + d_c > k(d) + \int_0^d |k'(d)| = k(d) - \int_0^d k'(d) = k(0). \tag{15}$$

To interpret (15), notice that according to Theorem 3.2, $\|\bar{x}\|_0 < k(0)$ is a sufficient reconstruction condition for $d = 0$ and that $\|\bar{x}_T\|_0 = \|\bar{x}\|_0 - d_c < k(d)$ or $\|\bar{x}\|_0 < k(d) + d_c$ is the reconstruction sufficient condition for $d > 0$; hence, (15) means that the sufficient reconstruction condition becomes loose. Furthermore, from (13), the right-hand side of (14) is strictly less than $d = d_c + d_w$, so condition (14) can possibly hold. To have (14), support detection must return enough number of correct detections. In addition, the higher the sample ratio m/n is, the smaller $|k'(d)|$ is and thus (14) becomes easier to hold and similarly (15) is more likely to hold with a larger difference, making the sufficient reconstruction condition even easier to satisfy. In practice, $\frac{c}{1+\log(\frac{n}{m})}$ is much smaller than $\frac{1}{2}$, or even often smaller than $\frac{1}{5}$ and correspondingly $-2/5 < k'(d) < 0$. Therefore, ISD has a higher chance to exactly reconstruct $\bar{x}$ than BP from the same number of measurements as long as an effective support detection means is available. This hints that developing effective support detection methods is critical for the success of ISD. See subsection 4.1 below for more discussions.

3.3 Stability of Truncated ℓ_1 Minimization

Because many practical signals are not exactly sparse, we study the reconstruction error of (4) applied to general signals, which is expressed in the best L–term approximation error of $\bar{x}$:

$$\sigma_L(\bar{x})_1 := \inf\{\|\bar{x} - x\|_1 : \|x\|_0 \le L, x \in \mathbb{R}^{\dim(\bar{x})}\}.$$

For a signal $\bar{x}$ with a fast decaying tail in term of distribution, this error is much smaller than $\|\bar{x}\|_1$. Theorem 3.3 below states that under certain conditions on A, (4) returns a solution with an ℓ_1-error bounded by $\sigma_L(\bar{x})$ up to a constant factor depending only on $|T|$. The theorem needs the following lemma, which is an extension of Lemma 4.2 in [13].

Lemma 3.2. *Consider problem* (4) *with a given* T*, and let* $z, z' \in \mathcal{F}(b)$*. Assume that* A *satisfies* t*-NSP*$(t, L, \bar{\gamma})$*, where* $t = |T|$ *and* $\bar{\gamma} < 1$*. Let* $S \subset T$ *be the set of indices corresponding to the largest* L *entries in* z_T*. We have*

$$\|(z - z')_{T\cap S^C}\|_1 \le \frac{1}{1-\bar{\gamma}} \left(\|z'_T\|_1 - \|z_T\|_1 + 2\sigma_L(z_T)_1\right), \tag{16}$$

where $\sigma_L(z_T)_1$ *is the best* L*-term approximation error of* z_T*.*

Proof. We have $\|z_{T\cap S^C}\|_1 = \sigma_L(z_T)_1$ and

$$\begin{aligned}
\|(z' - z)_{T\cap S^C}\|_1 &\le \|z'_{T\cap S^C}\|_1 + \|z_{T\cap S^C}\|_1 \\
&= \|z'_T\|_1 - \|z'_S\|_1 + \sigma_L(z_T)_1 \\
&= \|z_T\|_1 + \|z'_T\|_1 - \|z_T\|_1 - \|z'_S\|_1 + \sigma_L(z_T)_1 \\
&= \|z_S\|_1 - \|z'_S\|_1 + \|z'_T\|_1 - \|z_T\|_1 + 2\sigma_L(z_T)_1 \\
&\le \|(z - z')_S\|_1 + \|z'_T\|_1 - \|z_T\|_1 + 2\sigma_L(z_T)_1.
\end{aligned}$$

Equation (16) follows from the above inequality and the definition of t-NSP$(t, L, \bar{\gamma})$, which says

$$\|(z' - z)_S\|_1 \le \bar{\gamma}(\|(z' - z)_{T\cap S^C}\|_1). \tag{17}$$

□

Lemma 3.2 leads to the following theorem.

Theorem 3.3. *Consider problem* (4) *for a given* T*. Assume that* A *satisfies* t*-NSP*$(t, L, \bar{\gamma})$*, where* $t = |T|$ *and* $\bar{\gamma} < 1$*. Let* x^* *be the solution of* (4) *and* $\bar{x}$ *be the true signal. Then,* $\|x^*_T\|_1 - \|\bar{x}_T\|_1 \le 0$*, and*

$$\|x^* - \bar{x}\|_1 \le 2C_T \cdot \sigma_L(\bar{x}_T)_1, \tag{18}$$

where

$$C_T = \frac{1 + (1 + \max\{1, |T^C|/L\})\bar{\gamma}}{1 - \bar{\gamma}}.$$

Proof. For notation cleanliness, we introduce

$$S_1 := T^C = \mathcal{I},\ S_2 := S \subset T,\ S_3 := T \cap S^C.$$

Clearly, (S_1, S_2, S_3) is a partition of $\{1, \ldots, n\}$.

Case 1: $|S_1| \leq L$. We can find $S' \subset S_2$ such that $|S_1 \cup S'| = L$. From t-NSP$(t, L, \bar{\gamma})$ for A, we get

$$\|(z - z')_{S_1}\|_1 \leq \|(z - z')_{S_1 \cup S'}\|_1 \leq \bar{\gamma}\|(z - z')_{S_3}\|_1. \tag{19}$$

Case 2: $|S_1| > L$. Let $S'' \subset S_1$ denote the set of indices corresponding to the largest L entries of $(z - z')_{S_1}$. From t-NSP$(t, L, \bar{\gamma})$ for A,, we have

$$\|(z - z')_{S_1}\|_1 \leq \frac{|S_1|}{L}\|(z - z')_{S''}\|_1 \leq \frac{|S_1| \cdot \bar{\gamma}}{L}\|(z - z')_{S_3}\|_1 \tag{20}$$

Combining (19) and (20) gives

$$\|(z - z')_{S_1}\|_1 \leq \max\left\{1, \frac{|S_1|}{L}\right\}\bar{\gamma}\|(z - z')_{S_3}\|_1.$$

This, together with (16) and (17), gives

$$\begin{aligned} \|z - z'\|_1 &= \|(z - z')_{S_1}\|_1 + \|(z - z')_{S_2}\|_1 + \|(z - z')_{S_3}\|_1 && (21)\\ &\leq (1 + (1 + \max\{1, |S_1|/L\})\,\bar{\gamma})\,\|(z - z')_{S_3}\|_1 && (22)\\ &\leq C_T\left(\|z'_T\|_1 - \|z_T\|_1 + 2\sigma_L(z_T)_1\right). && (23) \end{aligned}$$

Finally, let z and z' denote the true signal $\bar{x}$ and the solution x^* of (4), respectively. The optimality of x^* gives $\|x^*_T\|_1 - \|\bar{x}_T\|_1 \leq 0$, from which (18) follows. □

Theorem 3.3, states that the reconstruction error of (4) is bounded by the best L-term approximation error of $\bar{x}_T$ up to a multiple depending on $\bar{\gamma}$ and $|T^C|$. When $t = n$, it reduces to the existing result for BP established in [11]:

$$\|x^* - \bar{x}\|_1 \leq 2\frac{1+\gamma}{1-\gamma} \cdot \sigma_{L'}(\bar{x})_1, \tag{24}$$

when A satisfies the NSP of order L' for $\gamma \in (0, 1)$.

To compare the two bounds in (18) and (24), we need to study the tail of $(\bar{x})_i$ sorted by magnitude. We claim that making correct detections alone, namely, having all or most in T^C being true nonzeros of $\bar{x}$, is not sufficient for (18) to be better than (24). To see this, assume that $\bar{\gamma} = \gamma$ in both bounds. Support detection makes $t = |T| < n$ and $|T^C| > 0$. As discussed at the end of Subsection 3.1 above, we get $L' \leq L$ so it is unclear whether $\sigma_L(\bar{x}_T)_1 < \sigma_{L'}(\bar{x})_1$ or vice versa. In addition, C_T is bigger than $(1 + \gamma)/(1 - \gamma)$. Hence, the bound in (18) can be bigger than (24). Therefore, only if the tail decays fast enough in the sense that $\sigma_L(\bar{x}_T)_1 \ll \sigma_{L'}(\bar{x})_1$ can (18) give an improved bound over (24). This comparison also applies to two instances of (4), one having a bigger T than the other; in other words, for support detection to reduce the reconstruction error, there need to be enough correct detections and $\bar{x}$ must have a fast decaying distribution of nonzeros. This conclusion matches our numerical results given in Section 5 below.

3.4 Iterative Behavior of ISD

The results in the above two subsections concern signal reconstruction by a single instance of (4), and they require A to have the t-NSP with $\bar{\gamma} < 1$. This subsection presents a sufficient condition for the iteration of ISD to yield a decreasing sequence of γ, a preliminary convergence result of ISD.

Theorem 3.4. *Suppose that A has the t-NSP$(t, L, \bar{\gamma})$ as well as $(t', L', \bar{\gamma}')$ with $t' < t$ and $L' < L$. If $(L - L') > \bar{\gamma}(t - t' - (L - L'))$, then $\bar{\gamma}' < \bar{\gamma}$.*

Proof. Let $0 < J' < J$ and $1 < \gamma < \infty$. For given T', η', and S' that satisfy $T' \subset \{1, \ldots, n\}$, $|T'| = t'$, $\eta' \in \mathcal{N}(A)$, $\eta' \neq \mathbf{0}$, $S' \subset T'$, $|S'| = J'$, $\bar{\gamma}' = \|\eta'_{S'}\|_1 / \|\eta'_{T' \setminus S'}\|_1$, we have

$$\begin{aligned} \|\eta'_{T' \setminus S'}\|_1 &= \|\eta'_{T \setminus S}\|_1 - \|\eta'_{T \setminus S - (T' \setminus S')}\|_1 \\ &\geq \bar{\gamma}^{-1} \|\eta'_S\|_1 - \|\eta'_{T \setminus S - (T \setminus S')}\|_1 \\ &= \bar{\gamma}^{-1} \|\eta'_{S'}\|_1 + \bar{\gamma}^{-1} \|\eta'_{S - S'}\|_1 - \|\eta'_{T \setminus S - (T \setminus S')}\|_1, \end{aligned}$$

for any S satisfying $S \supseteq S'$, $|S| = J$, and $S \subset T$, $|T| = t$, $S - S' \subseteq T \setminus T'$. In particular, we choose S such that $S - S'$ consists of the largest $J - J'$ entries of $\eta'_{T \setminus T'}$ in magnitude.

According to $(J - J') > \bar{\gamma}(t - t' - (J - J'))$, we have

$$|S - S'| > \bar{\gamma} |T \setminus S - (T' \setminus S')|. \tag{25}$$

If $\eta'_{S - S'} \neq \mathbf{0}$, then this condition means $\bar{\gamma}^{-1} \|\eta'_{S - S'}\|_1 > \|\eta'_{T \setminus S - (T' \setminus S')}\|_1$ and, thus, $\|\eta'_{T' \setminus S'}\|_1 > \bar{\gamma}^{-1} \|\eta'_{S'}\|_1$. Otherwise, i.e., $\eta'_{S - S'} = \mathbf{0}$, then we have $\bar{\gamma}^{-1} \|\eta'_{S - S'}\|_1 = \|\eta'_{T \setminus S - (T' \setminus S')}\|_1 = 0$. However, we can still show $\|\eta'_{T' \setminus S'}\|_1 > \bar{\gamma}^{-1} \|\eta'_{S'}\|_1$ by showing $\|\eta'_{T \setminus S}\|_1 > \bar{\gamma}^{-1} \|\eta'_S\|_1$, i.e., the first inequality in the equation array above holds strictly. To see this, we first get $\eta'_{T' \setminus S'} \neq \mathbf{0}$ from $\bar{\gamma}^{-1} \|\eta'_{S - S'}\|_1 = \|\eta'_{T \setminus S - (T' \setminus S')}\|_1 = 0$, $\|\eta'_{T' \setminus S'}\|_1 \geq \bar{\gamma}^{-1} \|\eta'_{S'}\|_1$, $\bar{\gamma}^{-1} > 0$, and $\eta' \neq \mathbf{0}$. Next, we generate $\bar{S}$ by first letting it be S, second dropping any one entry in $S - S'$ (which has a zero value), and picking up a nonzero entry in $T' \setminus S'$. Such $\bar{S}$ satisfies

$$\|\eta'_{T \setminus S}\|_1 > \|\eta'_{T \setminus \bar{S}}\|_1 \geq \bar{\gamma}^{-1} \|\eta'_{\bar{S}}\|_1 > \bar{\gamma}^{-1} \|\eta'_S\|_1.$$

Therefore, we have

$$(25) \Rightarrow \|\eta'_{T' \setminus S'}\|_1 > \bar{\gamma}^{-1} \|\eta'_{S'}\|_1$$

Therefore $\bar{\gamma}' < \bar{\gamma}$. □

In ISD, at the beginning of iteration $s = 0$, we have $t = n$ and $L = \|\bar{x}\|_0$. If A satisfies t-NSP$(t, L, \bar{\gamma})$ for $\bar{\gamma} < 1$, then the solution $x^{(0)}$ of BP is equal to $\bar{x}$ according to Theorem 3.1, and ISD terminates. Otherwise, $x^{(0)}$ is not equal to $\bar{x}$ and thus subject to support detection. Recall that the numbers of correct and wrong detections are denoted by d_c and d_w, respectively. Let $t' = n - d_c - d_w$ and $L' = \|\bar{x}\|_0 - d_c$. Theorem 3.4 states that if $(L - L') > \bar{\gamma}(t - t' - (L - L'))$, then the new $\bar{\gamma}'$ is smaller than $\bar{\gamma}$. If $\bar{\gamma}' < 1$, then the solution $x^{(1)}$ is equal to $\bar{x}$ and ISD terminates. Otherwise, the iteration continues. Here, the condition of $(L - L') > \bar{\gamma}(t - t' - (L - L'))$ is identical to $d_c > \bar{\gamma} d_w$, which means that if the number of correct detections is at least $\bar{\gamma}$ times larger than that of the false detections, then the t-NSP parameter $\bar{\gamma}$ reduces, so the t-NSP based reconstruction condition in Theorem 3.1 becomes either satisfied (if $\bar{\gamma}' < 1$) or closer to be satisfied. Clearly, this analysis can be applied, not just to iterations $s = 0, 1$, but to any two consecutive iterations of ISD. Theorem 3.4 also means that ISD is robust to a small number d_w of wrong detections as long as the number d_c of correct detections is larger than d_w by a certain amount.

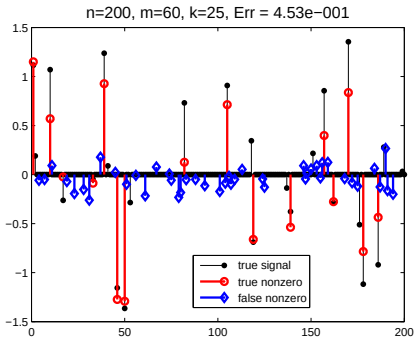
4 Threshold–ISD for Fast Decaying Signals

4.1 A Support Detection Scheme Based on Thresholding

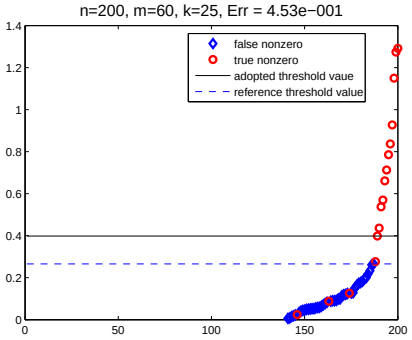
For ISD, successful signal reconstruction requires detecting true nonzero entries and keeping the number of wrong detections small enough at Step 2(c). We discovered an effective support detection strategy for signals with a fast decaying distribution of nonzero values (hereafter, we call them *fast decaying signals*) including sparse Gaussian signals and some sparse or compressible power–law decaying signals. It is based on thresholding the magnitudes of solutions in the form of

$$I^{(s+1)} := \{i : |x_i^{(s)}| > \epsilon^{(s)}\}, \quad s = 0, 1, 2, \ldots, \tag{26}$$

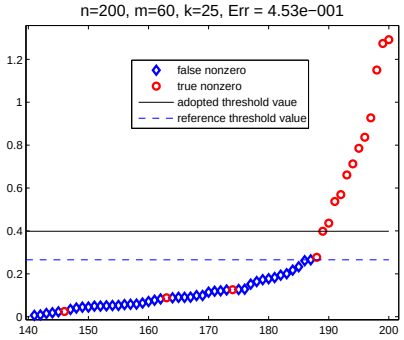
using certain thresholds $\epsilon^{(s)}$ for Step 2(c). We call such version *threshold–ISD*. Before discussing the choice of $\epsilon^{(k)}$, we note that the support sets $I^{(s)}$ are not necessarily increasing and nested, i.e., $I^{(s)} \subset I^{(s+1)}$ may



(a) Failed reconstruction



(b) Threshold value based on jump detection



(c) Zoomed-in Figure (b)

Figure 2: $x^{(0)}$ and the true signal are plotted to illustrate the adopted empirical support detection scheme. The sensing matrix is Gaussian. The true signal is sparse with Gaussian nonzeros. "Err" is the relative error in ℓ_2 norm.

not hold for all s, which is important due to the following reasons. It is very difficult to completely avoid wrong detections by setting $\epsilon^{(s)}$ based on available immediate solutions $x^{(i)}$, for $i \leq s$, simply because the true solution is not known and a component of $x^{(i)}$, no matter how big, is possibly a true zero. The non-monotonic property (26) leaves threshold–ISD with a chance to remove previous wrong detections, making threshold–ISD less sensitive to $\epsilon^{(s)}$ and making $\epsilon^{(s)}$ easier to choose.

We studied different heuristics for $\epsilon^{(s)}$. First, the rule $\epsilon^{(s)} = \|x^{(s)}\|_\infty/(\prod_{i=1}^{s} q_i)$ with a sequence of $q_i > 0$ (e.g., in (5) $q_i = 5, \forall i$) is often, yet not always, effective because the parameter q_i is case-dependent and difficult to determine. An excessively large q_i likely results in too many false detections and consequently worse solutions whilst an excessively small q_i tends to cause a large number of threshold–ISD iterations. Because of the existence of a better heuristic, this rule is not recommended.

The second rule we have studied, which is not recommended either, is a toll–based rule, namely, setting $\epsilon^{(s)}$ so that $I^{(s)}$ has a given cardinality, typically increasing over iterations. It is no easier to set an appropriate cardinality than to determine appropriate $\epsilon^{(s)}$. We tried equally spaced cardinality sequences, such as $1, 2, \ldots$ and $2, 4, \ldots$, and found that the resulting threshold–ISD implementations were either slower or required more measurements for successful reconstruction, or sometimes both.

For the threshold $\epsilon^{(s)}$, the rule of our choice is based on locating the "first significant jump" in the sequence of increasingly sorted $|x_i^{(s)}|$, as illustrated in Figures 2. Assume that $x^{(s)}$ is not equal to the true signal $\bar{x}$ and $x^{(s)}$ has m nonzero entries[2] corresponding to a certain nonsingular basic submatrix B of A. Slightly abusing the notation, we let B also denote the set of basic indices, i.e., $x_i^{(s)} \neq 0$ if and only if $i \in B$. Furthermore, we divide B into disjoint subsets U and V such that $B = U \cup V$, where U consists of true nonzero indices (i.e, $U \subset \bar{T}$) and V contains the rest, and we have $|U| < k < |V|$. For fast decaying signals, we argue that $|x_i^{(s)}|$, $i \in U$, tend to have relatively large magnitudes than those in V, and most components in U are also among the larger ones in $x_{\bar{T}}$, the set of all true nonzeros; in other words, the larger components of $x^{(s)}$ are likely the large, true nonzero components (though their values can be inaccurate). This can be seen in the first–iteration solution $x^{(0)}$ of the demo plotted in Figures 2, where the sub–figures (a) and (b) plot the components of $x^{(0)}$ in their original and sorted positions, respectively. The components in U and V are highlighted as red circles $\circ$ and blue diamonds $\diamond$, respectively, and they are roughly separated in sub–figures (b) and (c).

The rough separation of U from V follows intuitively from the sensitivity results of CS: the reconstruction error is bounded by the error in the sparse signal (also see Theorem 3.3). For fast decaying signals, one can treat an enough number of small true nonzero components as the signal error and thus have an even sparser non-error part of the signal consisting of the remaining large nonzero components. The reconstruction should therefore contain these large components plus the reconstruction error (i.e., "smearing") caused by the signal error. Mathematically, $x_B^{(s)}$ can be decomposed into the sum of two parts: the true part and the smearing part. The true part is simply $\bar{x}_B = [\bar{x}_U; \mathbf{0}]$; the smearing part is $\varpi := B^{-1} \sum_{i \in \bar{T} \setminus U} A_i \bar{x}_i$, the projection of the part of the measurements b contributed by the nonzeros of $\bar{x}$ missing in $x^{(s)}$, over B. We have $x_B^{(s)} = \bar{x}_B + \varpi$ (notice that ϖ_i is likely nonzero for all $i \in B$, so it smears both U and V). Because the projection over the wider m-by-m matrix B has a diluting effect[3], the components of ϖ tend to be smaller than their generators $\bar{x}_i$ for $i \in \bar{T} \setminus U$; furthermore, because the components of $\bar{x}_{\bar{T}\setminus U}$ tend to be smaller than those in $\bar{x}_U$, the components of ϖ are in turn further smaller than those in $\bar{x}_U$, or writing in a simple form, $\varpi \lessapprox \bar{x}_{\bar{T}\setminus U} \lessapprox \bar{x}_U$. Therefore, the large components of $x_B^{(k)} = \bar{x}_B + \varpi$ likely belong to U, the middle (smaller ones) belong to V and U, and the smallest ones belong to V only.

To tell the large components of $x_B^{(k)}$ (thus, a subset of U) from the rest, observe that the large components are small in number whilst the smaller ones are large in number. When going through the sorted sequence from small to large, the part containing the smaller values increases slowly and the part containing the larger ones increases sharply. Therefore, we propose to detect the large components of $x_B^{(k)}$ by finding the first significant jump, its position denoted by i^*, in the increasingly sorted sequence $\{|x_i^{(s)}|\}$. Given i^*, we set

[2](4) can be reduced to a linear program. It is possible that $x^{(k)}$ has either less or more than m nonzero entries but for most matrices A used in CS and most signals, $x^{(s)}$ has exactly m nonzero entries with a nonsingular basis B. This assumption offers technical convenience, and is not essential to the argument that follows.

[3]This diluting property follows sufficiently from the RIP of A but not necessarily. Because the solution of (4) depends only on the affine space $\{x : Ax = b\}$, which is equivalent to $\{x : RAx = Rb\}$ for any nonsingular matrix $R^{\geqslant \times \geqslant}$, the property depends on the property of the null space of A.

the threshold value $\epsilon^{(s)} = |x^{(s)}_{i^*+1}|$ as indicated by the solid line in Figure 2. This sets $I^{(s+1)}$ to consist of the largest $(n - i^*)$ components of $x^{(s)}$ in magnitude.

Here we adopt a very simple method to find i^* by finding the smallest i such that $|x^{(s)}_{i+1}| - |x^{(s)}_i| > \tau^{(s)}$, where the size of $\tau^{(s)}$ may vary for different kinds of sparse or compressible signals. For the sparse Gaussian signals, we set $\tau^{(s)}$ simply as $m^{-1}\|x^{(s)}\|_\infty$, which is the average gap size. In the example shown in Figure 2 where the true signal is sparse Gaussian signal, this threshold of "first significant jump" incurs no wrong detection. For comparison, we also plotted the lowest possible threshold giving no wrong detection (called the reference threshold) in the dashed line (which is unknown in practice). For the first 5 iterations in the experiments reported in Section 5 below, we made the detection slightly more conservative by increasing $\tau^{(s)}$ to $\frac{6}{s+1}\tau^{(s)}, s = 0, 1, 2, 3, 4$.

Besides sparse Gaussian signals, we also tried the "first significant jump" rule on synthetic sparse and compressible power–law decaying signals, as well as the wavelets coefficients of the Shepp-Logan phantom image and the cameraman image in Section 5. The sparse and compressible power–law decaying signals were constructed by first generating a sequence of numbers obeying the power-decay rule like $\{i^{-1/\lambda}\}_{i=1}^k$ followed by multiplying each entry by a random sign and applying a random permutation to the signed sequence. We set $\tau^{(s)}$ according to λ. For example, when $\lambda = 1/3$, one can set $\tau^{(s)} = \|x^{(s)}\|_\infty/m/20$; when $\lambda = 0.8$, or 1, one can set $\tau^{(s)} = \|x^{(s)}\|_\infty/m/5$; when $\lambda = 2$ or 4, one can set $\tau^{(s)} = \|x^{(s)}\|_\infty/m/2$. For the wavelets coefficients of the phantom image and cameraman image, one can set $\tau^{(s)} = \|x^{(s)}\|_\infty/m$. Like we did for sparse Gaussian signals, for the first 9 iterations in the experiments, we made the detection slightly more conservative by increasing $\tau^{(s)}$ to $\frac{8}{s+1}\tau^{(s)}, s = 0, 1, 2, 3, 4, 5, 6, 7$. The above heuristics, which worked very well in our experiments, may not be optimal or even nearly optimal, but it has been observed that the performance of threshold–ISD was not very sensitive to the setting of $\tau^{(s)}$.

Finally, we give three comments on support detections. First, the "first significant jump" rule is overall effective. However, to avoid the parameter $\tau^{(s)}$ used in the rule, one can apply other available effective jump detection methods [31, 25]. Second, any threshold-based support detection rule requires true signals to have a fast decaying distribution of nonzeros in order to work reliably. While most signals of interest are such signals, there do exist signals that decay slowly or have no decay at all (e.g., sparse Bernoulli signals). For sparse Bernoulli signals, we tried different kinds of thresholding strategies, and all failed to make ISD significantly better than BP. Third, one should be able to find better support detection methods for real world signals. Examples of such signals include those with grouped nonzero components (cf. the model-based CS [2]) and natural/medical images in which the set of nonzero components (often as edges) has a certain geometry (e.g., each object boundary is continuous; cf. CS MRI with edge detections [23]).

4.2 YALL1 and Warm-Start

Threshold–ISD obtains solutions from a sequence of truncated ℓ_1 minimization problems (4), which can be solved by most existing ℓ_1 algorithms/solvers with straightforward modifications. We chose YALL1 [33] since it is among the fastest ones at solving ℓ_1 problems whether their solutions are sparse or not. In the numerical tests reported in Section 5 below, we also applied YALL1 to all plain (unweighed) and weighted ℓ_1 minimization problems used in the compared algorithms.

Consulting YALL1's manual [39], here we give a short introduction to YALL1. Based upon applying the alternating direction method to the Lagrange dual of

$$\min_x \{\sum_{i=1}^n w_i|x_i| : Ax = b\},$$

the iteration in YALL1 version beta–3 has the basic form:

$$y^{l+1} = \alpha A z^l - \beta(Ax^l - b), \tag{27}$$

$$z^{l+1} = P_w(A^* y^{l+1} + x^l/\mu), \tag{28}$$

$$x^{l+1} = x^l + \gamma\mu(A^* y^{l+1} - z^{l+1}), \tag{29}$$

where $\mu > 0$, $\gamma \in (0, (1 + \sqrt{5})/2)$, $\alpha = 1$ and $\beta = \frac{1}{\mu}$. P_w is an orthogonal projection onto the box $\mathbf{B}_w \triangleq \{z \in \mathbb{C}^n : |z_i| \le w_i, i = 1, \ldots, n\}$. This is a first-order primal–dual algorithm in which the primal

variables x and dual variables y and z (a dual slack) are updated at every iteration. We stopped YALL1 iterations once the relative change between two sequent x^l and x^{l+1} in 2-norm

$$\|x^{l+1} - x^l\|_2 / \|x^l\|_2$$

falls below a certain prescribed stopping tolerance.

YALL1 is capable of returning a highly accurate solution. However, because each $I^{(s+1)}$ was determined by the "first significant jump" rule in our experiments, it was useless to have a highly accurate solution $x^{(s)}$ just for determining $I^{(s+1)}$. Therefore, in all but the last threshold–ISD iteration, a loose stopping tolerance was set in YALL1, which is described in Subsection 5.5 below. To determine whether a threshold–ISD iteration k is the final iteration, the loose stopping tolerance was put in place to stop YALL1 and upon stopping of YALL1, $I^{(s+1)}$ was generated and compared to $I^{(s)}$ and $I^{(s-1)}$. If they were exactly or almost the same, solving another (4) based on $I^{(s+1)}$ would likely give the same or a very similar solution, so another iteration was not needed. At this time, we did not return the current $x^{(s)}$ but let YALL1 resume from where it stopped, but now with a tighter final stopping tolerance to return an accurate final solution. If $I^{(s+1)}$ was significantly different from either $I^{(s)}$ or $I^{(s-1)}$, threshold–ISD was continued with a new iteration $s+1$.

We note that based on empirical experience, one can introduce additional rules to stop the subproblem solver YALL1. For example, if measurements have no noise and the true signal is known to have no more than k nonzero elements, then YALL1 can be stopped once $\|x^l\|_0$, the number of nonzero elements in its current point, reaches $\approx 2k$. This is because $\|x^l\|_0$ typically grows in l.

To further accelerate threshold–ISD, we warm-started YALL1 whenever possible. Specifically, for each instance of (2), (4), and reweighted ℓ_1 subproblems, YALL1 was started from the solution (x, y, z) of the previous instance if available.

As a result of applying varying stopping tolerance and warm–start to YALL1, the total time for threshold–ISD to obtain a final solution, including the time of solving multiple instances of (4) altogether, was almost the same on average as the time of solving a single BP problem to the final accuracy[4].

We advise the reader that whether ISD is used with a thresholding rule or not, appropriate subproblem accuracies and effective warm-staring strategies should always be taken into account.

5 Numerical Implementation and Experiments

Threshold–ISD was compared to the BP model (2), the iterative reweighted least-squares algorithm (IRLS) [10], and the iterative reweighted ℓ_1 minimization algorithm (IRL1) [9]. IRLS and IRL1 appear to be state-of-the-art in term of the number of measurements required. The comparisons show that threshold–ISD requires as few measurements as IRLS or IRL1 meanwhile threshold–ISD can be computed as efficiently as the BP model.

5.1 Reviews of IRL1 and IRLS

Let us first briefly review the algorithms IRL1 [9] and IRLS [10]. They are both iterative procedures for approximately solving the following ℓ_p minimization problem:

$$\min \|x\|_p \quad \text{s.t.} \quad Ax = b \tag{30}$$

where $p \in [0, 1]$. At the s-th iteration, the IRL1 algorithm computes

$$x^{(s)} \leftarrow \min_x \{\sum_{i=1}^{n} w_i^{(s)} |x_i| : Ax = b\}, \tag{31}$$

where the weights are set as

$$w_i^{(s)} := (|x_i^{(s-1)} + \eta|)^{p-1}, \tag{32}$$

and η is a regularization parameter. Initially, $x^{(0)}$ is the solution of the BP problem.

[4]Recall that in the first threshold–ISD iteration, YALL1 solves a BP problem not with a tight final but a loose stopping tolerance.

IRLS iteratively minimizes a weighted ℓ_2 function to generate $x^{(s)}$:

$$x^{(s)} \leftarrow \min_x \{\sum_i \tilde{w}_i^{(s)} |x_i|^2 : Ax = b\}, \tag{33}$$

The solution of (33) can be given explicitly as

$$x^{(s)} = Q_s A^{\mathrm{T}} (A Q_s A^{\mathrm{T}})^{-1} b \tag{34}$$

where Q_s is the diagonal matrix with entries $1/\tilde{w}_i^{(s)}$ and the weights are set as

$$\tilde{w}_i^{(s)} := (|x_i^{(s-1)}|^2 + \zeta)^{p/2-1} \tag{35}$$

and ζ is a regularization parameter. Initially, $x^{(0)}$ is the least-squares solution of $Ax = b$.

We set $p := 0$ uniformly in the experiments since it is reported that this value leads to better reconstructions than $p > 0$. Notice that if $x^{(s-1)}$ in both (32) and (35) are set equal to x and $\eta = \zeta = 0$, then following the convention $0/0 = 0$, we have $\sum_i w_i^{(s)} |x_i|^2 = \sum_i \tilde{w}_i^{(s)} |x_i|^2 = \|\bar{x}\|_0$. This result, though not holding for $\eta, \zeta > 0$, indicates that the two objective functions are smooth approximations to $\|x\|_0$. When η and ζ are large, $\sum_{i=1}^n w_i^{(s)} |x_i|$ and $\sum_i \tilde{w}_i^{(s)} |x_i|^2$ are close to $\|x\|_1$ and $\|x\|_2^2$, respectively, so they tend to have fewer local minima. Therefore, η and ζ were both initially large and gradually reduced as s increased. The setting of these parameters were given in Subsection 5.4 below.

5.2 Denoising Minimization Problems

The measured data is sometimes inaccurate due to various kinds of imprecisions or contaminations. Assume that $b = Ax + z$, where z is i.i.d. Gaussian with zero mean and standard deviation (noise level) σ. When σ is not big, the unconstrained BP problem (3) is known to yield a faithful reconstruction for an appropriate ρ depending on σ. We found that (3) could be solved faster and yield a slightly more accurate solution than the constrained BP problem (2). Therefore, (3) was used in our tests with noisy measurements. We still use the name BP to refer to the algorithm that solves one instance of (3). In our figures, we use "L1/L2" for (3).

For IRL1, reweighting was applied to the above noise–aware problem (3), and each of its iteration was changed from (31) to

$$x^{(s)} \leftarrow \min_x \{\sum_{i=1}^n w_i^{(s)} |x_i| + \frac{1}{2\rho} \|b - Ax\|_2^2\}, \tag{36}$$

where weights $w_i^{(s)}$ were generated as before. Given an index set T, threshold–ISD solved the following truncated ℓ_1 version of (3):

$$x^{(s)} \leftarrow \min \|x_{T^{(s)}}\|_1 + \frac{1}{2\rho} \|b - Ax\|_2^2. \tag{37}$$

The same ρ was set for the three problems (3), (36) and (37), which were all solved by YALL1 iterations (27), (28) and (29) with new $\alpha := \frac{\mu}{\mu+\rho}$ and $\beta := \frac{1}{\mu+\rho}$.

For IRLS, however, we did not relax $Ax = b$ in (33) for noisy measurements because the resulting unconstrained problem is no easier to solver and neither does it return solutions with less error, at least when the error level σ is not excessively large. Therefore, (33) was solved by IRLS for noisy measurements.

5.3 Transform Sparsity

In many situations, it is not the true signal $\bar{x}$ itself but its representation under a certain basis, frame, or dictionary that is sparse or compressible. In such a case, $\bar{y} = W\bar{x}$ is sparse or compressible for a certain linear transform W. For many signals and images, for example, W can be sinusoids, wavelets, curvelets, etc. Instead of minimizing $\|x\|_1$ and $\|x_T\|_1$, $\|Wx\|_1$ and $\|(Wx)_T\|_1$ should be minimized, respectively. Similarly, the weight variables in IRL1 and IRLS should be updated according to the components of (Wx) instead of those of x. In case of transform sparsity, the above simple changes were applied to all algorithms and solvers.

#	Nonzeros or Image Name	Noise σ	Dimension n	Sparsity k	Measurements m	Repetitions
1	Gaussian	0	600	8	16:4:100	100
	Gaussian	0	600	40	80:10:220	100
	Gaussian	0	600	150	250:10:400	100
2	Gaussian	0	3000	100	200:50:800	100
3	Gaussian	0.0001	2000	100	325	200
	Gaussian	0.001	2000	100	325	200
	Gaussian	0.01	2000	100	325	200
4	Power–Law w/ Random Signs	0	600	40 or 600	varied	100
5	Shepp–Logan Phantom	0	128×128	1685	2359:337:6066	10
	Shepp–Logan Phantom	0.001	128×128	1685	2359:337:7414	10
	Cameraman	0	256×256	65536	6553:2621:32763	10

Table 1: Summary of test sets.

5.4 Experimental Settings and Test Platforms

The test sets are summarized in Table 1. Our experiment included five different test sets, the first four of which used various synthetic signals and standard i.i.d. Gaussian sensing matrices A generated by `A=randn(m,n)` in MATLAB. The first two sets used noise-free measurements, and different amounts of white noise was added to measurements in the third set. With noise, exact reconstruction was impossible, so in the third set we did not change m but measured solution errors for different levels of noise. While all the signals in the first three test sets were sparse with i.i.d. Gaussian nonzeros values assigned to randomly selected entries, the fourth set used signals with entries following power laws among which a part of the signals had their tails truncated and thus sparse. As zero–decay sparse signals are just sparse ± 1 signals, this set also included sparse Bernoulli signals. The last (fifth) set used two–dimensional images of different sizes and tested sensing matrices A that were partial discrete cosine matrices formed by choosing the first and a random subset of the remaining rows from the full discrete cosine matrices. Since all the operations in threshold–ISD and IRL1 involving A (which are Ax and $A^\top x$) were computed by the discrete cosine transform, A were never explicitly formed or stored in memory in these two algorithms. On the other hand, IRLS needs explicit matrices A so it was not tested in the fifth set[5]. The fifth set also included tests with noise added to the measurements. The images were assumed to be sparse under the two–dimensional Haar wavelets.

Specifically, the sparse Gaussian signals were generated in MATLAB by

```
xbar =zeros(n,1);  p=randperm(n); xbar(p(1:k))=randn(k,1);
```

and the sparse Bernoulli signals were generated by the same commands except

```
xbar(p(1:k))=2*(rand(k,1)>0.5)-1;
```

The power–law decaying signals were generated by

```
xbar=zeros(n,1); p=randperm(n);
xbar(p(1:n))=sign(randn(n,1)).*((1:n).^(-1/lambda))';
xbar=xbar/max(abs(xbar));
```

and replacing the second line by `xbar(p(1:k))=sign(randn(k,1)).*((1:k).^(-1/lambda))'` we obtained the sparse ones. Variable `lambda` was set to different values (described in Subsection 4.1 above), which controls the rate of decay. The larger `lambda`, the lower the rate of decay.

As required by YALL1 version beta-3[6], a linear transform was applied to both A and b so that the new matrix A had orthonormal rows. Obviously, this did not change the set of points x satisfying $Ax = b$ or any solutions of minimization problems subject to $Ax = b$.

All test code was written and tested in MATLAB v7.7.0 running in GNU/Linux Release 2.6.9–55.0.2 on a Dell Optiplex GX620 with dual Intel Pentium D CPUs 3.20GHz (only one CPU was used by MATLAB) and 3 GB of memory.

[5]IRLS iterations could be modified to use Ax and $A^\top x$ rather than A in the explicit form, but for the purpose of this paper, no modification was done.

[6]Orthonormalization is no longer required by YALL1 version beta-4 but still recommended.

5.5 Stopping Tolerance and Smoothing Parameters

Performances of all tested code depend on parameters. For fairness, threshold–ISD, IRL1, IRLS, and BP were stopped upon

$$\frac{\|x^{l+1} - x^l\|_2}{\|x^l\|_2} \leq \epsilon, \tag{38}$$

with different intermediate but the same final stopping tolerances ϵ for their solutions returned for comparison.

Test sets 1, 2, and 4: In all tests, threshold–ISD was set to run no more than 9 iterations (with the reason given below in test sets 1 and 2), in which the first iteration had $\epsilon := 10^{-1}$ and the rest except for the last one had $\epsilon := 10^{-2}$. Smaller ϵ values did not make solutions or running times better. $\epsilon := 10^{-6}$ was set uniformly for BP, all iterations of IRL1, and final iterations of threshold–ISD. Larger intermediate ϵ values would make IRL1 return worse solutions. IRL1 had 9 iterations as recommended in [9], but its smoothing parameter η was initialized to 1 and reduced by half each time, different from but slightly better than the recommendation.

Recommended in [10] for IRLS and for test sets 1 and 2, its smoothing parameter ζ was initialized to 1 and reduced to $\zeta/10$ whenever (38) was satisfied for $\epsilon := \sqrt{\zeta}/100$ until ζ reaches 10^{-8} when the final $\epsilon = 10^{-6}$ became effective. We optimized ϵ and the stopping value of ζ for the more challenging test 4. For power–law decaying signals (either sparse or not), $\epsilon := 10^{-3/2}\sqrt{\zeta}$ and the stopping ζ was set to 10^{-9}; for sparse Bernoulli signals, $\epsilon := 10^{-1}\sqrt{\zeta}$ and the stopping ζ was set to 10^{-10}. Again, ϵ reached 10^{-6} finally. The above optimization to IRLS made its frequency of successful reconstruction slightly higher than threshold–ISD in a couple of tests.

Test set 3: Because of measurement noise and thus reconstruction errors, it was not necessary to impose a tight tolerance for this set of tests. All the four algorithms had the reduced final $\epsilon := \sqrt{\sigma}/100$ uniformly. Threshold–ISD ran no more than 9 iterations with the same intermediate ϵ values as in test sets 1, 2, and 4. IRL1 ran 9 iterations with intermediate $\epsilon := \sqrt{\sigma}/100$ constantly. For all but the last IRLS iteration, $\epsilon := \max\{\sqrt{\sigma}/100, \sqrt{\zeta}/100\}$.

Test set 5: Threshold–ISD was only compared to BP and IRL1 because matrices A were too large to form in IRLS. The only parameter change was the final $\epsilon := \max\{10^{-4}, \sigma/10\}$ for all the three tested algorithms.

5.6 Experimental Results

Test set 1: Sparse signals containing $k = 40, 8, 150$ nonzeros were used in this set, and corresponding results were plotted in Figures 3/4, 5, and 6, respectively.

Figure 3 shows that threshold–ISD converged quickly in just a few iterations. To compute the frequencies of acceptable reconstructions, the condition $\|x - \bar{x}\|_\infty \leq 10^{-3}$ was used in this and remaining tests. Based on Figure 3 (as well as our other tests), we limited the threshold–ISD iterations to 9. We note that the cyan curve, corresponding to BP, drops as $m > 180$ simply because the loose stopping tolerance was applied. This drop does *not* mean that BP gets worse as m increases. With a tight tolerance, the cyan curve would flat out.

Figure 4 depicts the performance of the four tested algorithms. Figure 4(a) shows that threshold–ISD and IRLS achieved almost the same recoverability, which was significantly higher than that of IRL1 in term of the number of required measurements. With no surprise, the recoverability of the BP method was the worst. Figure 4(b) shows that threshold–ISD was much faster than both IRL1 and IRLS, and was even comparable to the BP method (thanks to flexible stopping tolerances and warm starts). To sum up, threshold–ISD was not only the fastest one but also the one that required the least number of measurements.

With the small $k = 8$ (Figure 5), threshold–ISD had no speed advantage over IRLS but it was still much faster than IRL1. Quality–wise, threshold–ISD was on par with IRLS and IRL1 and better than BP. With the larger $k = 150$, threshold–ISD was much faster than both IRLS and IRL1, and all three achieved comparable recoverability with 100% starting around $m = 300$.

Test set 2: This test set used larger signals ($n = 3000$). Figure 7 again shows that no more than 9 threshold–ISD iterations were necessary. Figure 8 shows that threshold–ISD, IRLS, and IRL1 achieved

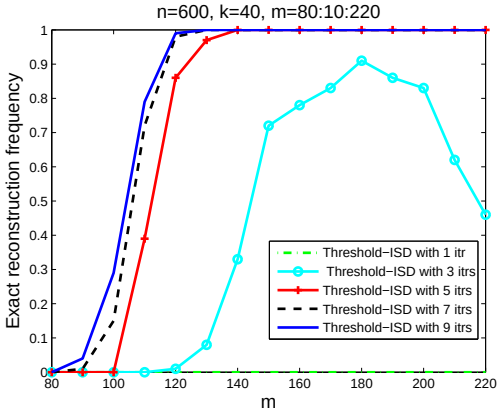
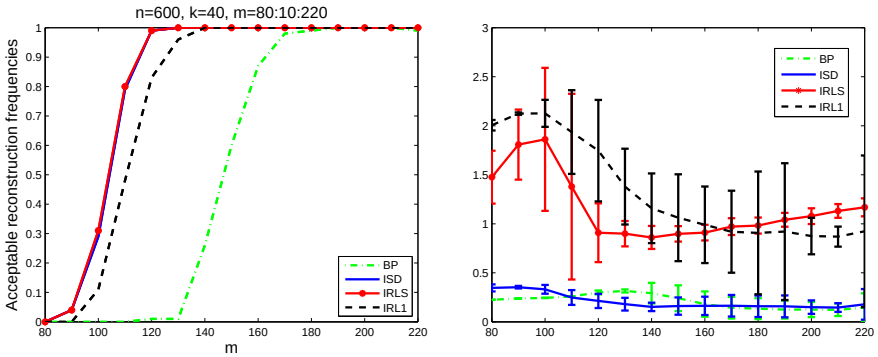
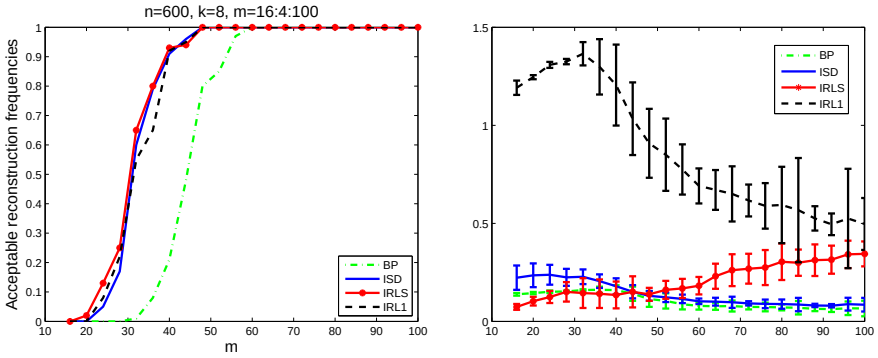


Figure 3: Test set 1 with $k = 40$: Acceptable reconstruction frequencies as a function of the number of measurements and the number of threshold–ISD iterations



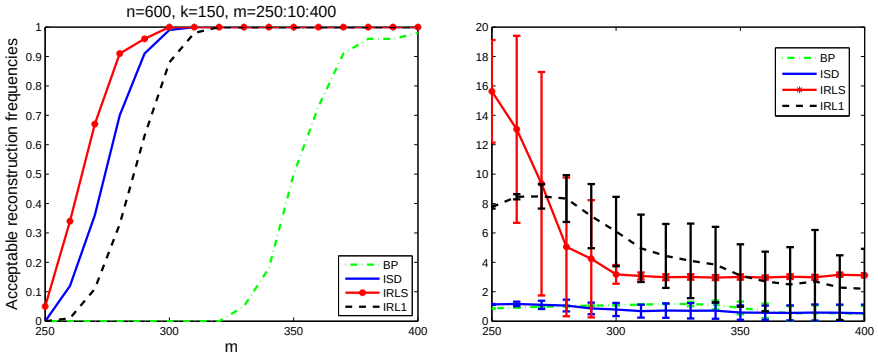
(a) Acceptable reconstruction frequencies (b) CPU time

Figure 4: Test set 1 with $k = 40$: Comparisons in recoverability and CPU time



(a) Acceptable reconstruction frequencies (b) CPU time

Figure 5: Test set 1 with $k = 8$: Comparisons in recoverability and CPU time



(a) Acceptable reconstruction frequencies (b) CPU time

Figure 6: Test set 1 with $k = 150$: Comparisons in recoverability and CPU time

similar recoverability as they did in test set 1. Because of the relatively large signal size, IRLS and IRL1 were however much slower than threshold–ISD. The fact that the latter, which involved solving 8 subproblems each time, ran as fast as BP, which solved merely one problem each time, suggests that effective support detection accelerates subproblem solution (by YALL1) in threshold–ISD. The results also show that threshold–ISD was scalable to both signal and measurement sizes.

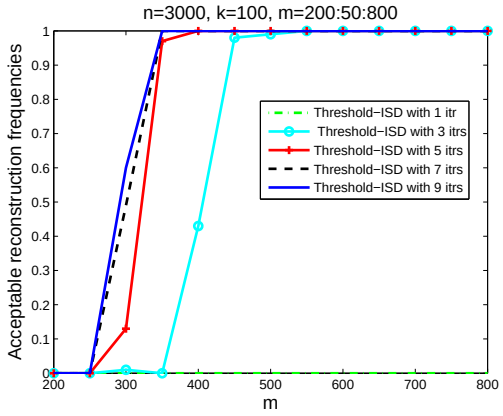
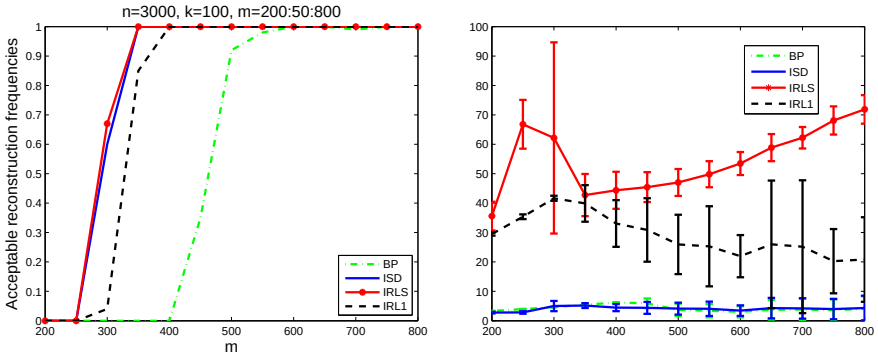


Figure 7: Test set 2: Acceptable reconstruction frequencies as a function of the number of measurements and the number of ISD iterations

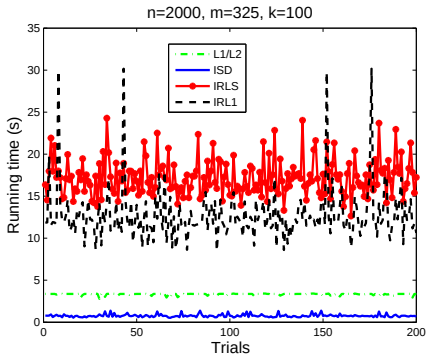
Test set 3: This test set compared solution times and errors of the tested algorithms given noisy measurements with three noise levels: $\sigma = 0.0001$, $\sigma = 0.001$ and $\sigma = 0.01$. The corresponding results are depicted in Figures 9, 10, and 11, respectively, each including three subplots for CPU times, ℓ_2 and ℓ_1 errors. It is clear from these figures that threshold–ISD was significantly faster than IRLS and IRL1 and slightly faster than BP. Threshold–ISD and IRLS were on par on solution quality except that at $\sigma = 0.01$, threshold–ISD had two fails among the total two hundred trials. IRL1 had much more fails, and BP was the worst. Taking both reconstruction errors and CPU times into consideration, threshold–ISD appeared to the best.

Test set 4: This test set included two subsets. The first subset (Figure 12) used sparse signals with nonzeros decaying in power–laws at rates $\lambda = 1, 2, 4$, as well as sparse Bernoulli signals. The second subset (Figure 13) used two compressible (non–sparse) signals with nonzeros decaying in power–laws at rates $\lambda = 1/3, 0.8$, as well as their sparse tail–removed versions obtained by removing all but the largest $k = 8, 40$ entries, respectively.

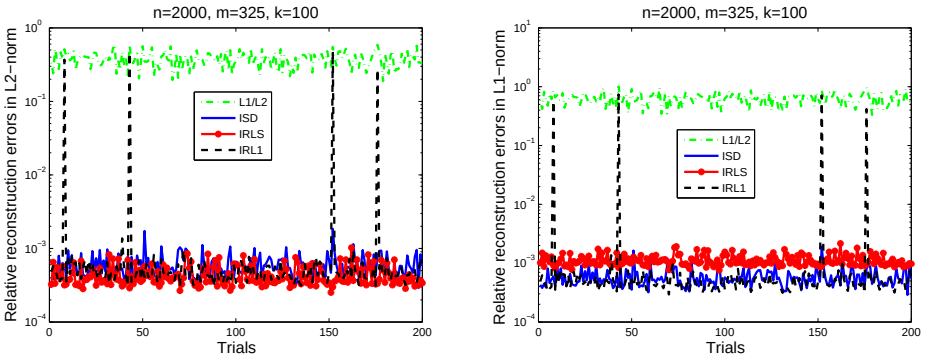


(a) Acceptable reconstruction frequencies (b) CPU time

Figure 8: Test set 2: Comparisons in recoverability and CPU time

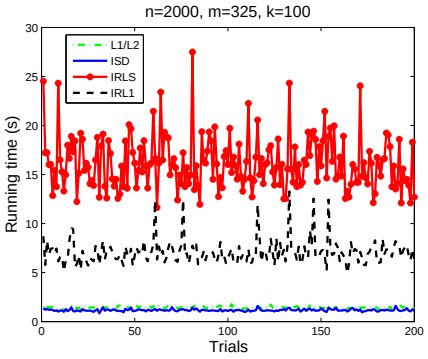


(a) CPU time

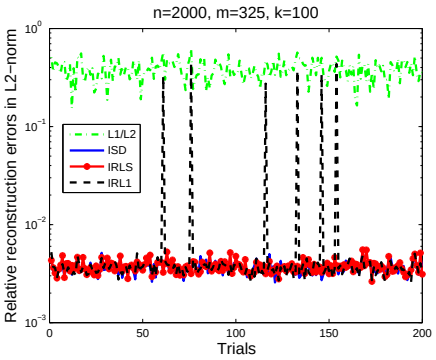


(b) Reconstruction errors in ℓ_2 norm (c) Reconstruction errors in ℓ_1 norm

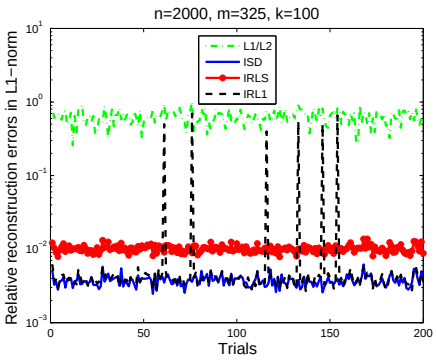
Figure 9: Test set 3 with $\sigma = 0.0001$: Comparisons in CPU time and reconstruction errors



(a) CPU time

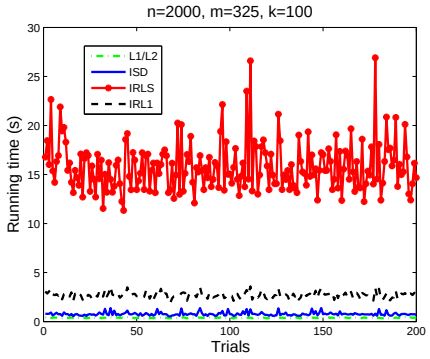


(b) Reconstruction errors in ℓ_2 norm

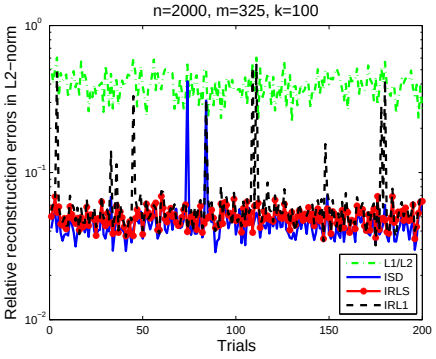


(c) Reconstruction errors in ℓ_1 norm

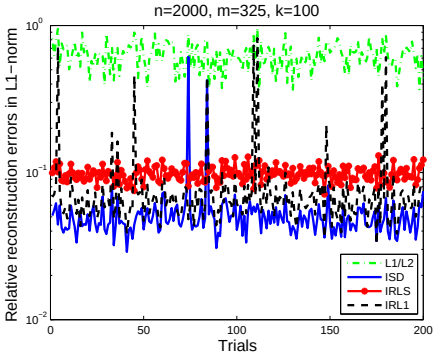
Figure 10: Test set 3 with $\sigma = 0.001$: Comparisons in CPU time and reconstruction errors



(a) CPU time



(b) Reconstruction errors in ℓ_2 norm

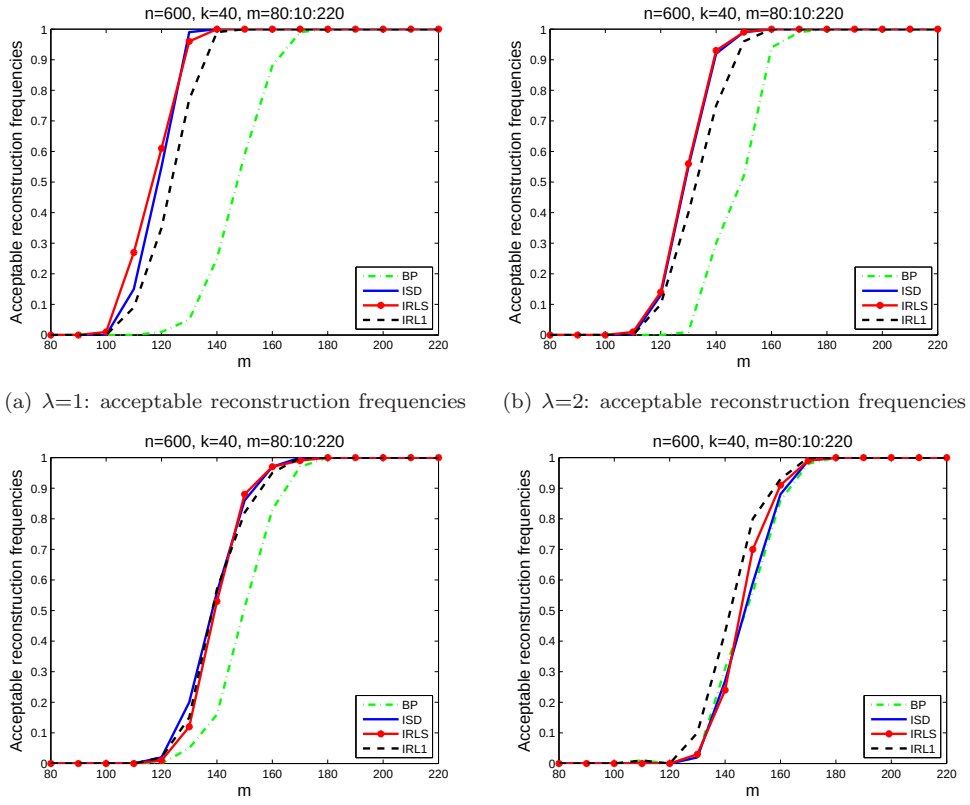


(c) Reconstruction errors in ℓ_1 norm

Figure 11: Test set 3 with $\sigma = 0.01$: Comparisons in CPU time and reconstruction errors

In the first subset of tests, threshold–ISD, IRL1, and IRLS had better recoverability than BP, but the advantage diminished as λ increased (i.e., the rate of decay decreased). In cases of zero–decay where the signals were sparse Bernoulli, all the tested algorithms had similar recoverability. These results match the conclusion in our theoretical analysis that thresholding–based support detection is effective only if the nonzeros of signals have a fast–decaying distribution.

The second subset of tests show how the tail of a compressible signal affects its reconstruction. By comparing Figures 13 (c) with (d) and (e) with (f) (i.e., recovering tail–removed v.s. tailed signals), we can observe that it was much easier for any of the tested algorithms to reconstruct a tail–free sparse signal than a compressible signal. In addition, because the signal corresponding to $\lambda = 0.8$ has a larger tail, the improvement of threshold–ISD, IRL1, and IRLS over BP was quite small.



(a) λ=1: acceptable reconstruction frequencies (b) λ=2: acceptable reconstruction frequencies

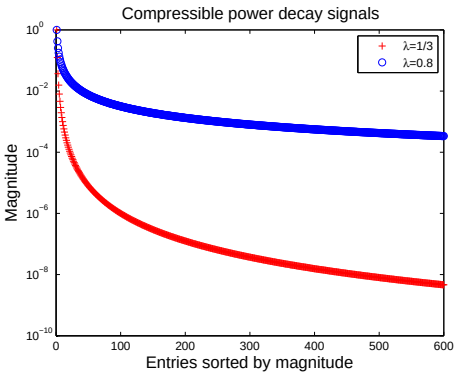
(c) λ=4: acceptable reconstruction frequencies (d) Bernoulli: acceptable reconstruction frequencies

Figure 12: Test set 4 with sparse power–law decaying and Bernoulli signals: Comparisons in recoverability

Test Set 5 : Figure 14 depicts the clean 128×128 Shepp–Logan phantom and its wavelets coefficients sorted by magnitude, which have a long tail and a sharp drop near 1600. Threshold–ISD, IRL1, and BP were tested to reconstruct the phantom from partial discrete cosine measurements both without and with white noise. The results are given in Figures 15 and 16, respectively. For Figure 15, a reconstruction $\tilde{x}$ was accepted if its 2–norm relative error $\|\tilde{x}(:) - \bar{x}(:)\|_2 / \|\bar{x}(:)\|_2$ was within 10^{-3}. Threshold–ISD was both faster than IRL1 and more accurate than IRL1 and BP in both of the tests.

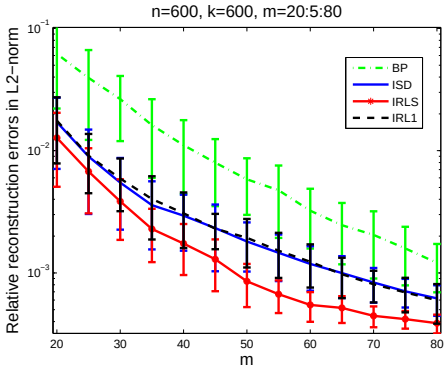
Figure 17 presents the reconstructed phantoms from noisy measurements corresponding to $m = 3370$, in which subplots (b), (d) and (f) highlight the differences between the reconstructions and the clean phantom. Threshold–ISD had a much higher signal–to–noise (SNR) ratio.

Figure 18 depicts the clean image "Cameraman" and its sorted wavelets coefficients, which form a long and slow decaying tail. As expected, threshold–ISD did not return a reconstruction significantly better than either BP or IRL1 as shown in Figure 19. Even though thresholding in the wavelets domain is not effective



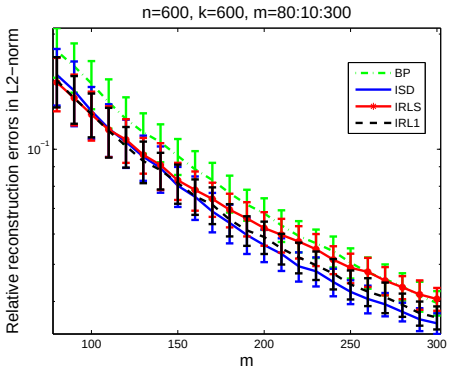
(a) Sparse signals

(b) Compressible signals



(c) λ=1/3: reconstruction errors in ℓ_2 norm

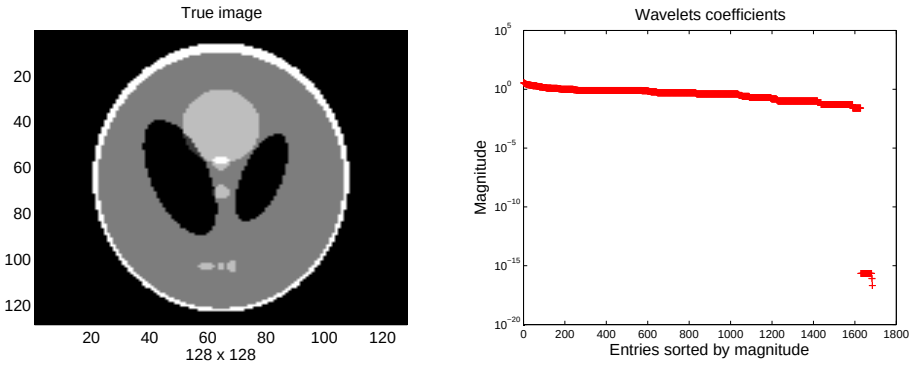
(d) λ=1/3: reconstruction errors in ℓ_2 norm



(e) λ=0.8: reconstruction errors in ℓ_2 norm

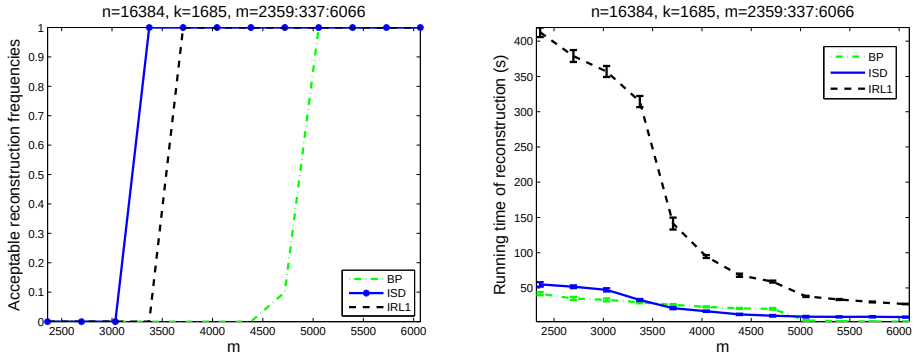
(f) λ=0.8: reconstruction errors in ℓ_2 norm

Figure 13: Test set 4 with sparse and compressible power–law decay signals. The sparse signals were obtained by removing the tails of the compressible signals. Comparisons in reconstruction errors. Fast decaying tails required for good performances of threshold–ISD, IRL1 and IRLS.



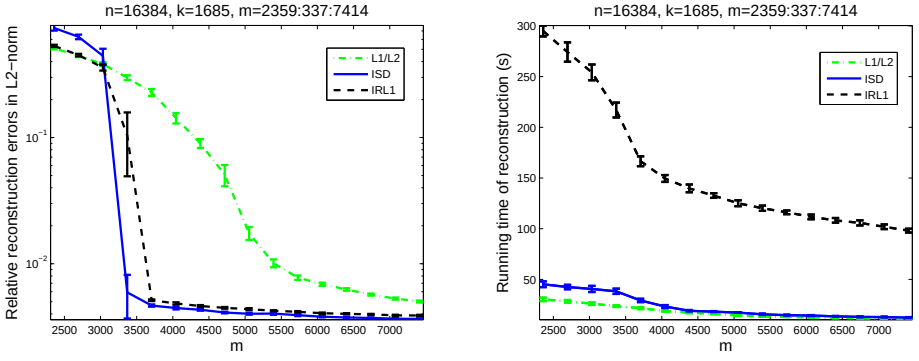
(a) (b)

Figure 14: Shepp–Logan phantom and its wavelets coefficients sorted by magnitude



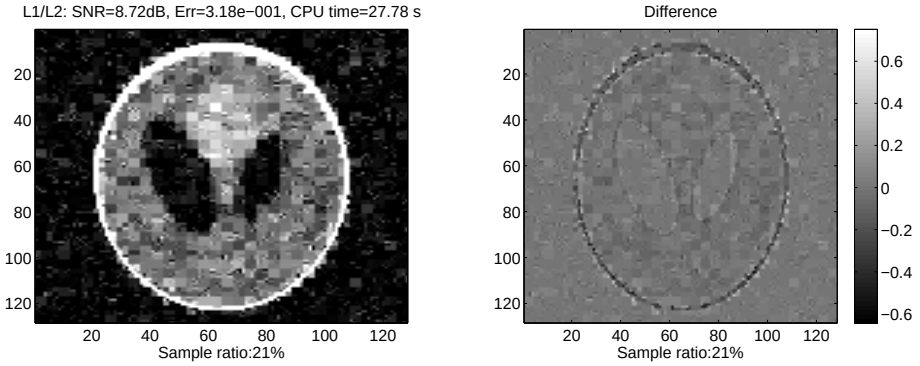
(a) (b)

Figure 15: Noiseless measurements: Acceptable reconstruction frequencies and running times



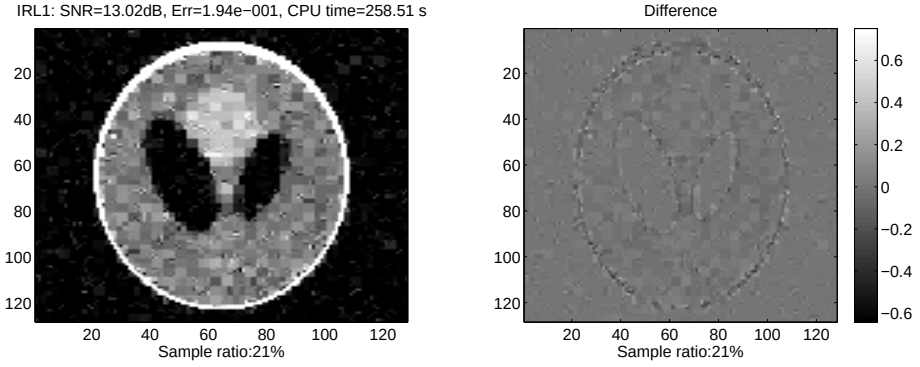
(a) (b)

Figure 16: Noisy measurements: Reconstruction errors and running times



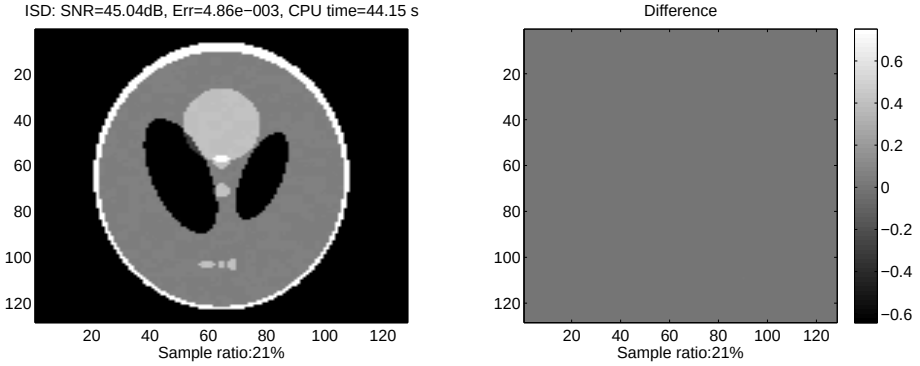
(a)

(b)



(c)

(d)



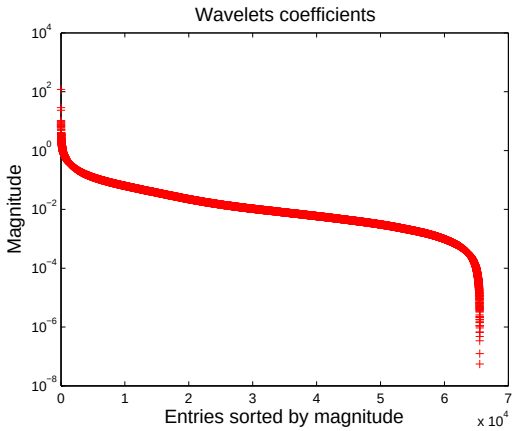
(e)

(f)

Figure 17: Noisy measurements: Reconstructed phantoms and highlighted errors corresponding to where $m = 3370$

for natural images, we note that the authors of [23], however, have obtained medical images with much better quality by combining ISD (applied to total variation) with edge detection techniques that replace thresholding. For this and other reasons, we believe that ISD with effective support detection is potentially very powerful.

(a)



(b)

Figure 18:

In summary, we compared threshold–ISD with IRLS, IRL1, and BP. Threshold–ISD can be solved as fast as BP yet achieves reconstruction quality as good as or better than the much slower IRLS. ISD relies on effective support detection. For threshold–ISD, its good performance requires fast–decaying signals.

6 Concluding Remarks

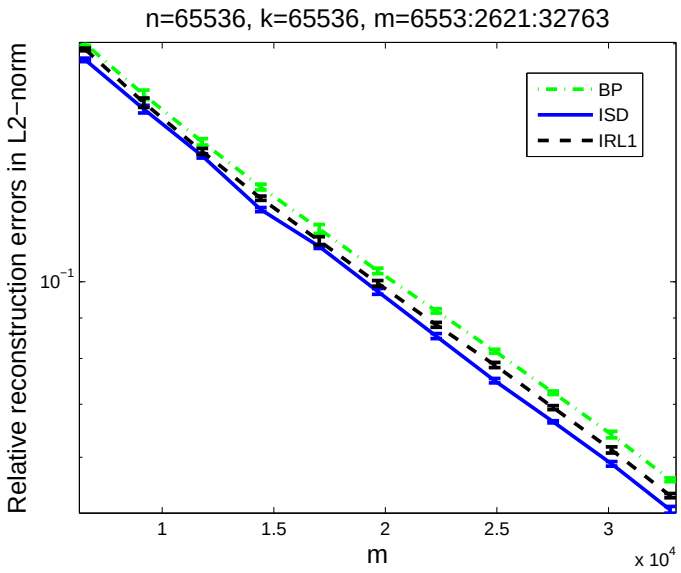
This paper introduces the iterative support detection method *ISD* for compressive sensing signal reconstruction. Both theoretical properties and practical performances are discussed. For signals with a fast decaying distribution of nonzeros, the implementation *threshold–ISD* equipped with the "first significant jump" thresholding rule is studied and demonstrated to be both fast and accurate for compressive sensing, compared to the classical approach BP and the state-of-the-art algorithms IRL1 and IRLS. Due to the limit of thresholding, threshold–ISD does not perform significantly better than its peers on other types of signals such as images.

However, support detection is not limited to thresholding. Effective support detection guarantees good performance of ISD according to our theoretical results. Therefore the future research includes studying specific signal classes and developing more effective support detection means, for example, by exploring signal structures (model–based CS[1]).

Since minimizing ℓ_1 is not the only approach for compressive sensing signal reconstruction, another line of future exposition is to apply iteration support detection to other reconstruction approaches such as the greedy algorithms, Bayesian algorithms, dictionary–based algorithms, and many others. We also feel that the usefulness of the above "first significant jump" rule is not limited to threshold–ISD.

Acknowledgement

The authors want to thank Professor Yin Zhang for making his YALL1 package available to us and Professors Weihong Guo and Yin Zhang for their valuable comments and suggestions that led to improvements of this report.



(a)

Figure 19: Reconstructing errors where the true signal is the wavelet coefficients of "Cameraman" and the sensing matrices are partial discrete cosine matrices

References

[1] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde. Model-based compressive sensing. *arXiv:0808.3572*, 2008.

[2] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde. Model-based comressive sensing. Preprint, 2008.

[3] J. Bioucas-Dias and M. Figueiredo. A new TwIST: two step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image Processing*, 16(12):2992–3004, 2007.

[4] E. Candès and J. Romberg. Sparsity and incoherence in compressive sampling. *Inverse Problems*, 23(3):969–985, 2007.

[5] E. Candès and T. Tao. Near optimal signal recovery from random projections: universal encoding strategies. *IEEE Transactions on Information Theory*, 52(1):5406–5425, 2006.

[6] E. J. Candes, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[7] E. J. Candes and T. Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, Dec. 2005.

[8] E. J. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on*, 52(12):5406–5425, 2006.

[9] E. J. Candes, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted l1 minimization. *The Journal of Fourier Analysis and Applications*, 2008.

[10] R. Chartrand and W. Yin. Iteratively reweighted algorithms for compressive sensing. In *ICASSP 2008*, pages 3869–3872, March 2008.

[11] A. Cohen, W. Dahmen, and R. Devore. Compressed sensing and best k-term approximation. *Journal of the American Mathematical Society*, 22(1):211–231, 2009.

[12] A. d'Aspremont and L. El Ghaoui. Testing the nullspace property using semidefinite programming. arXiv:0807.3520, February 2008.

[13] I. Daubechies, R. DeVore, M. Fornasier, and S. Gunturk. Iteratively re-weighted least squares minimization: Proof of faster than linear rate for sparse recovery. *Information Sciences and Systems, 2008. CISS 2008. 42nd Annual Conference on*, pages 26–29, March 2008.

[14] R. A. DeVore. Deterministic constructions of compressed sensing matrices. *Submitted*, 2007.

[15] D. Donoho and X. Huo. Uncertainty principles and ideal atomic decompositions. *IEEE Transactions on Information Theory*, 47:2845–2862, 2001.

[16] D. Donoho, Y. Tsaig, I. Drori, and J.-C. Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. *Submitted to IEEE Transactions on Information Theory*, 2006.

[17] D. L. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.

[18] D. L. Donoho and M. Elad. Optimally sparse representation in general (non-orthogonal) dictionaries via l1 minimization. *Proc. Natl. Acad. Sci.*, 100(5), 2003.

[19] D. L. Donoho and Y. Tsaig. Fast solution of ℓ_1 -norm minimization problems when the solution may be sparse. *Information Theory, IEEE Transactions on*, 54(11):4789–4812, Nov. 2008.

[20] M. Elad and A. Bruckstein. A generalized uncertainty principle and sparse representations in pairs of bases. *IEEE Transactions on Information Theory*, 48:2558–2567, 2002.

[21] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):586–597, 2007.

[22] J.J. Fuchs. On sparse representations in arbitrary redundant bases. *IEEE Transactions on Information Theory*, 50(1341–1344), 2004.

[23] W. Guo and W. Yin. Enhanced compressive MR imaging using mutual information and edge detectors. *In preparation*, 2009.

[24] E. T. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for ℓ_1-minimization. *Submitted to SIAM Journal on Optimization*, 2007.

[25] J. Jung and V. R. Durante. An iterative adaptive multiquadric radial basis function method for the detection of local jump discontinuities. *Appl. Numer. Math.*, 59(7):1449–1466, 2009.

[26] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. A method for large-scale l1-regularized least squares. *IEEE Journal on Selected Topics in Signal Processing*, 1(4):606–617, 2007.

[27] D. Needell and J. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Preprint*, 2008.

[28] M. Rudelson and R. Vershynin. Geometric approach to error correcting codes and reconstruction of signals. *International Mathematical Research Notices*, 64:4019–4041, 2005.

[29] J. Tropp and A. Gilbert. Signal recovery from partial information via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

[30] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang. A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization and continuation. Submitted, 2009.

[31] A. K. Yasakov. Method for detection of jump-like change points in optical data using approximations with distribution functions. In J. S. Jaffe, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 2258 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 605–612, oct 1994.

[32] W. Yin. Analysis and generalizations of the linearized bregman method. Technique report 09-02, Department of Computational and Applied Mathematics, Rice University, 2009.

[33] Y. Zhang. YALL1: Your algorithms for L1. http://www.caam.rice.edu/ optimization/L1/YALL1/.

[34] Y. Zhang. A simple proof for recoverability of ℓ_1-minimization: go over or under? *Rice University CAAM Technical Report TR05-09*, 2005.

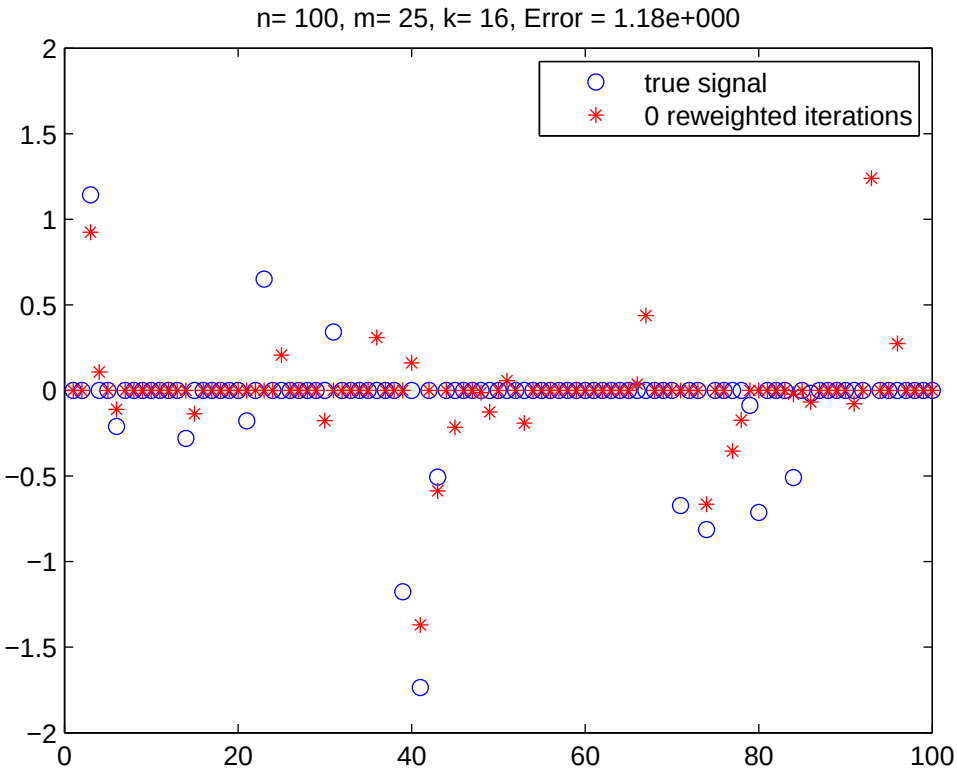
[35] Y. Zhang. A simple proof for recoverability of l1-minimization: Go over or under? Technical report 05-09, Department of Computational and Applied Mathmeatics, Rice University, 2005.

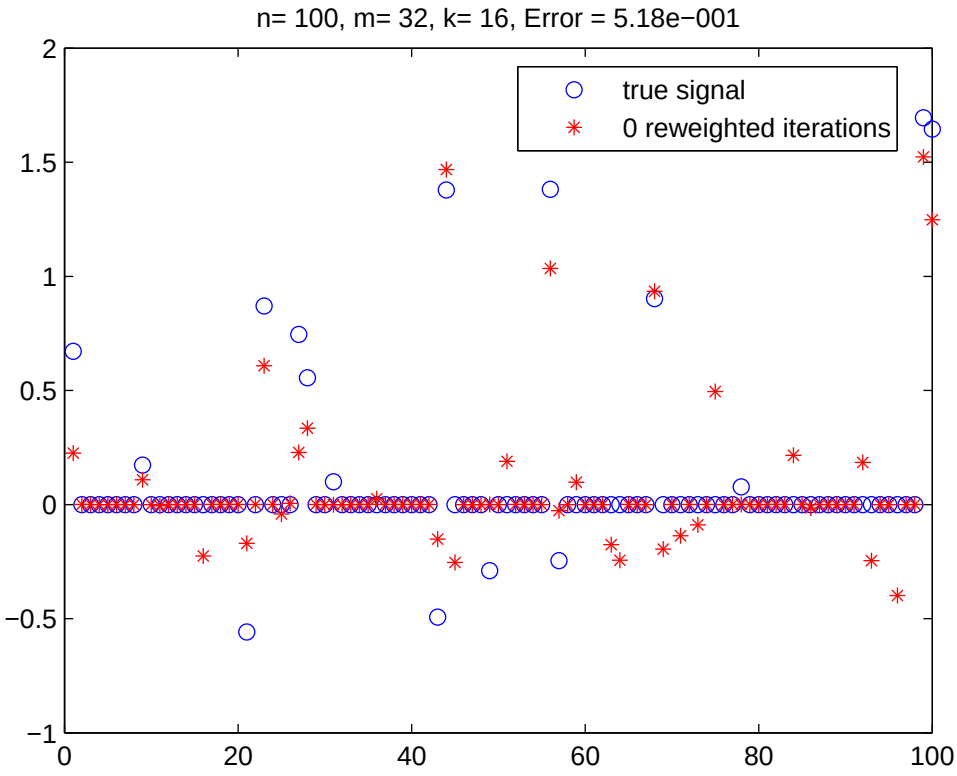
[36] Y. Zhang. A simple proof for recoverability of l1-minimization (ii): the nonnegativity case. Technical report 05-10, Department of Computational and Applied Mathmeatics, Rice University, 2005.

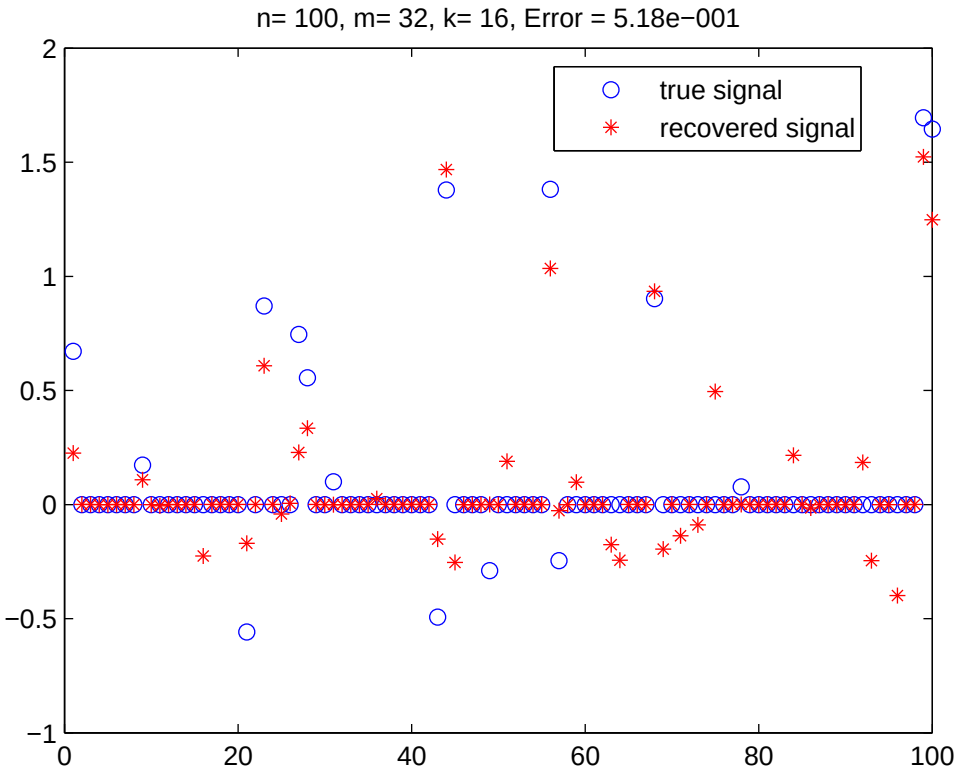
[37] Y. Zhang. When is missing data recoverable? *Rice University CAAM Technical Report TR06-15*, 2006.

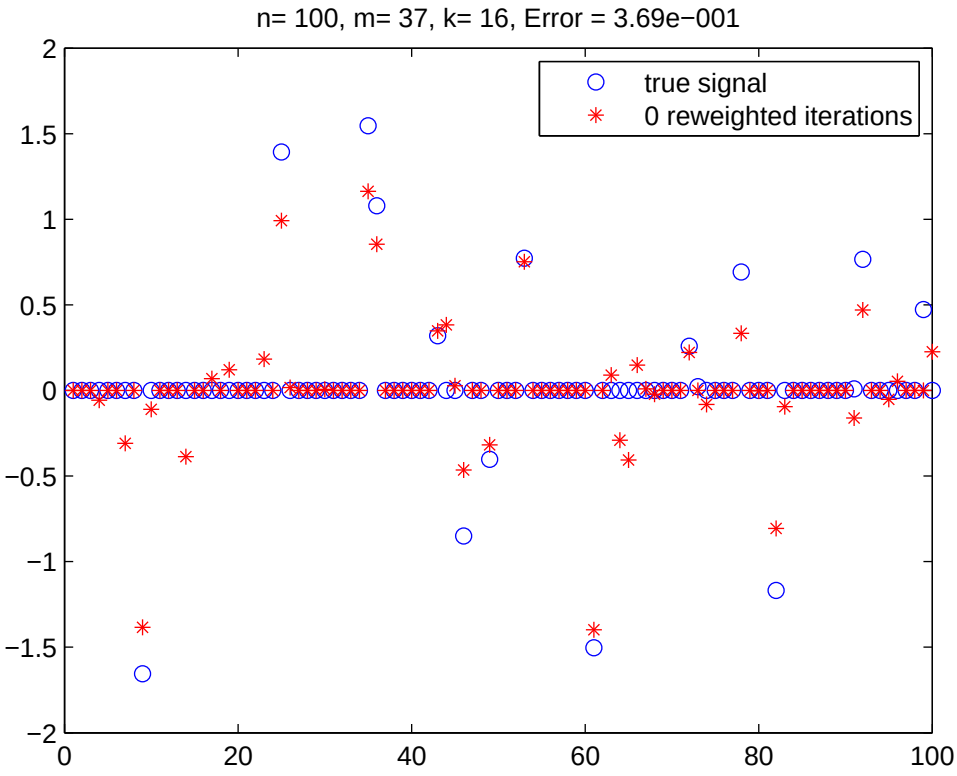
[38] Y. Zhang. On theory of compressive sensing via ℓ_1-minimization: Simple derivations and extensions. Technical report 08-11, Department of Computational and Applied Mathmeatics, Rice University, 2008.

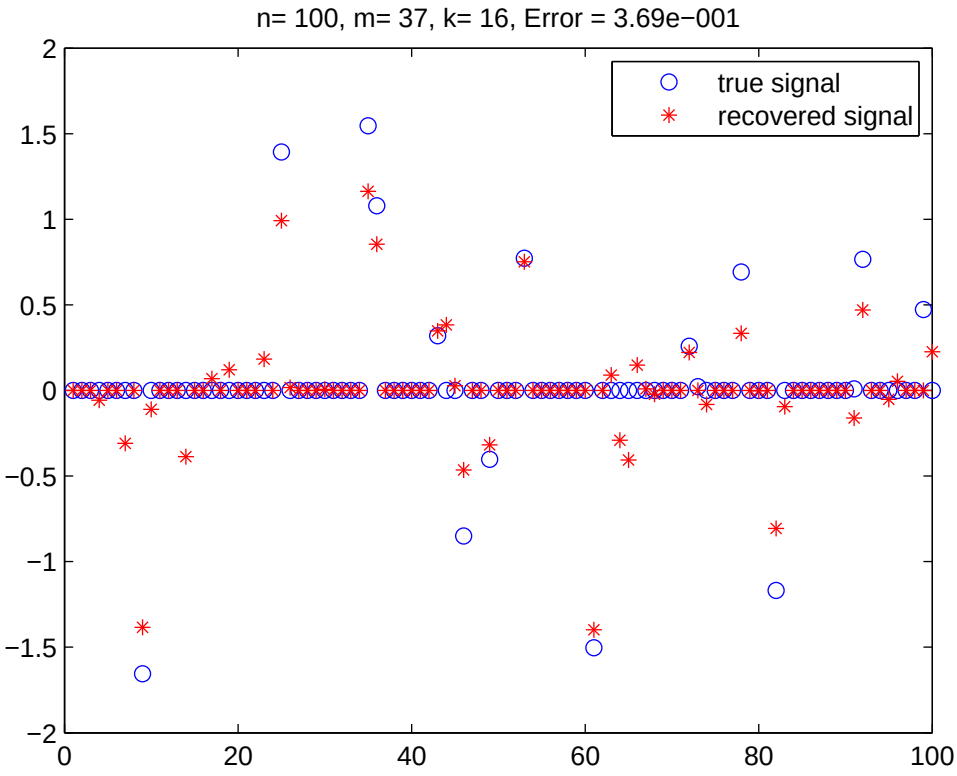
[39] Y. Zhang. User's guide for YALL1: Your algorithms for L1 optimization. Technique report 09-17, Department of Computational and Applied Mathematics, Rice University, 2009.

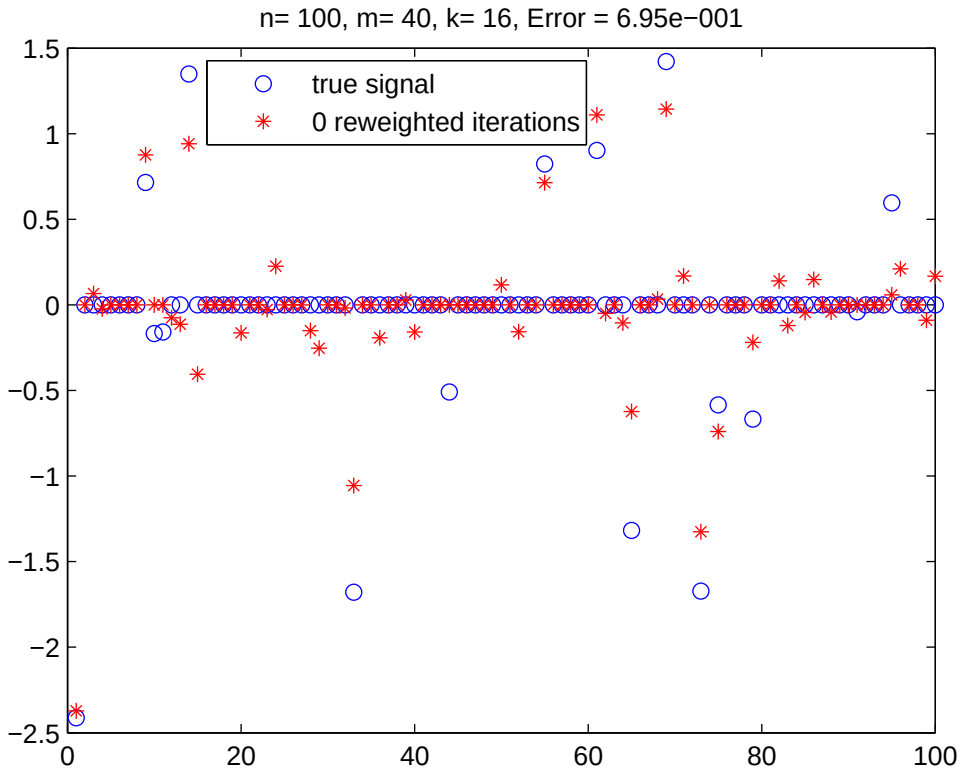

n= 100, m= 25, k= 16, Error = 1.18e+000
true signal
0 reweighted iterations
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
20
40
60
80
100

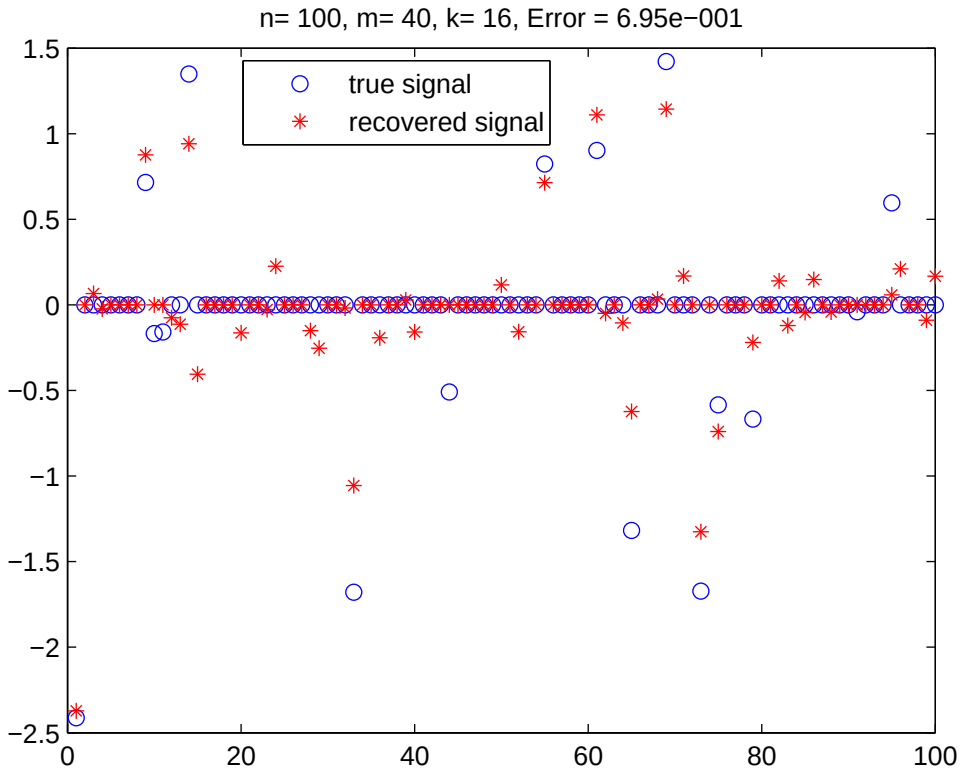

n= 100, m= 32, k= 16, Error = 5.18e−001
true signal
0 reweighted iterations
2
1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

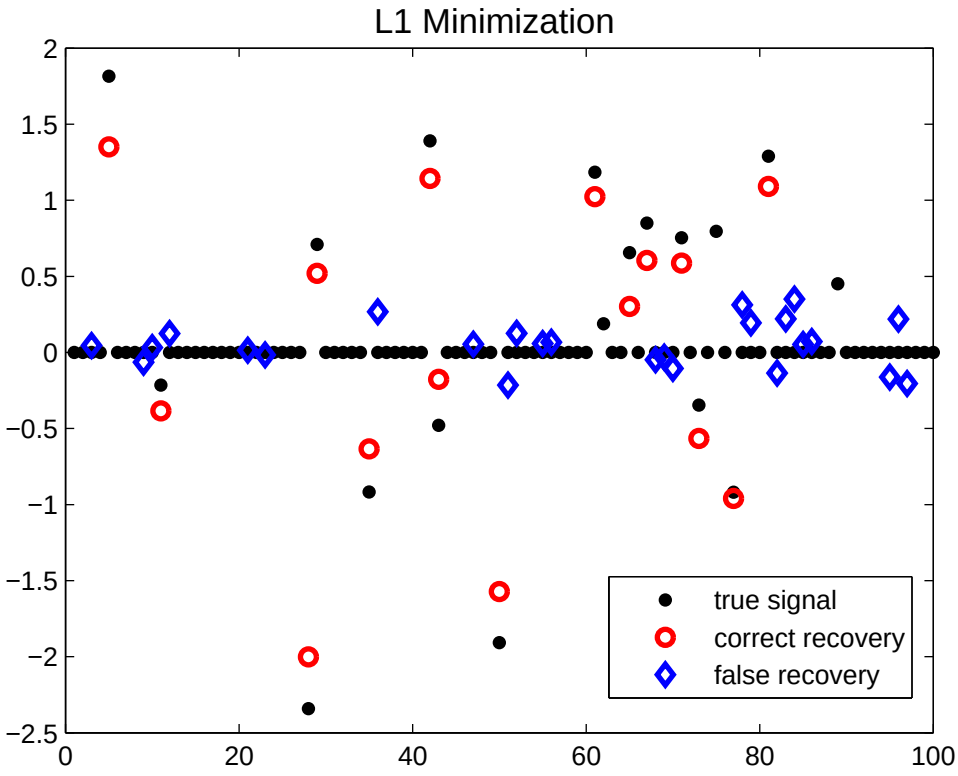

n= 100, m= 32, k= 16, Error = 5.18e−001
true signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

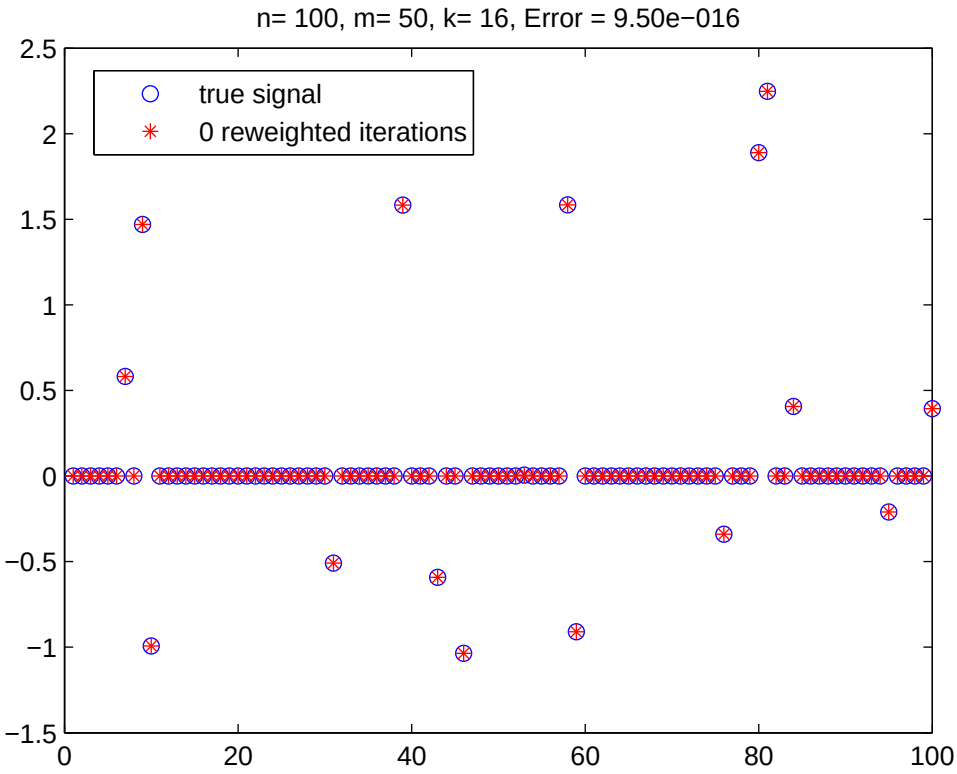

n= 100, m= 37, k= 16, Error = 3.69e−001
true signal
0 reweighted iterations
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
20
40
60
80
100

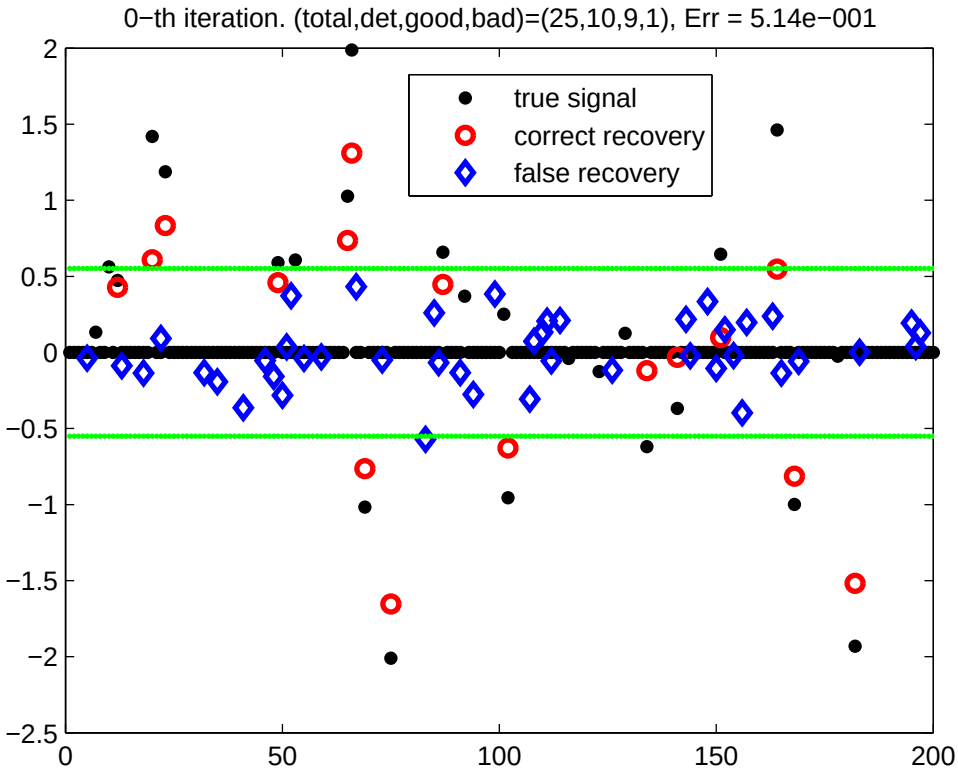

n= 100, m= 37, k= 16, Error = 3.69e−001
true signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
20
40
60
80
100

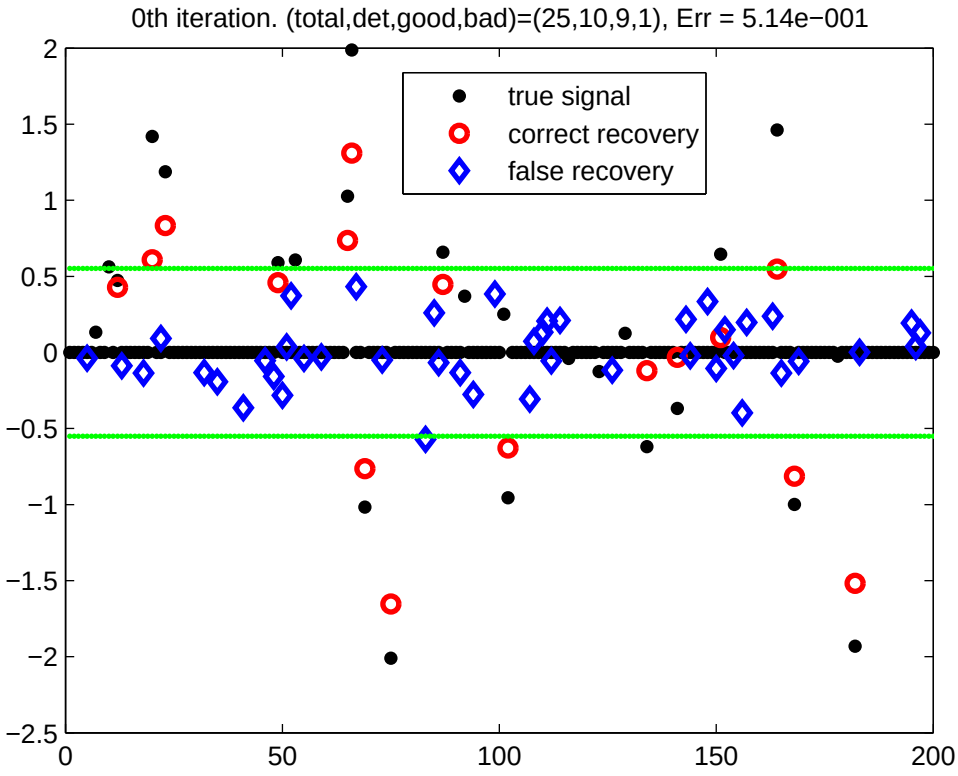

n= 100, m= 40, k= 16, Error = 6.95e−001
true signal
0 reweighted iterations
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
20
40
60
80
100

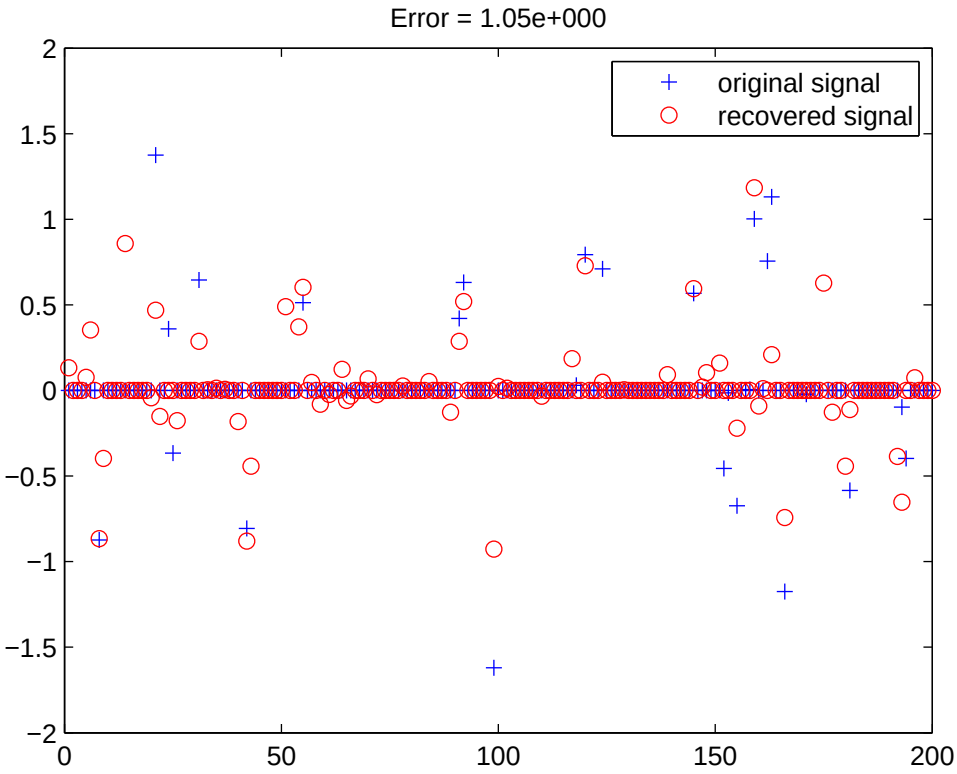

n= 100, m= 40, k= 16, Error = 6.95e−001
true signal
recovered signal
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
20
40
60
80
100

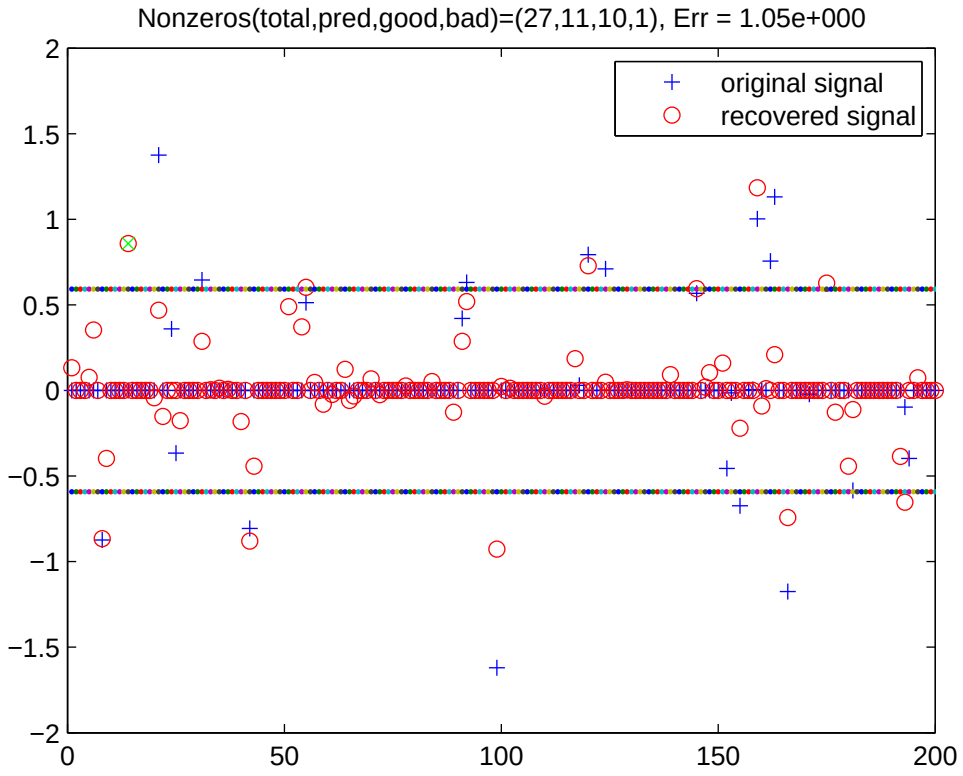

L1 Minimization
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
20
40
60
80
100
true signal
correct recovery
false recovery

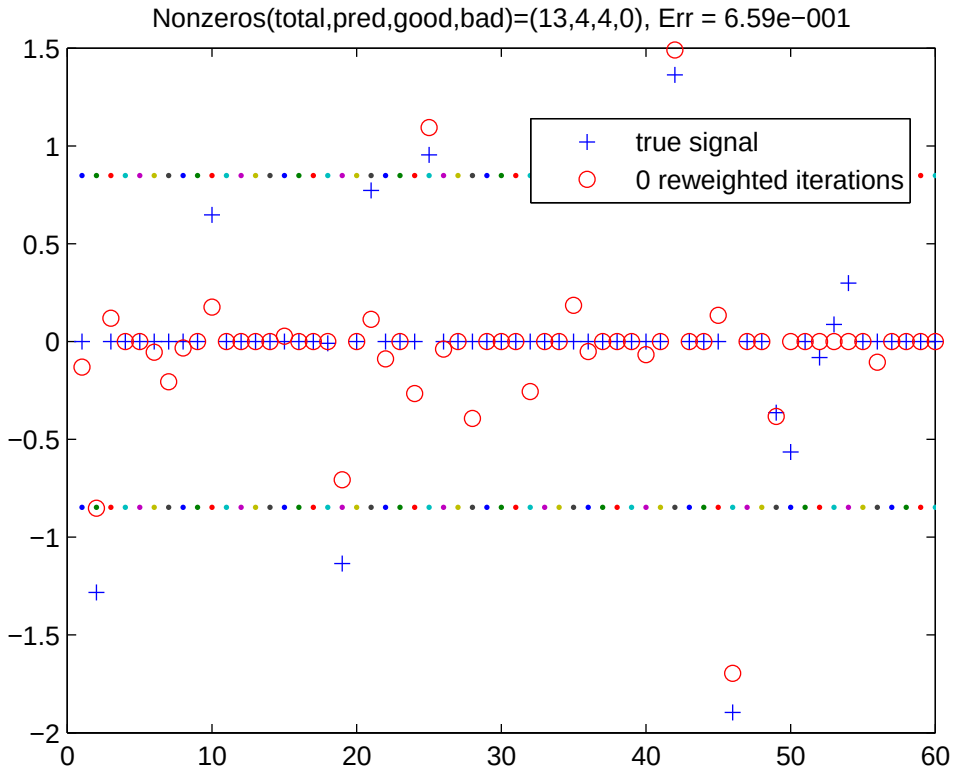

n= 100, m= 50, k= 16, Error = 9.50e−016
true signal
0 reweighted iterations
2.5
2
1.5
1
0.5
0
−0.5
−1
−1.5
0
20
40
60
80
100

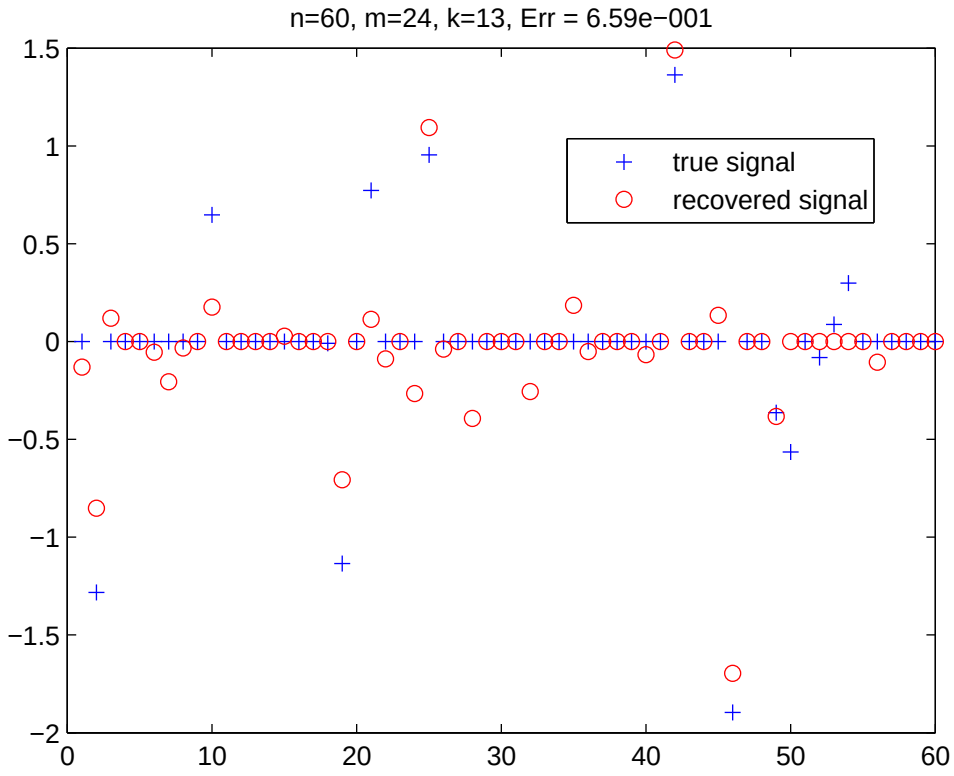

0−th iteration. (total,det,good,bad)=(25,10,9,1), Err = 5.14e−001
true signal
correct recovery
false recovery
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
50
100
150
200

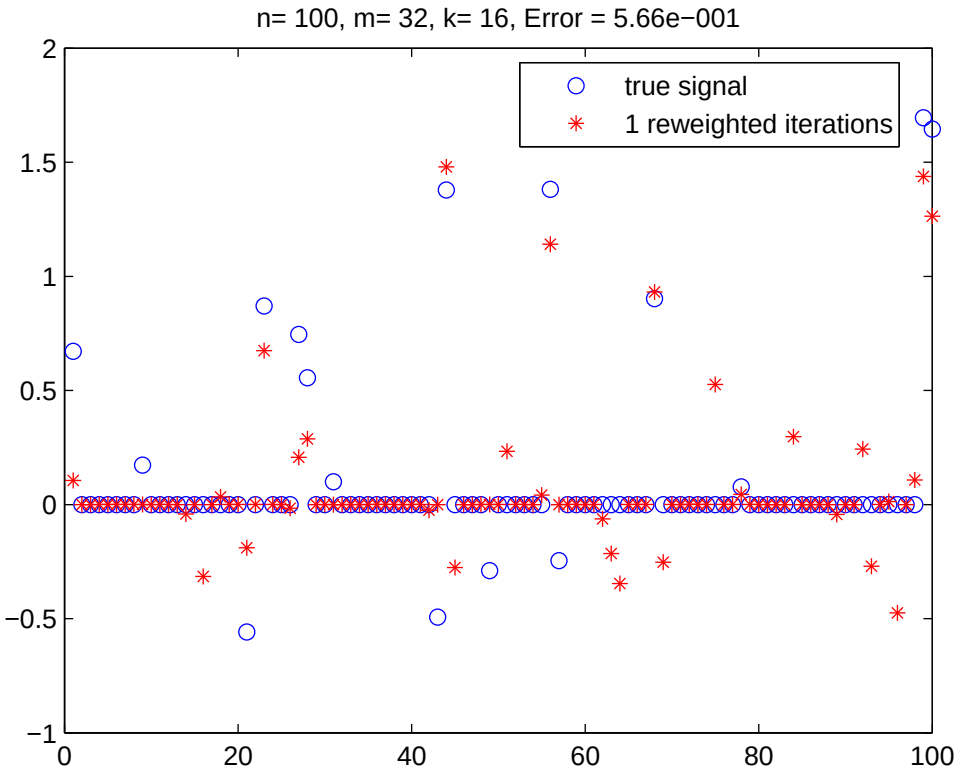

0th iteration. (total,det,good,bad)=(25,10,9,1), Err = 5.14e−001
true signal
correct recovery
false recovery
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
50
100
150
200

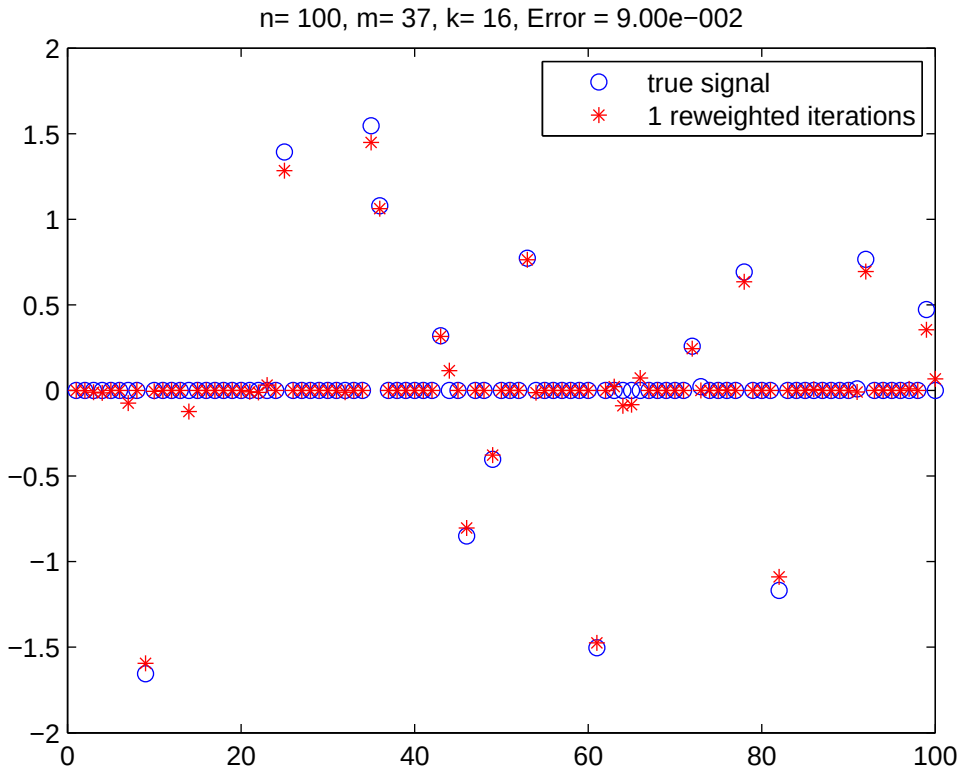

Error = 1.05e+000
original signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
50
100
150
200

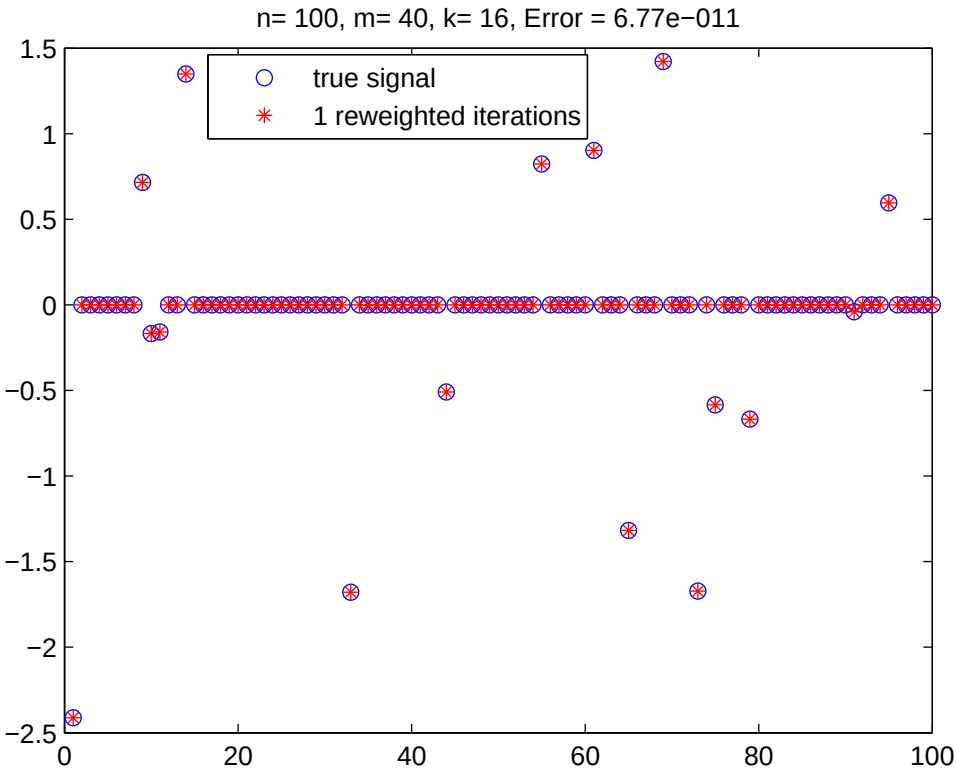

Nonzeros(total,pred,good,bad)=(27,11,10,1), Err = 1.05e+000
original signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
50
100
150
200

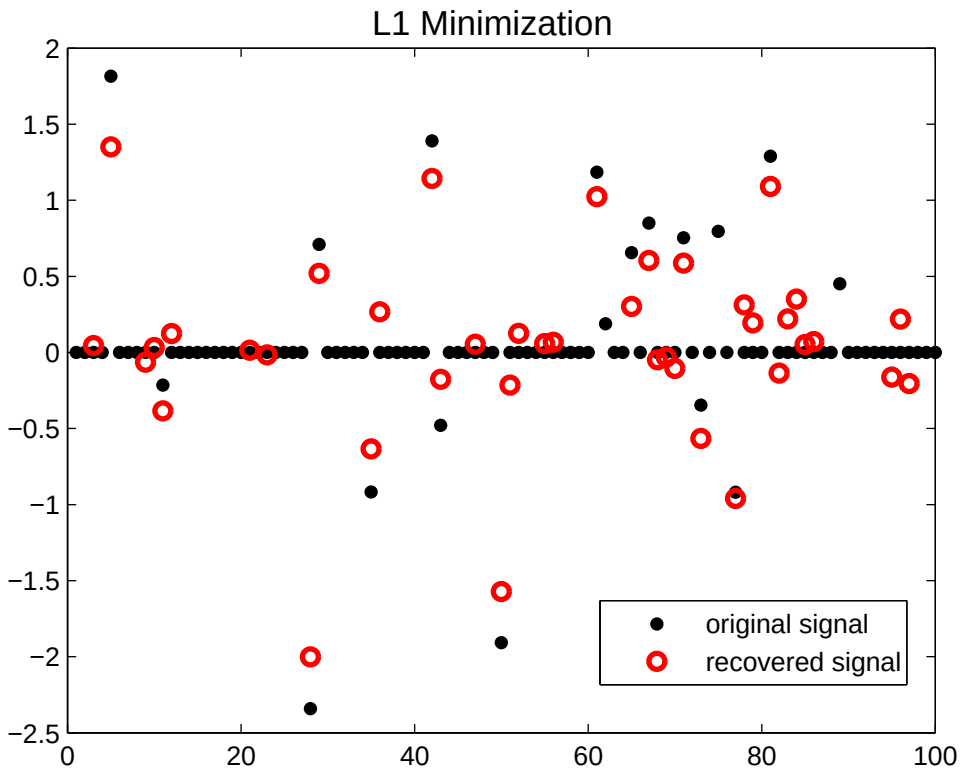

Nonzeros(total,pred,good,bad)=(13,4,4,0), Err = 6.59e−001
true signal
0 reweighted iterations
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
10
20
30
40
50
60

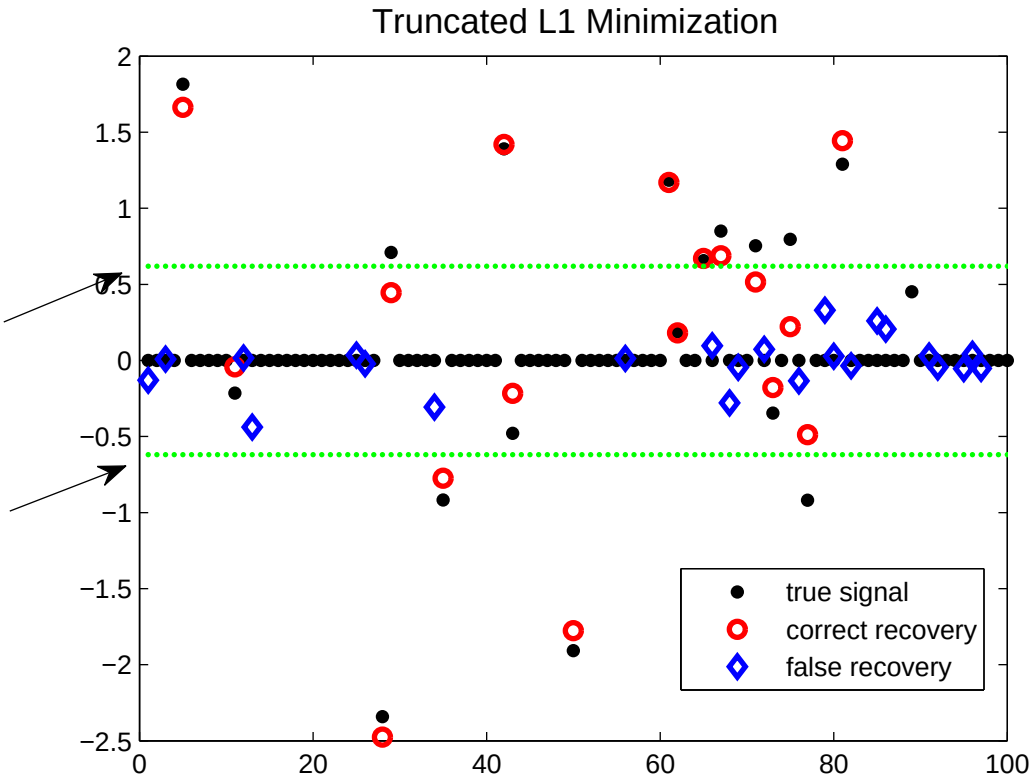

n=60, m=24, k=13, Err = 6.59e−001
true signal
recovered signal
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
10
20
30
40
50
60

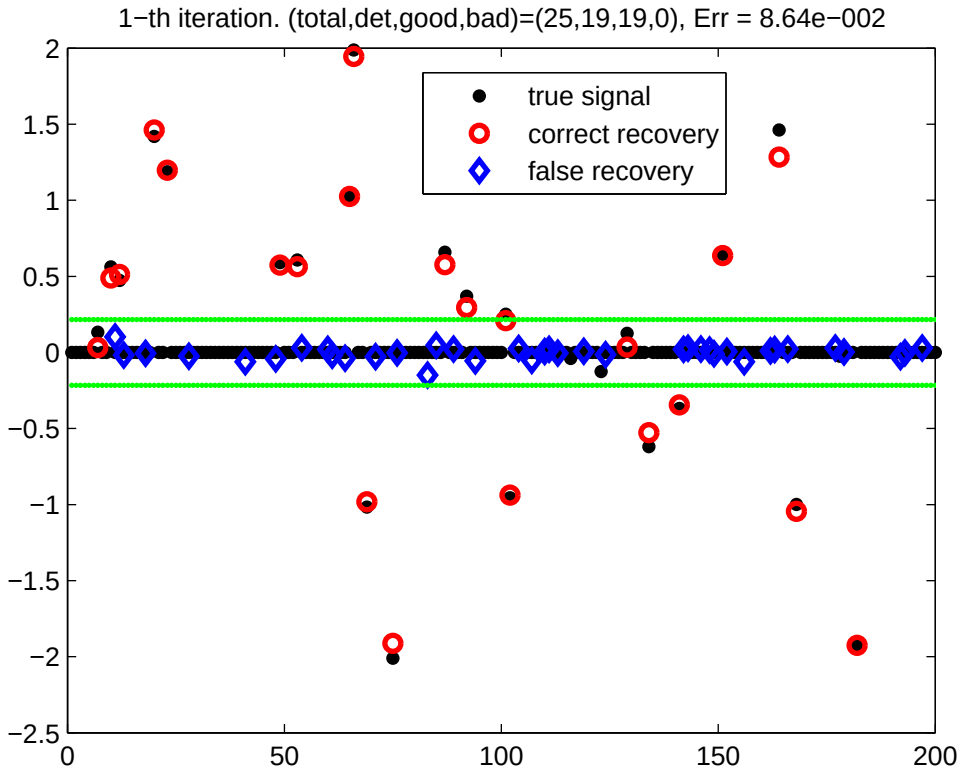

n= 100, m= 32, k= 16, Error = 5.66e−001
true signal
1 reweighted iterations
2
1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

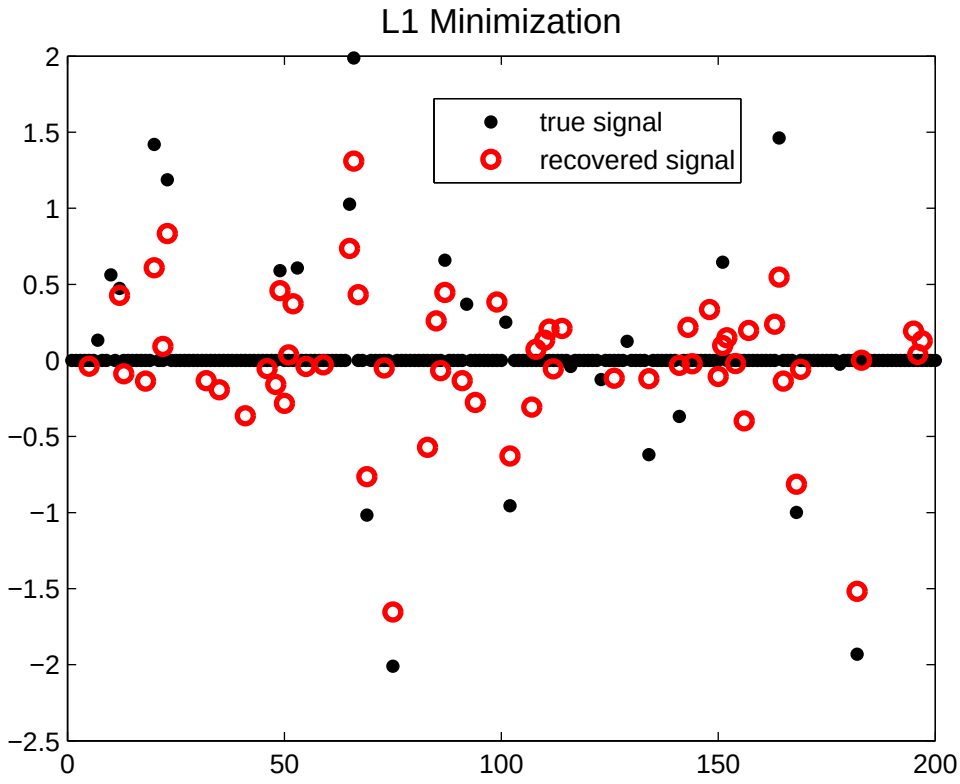

n= 100, m= 37, k= 16, Error = 9.00e−002
true signal
1 reweighted iterations
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
20
40
60
80
100

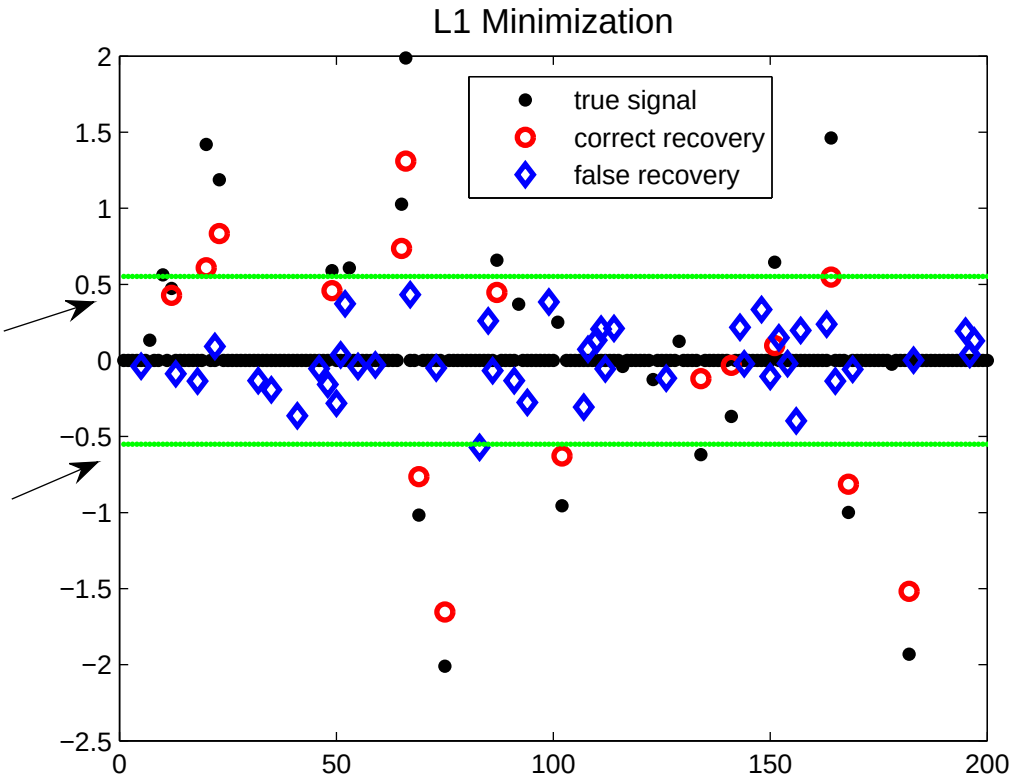

n= 100, m= 40, k= 16, Error = 6.77e−011
true signal
1 reweighted iterations
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
20
40
60
80
100

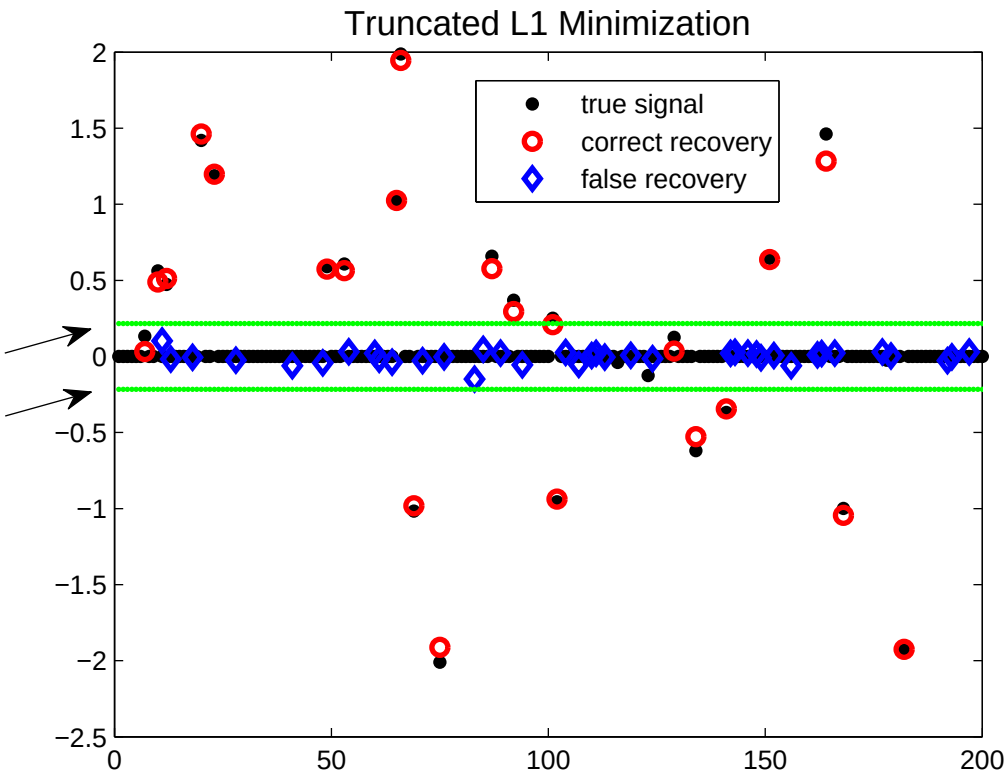

L1 Minimization
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
20
40
60
80
100
original signal
recovered signal

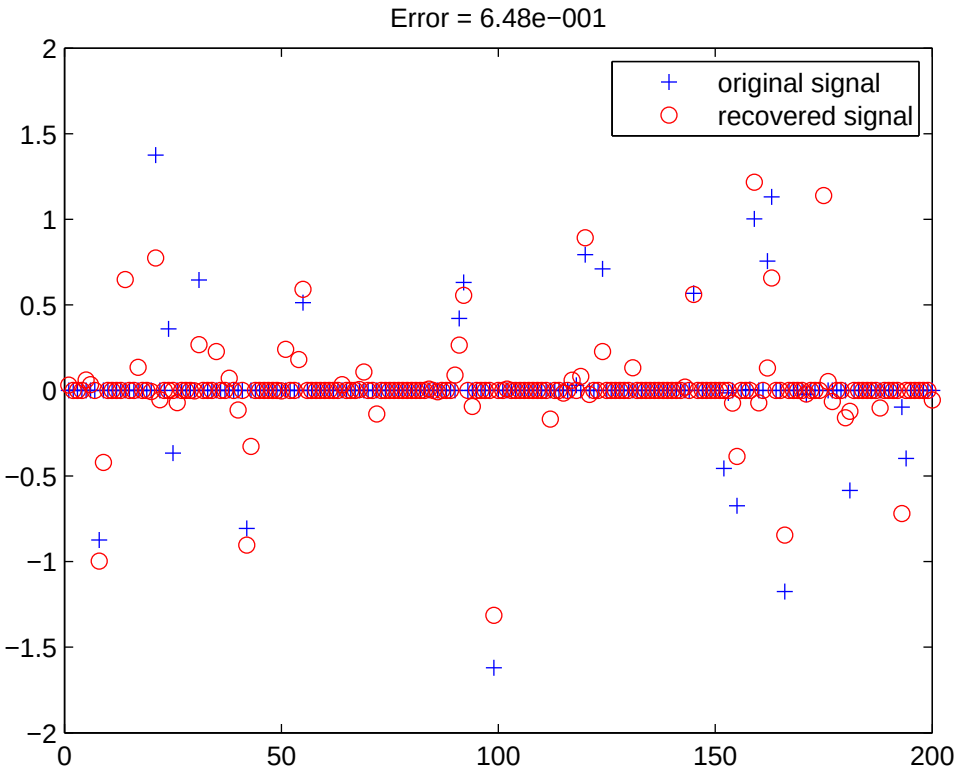

Truncated L1 Minimization
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
20
40
60
80
100
true signal
correct recovery
false recovery

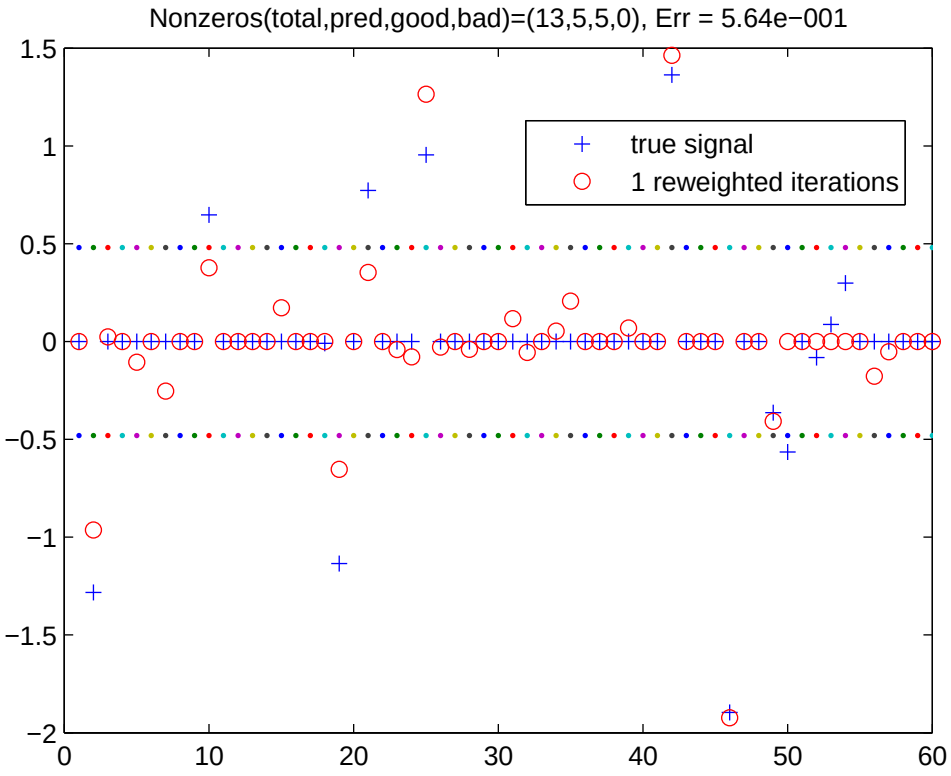

1−th iteration. (total,det,good,bad)=(25,19,19,0), Err = 8.64e−002
true signal
correct recovery
false recovery
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
50
100
150
200

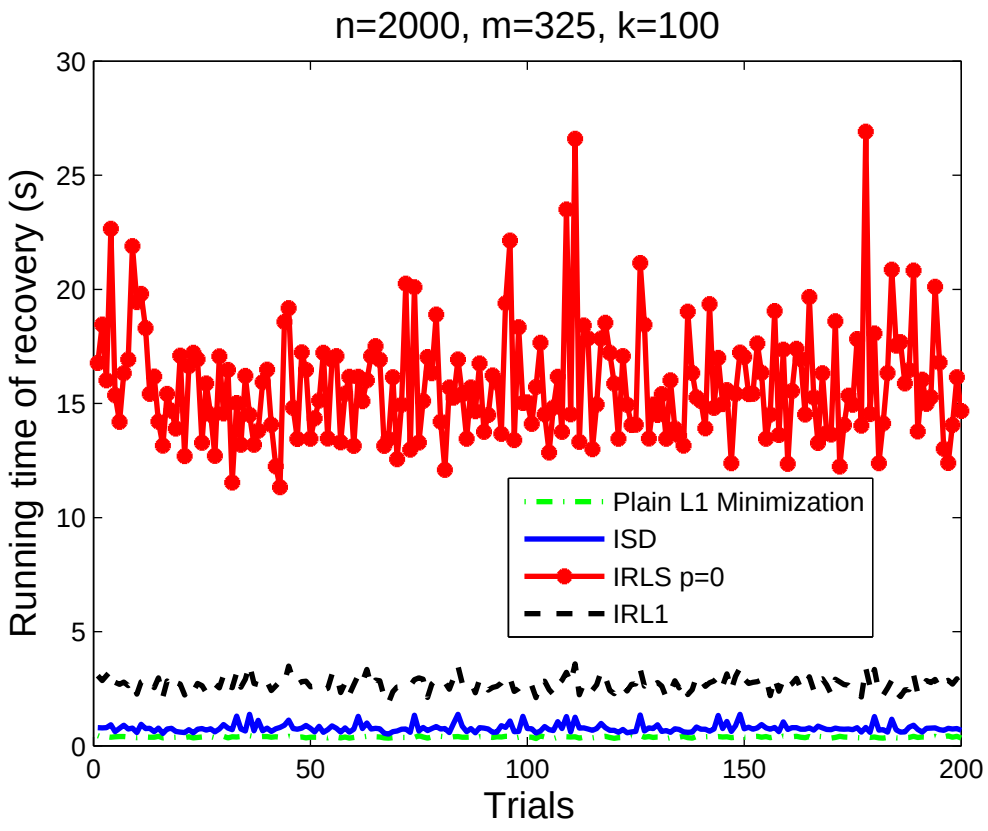

L1 Minimization
true signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
50
100
150
200

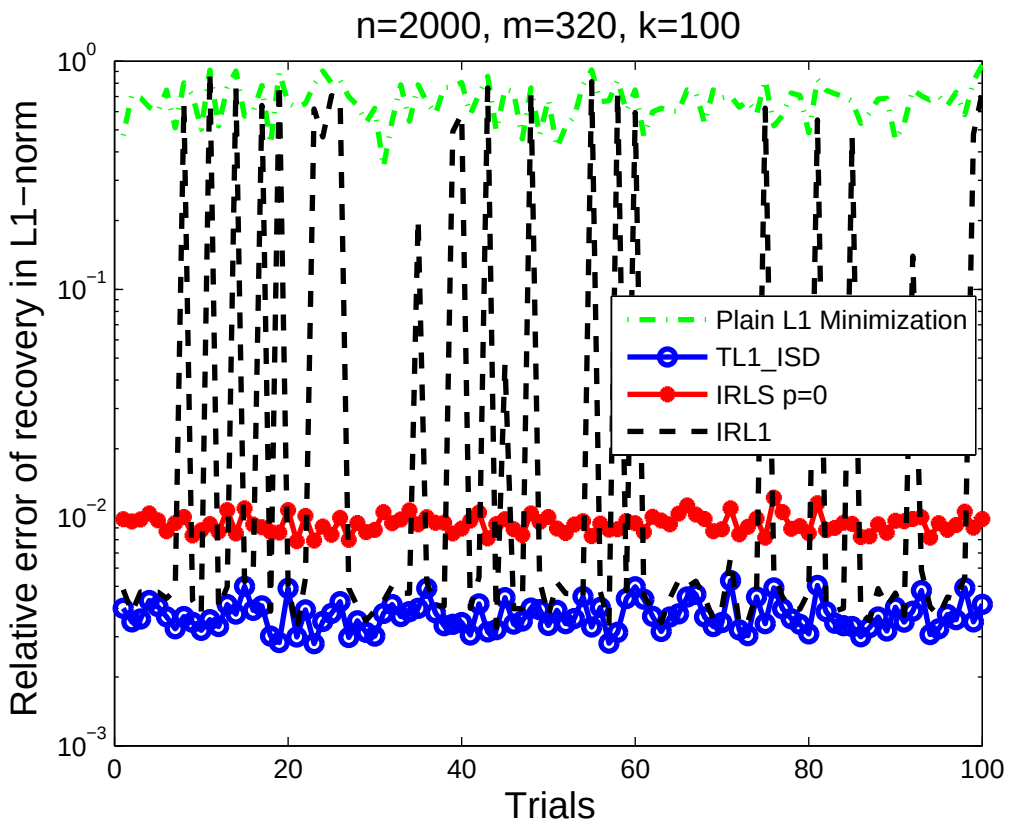

L1 Minimization
true signal
correct recovery
false recovery
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
50
100
150
200

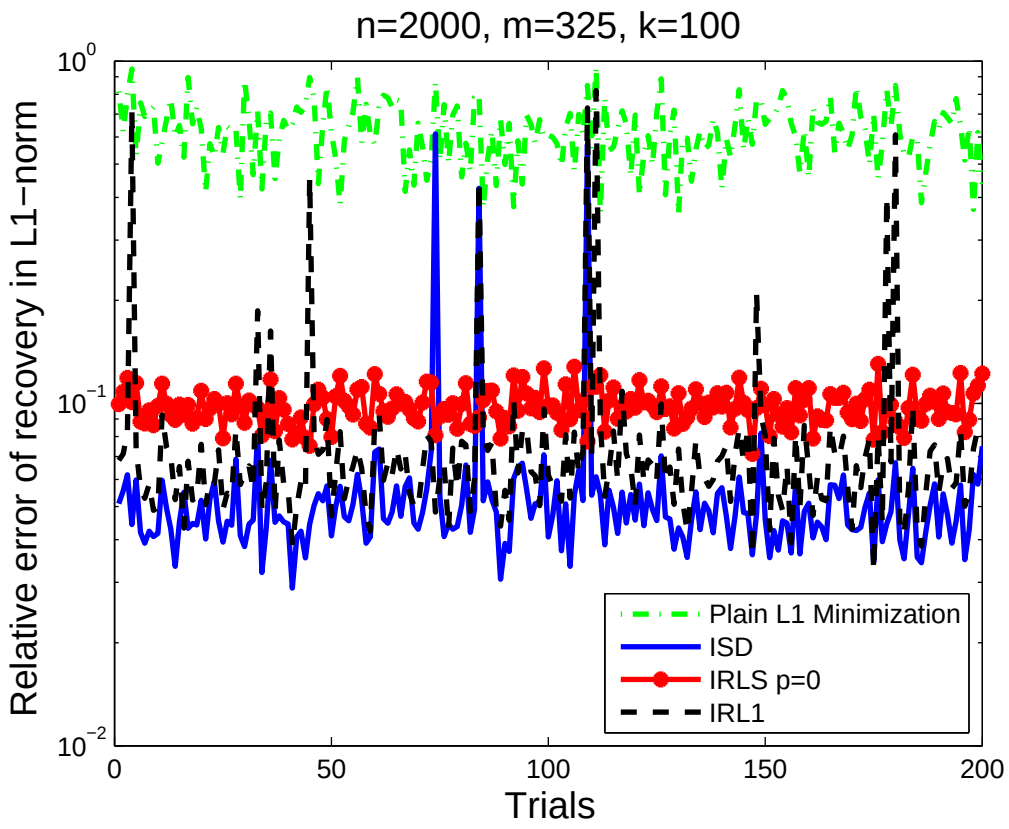

Truncated L1 Minimization
true signal
correct recovery
false recovery
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
50
100
150
200

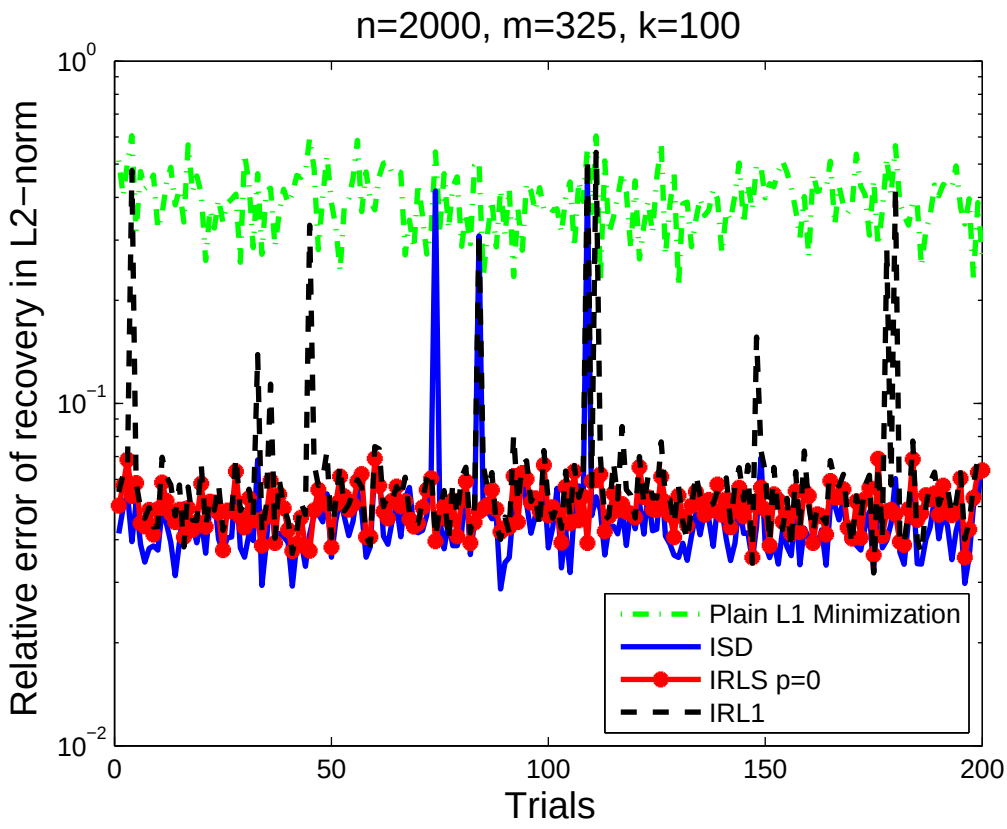

Error = 6.48e−001
original signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
50
100
150
200

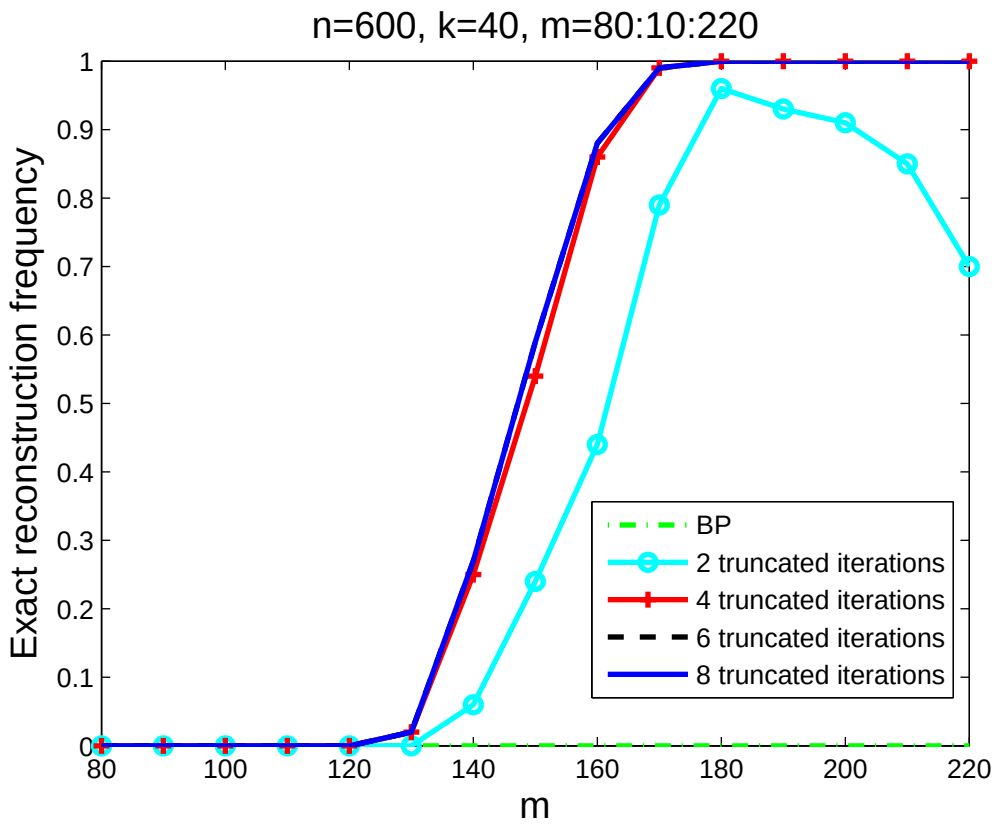

Nonzeros(total,pred,good,bad)=(13,5,5,0), Err = 5.64e−001
true signal
1 reweighted iterations
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
10
20
30
40
50
60

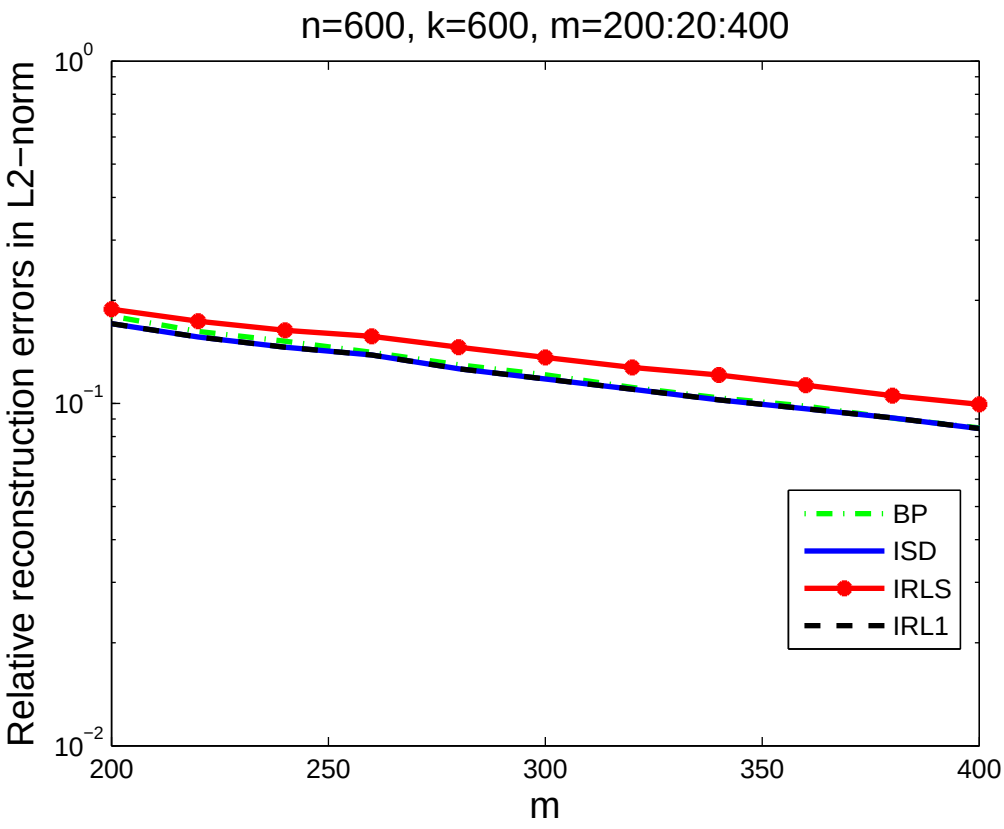

n=2000, m=325, k=100
Running time of recovery (s)
Trials
0
5
10
15
20
25
30
0
50
100
150
200
Plain L1 Minimization
ISD
IRLS p=0
IRL1


n=2000, m=320, k=100
Relative error of recovery in L1−norm
Trials
Plain L1 Minimization
TL1_ISD
IRLS p=0
IRL1
10^{0}
10^{-1}
10^{-2}
10^{-3}
0
20
40
60
80
100


n=2000, m=325, k=100
Relative error of recovery in L1−norm
Trials
10^0
10^{-1}
10^{-2}
0
50
100
150
200
Plain L1 Minimization
ISD
IRLS p=0
IRL1


n=2000, m=325, k=100
Relative error of recovery in L2−norm
Trials
10^0
10^{-1}
10^{-2}
0
50
100
150
200
Plain L1 Minimization
ISD
IRLS p=0
IRL1


n=600, k=40, m=80:10:220
Exact reconstruction frequency
m
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
80
100
120
140
160
180
200
220
BP
2 truncated iterations
4 truncated iterations
6 truncated iterations
8 truncated iterations


n=600, k=600, m=200:20:400
Relative reconstruction errors in L2−norm
10^{0}
10^{-1}
10^{-2}
200
250
300
350
400
m
BP
ISD
IRLS
IRL1

BP: SNR=17.03dB, Err=1.22e−01, CPU time=41.83 s

18% data

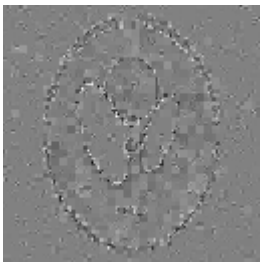
Original

256 x 256

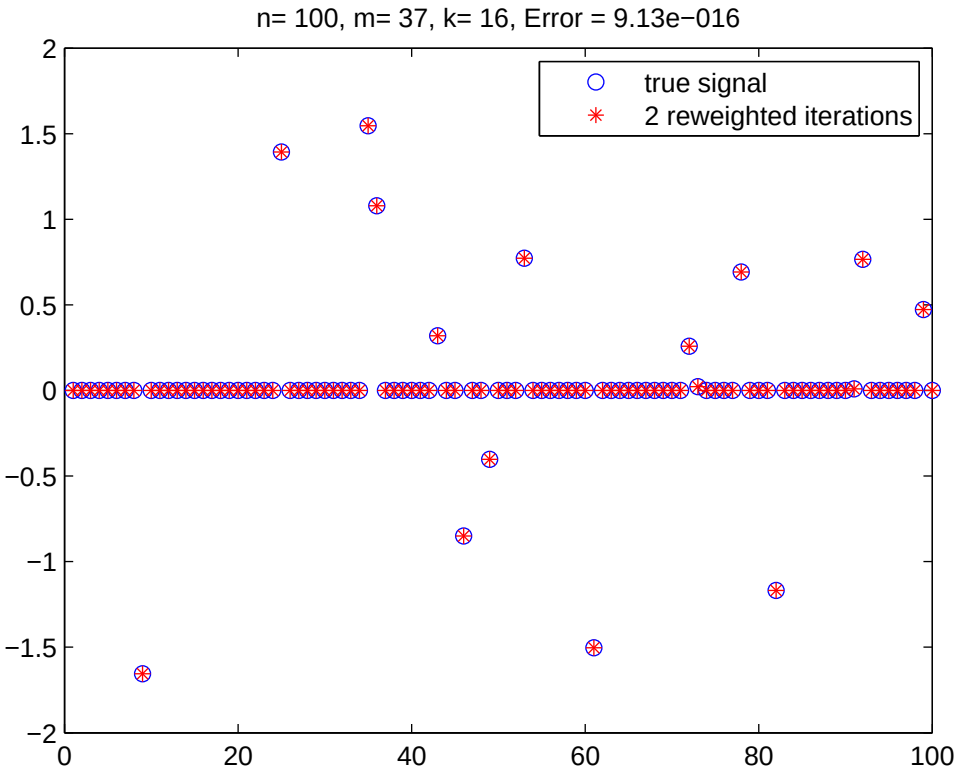
IRL1: SNR=21.65dB, Err=7.18e−002, CPU time=313.67 s

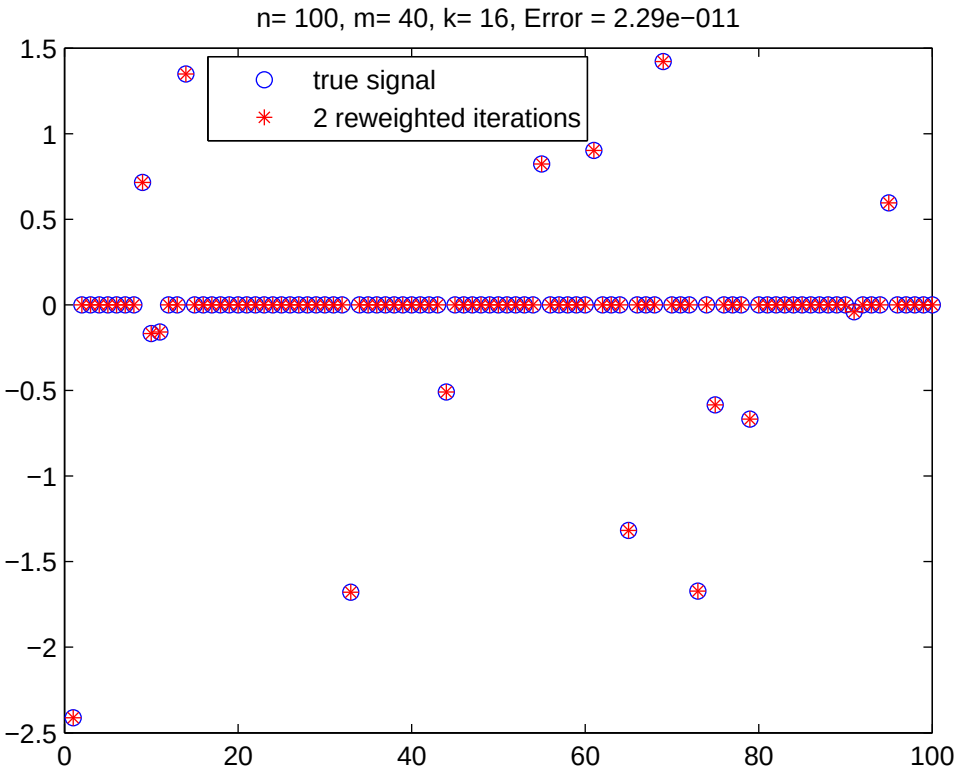
Sample ratio:21%

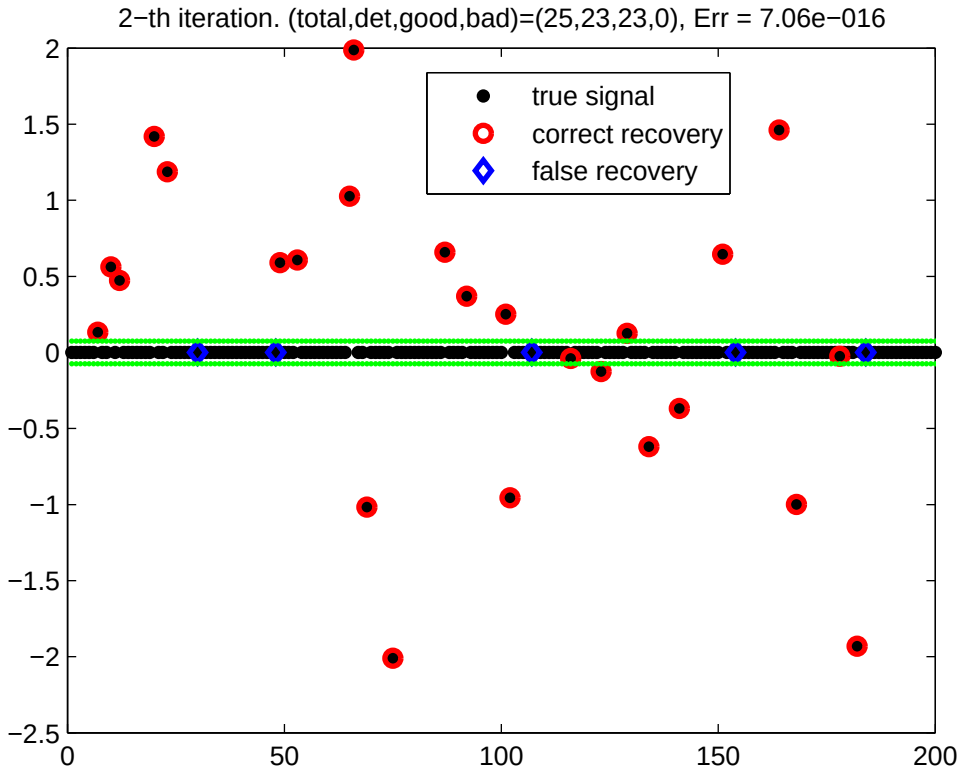
Subtractoin

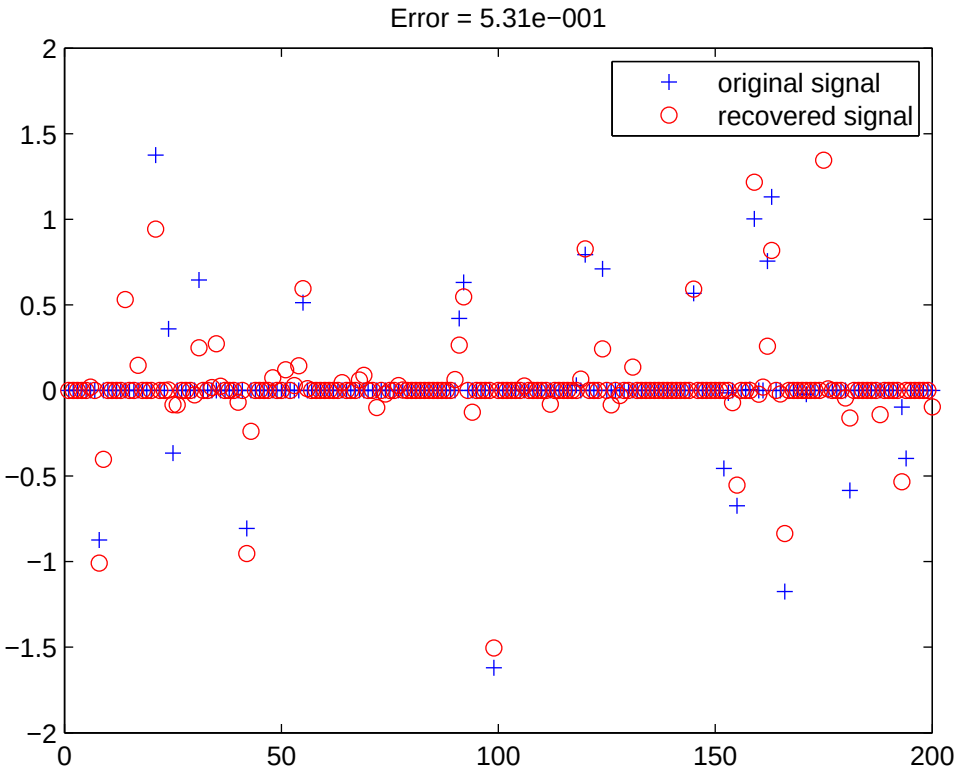


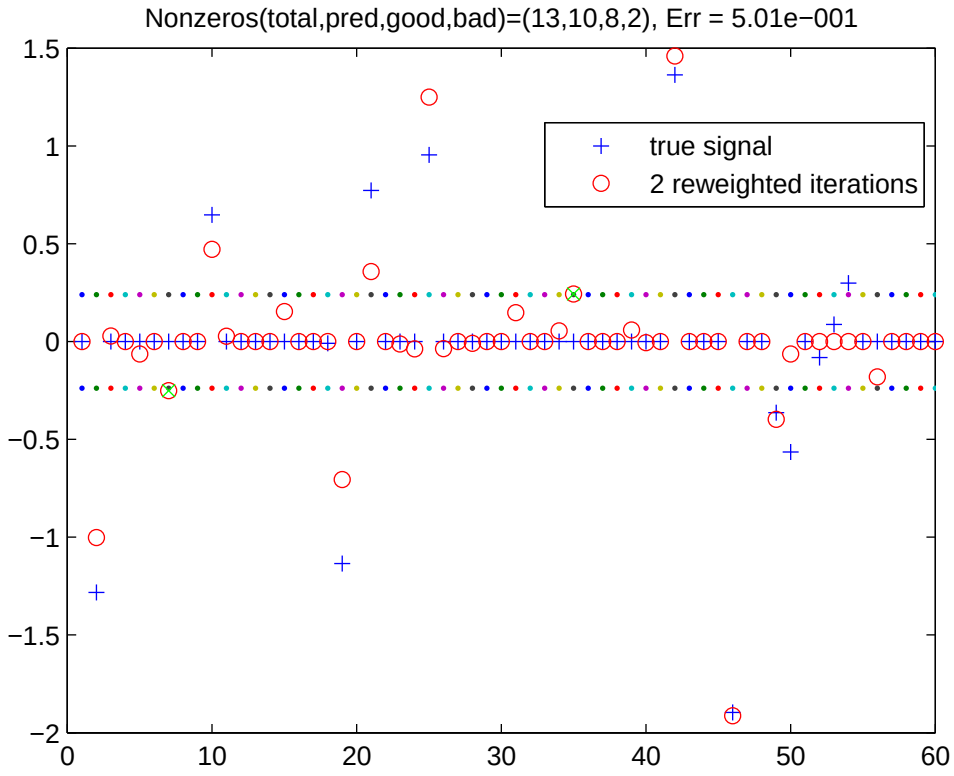
Sample ratio:21%

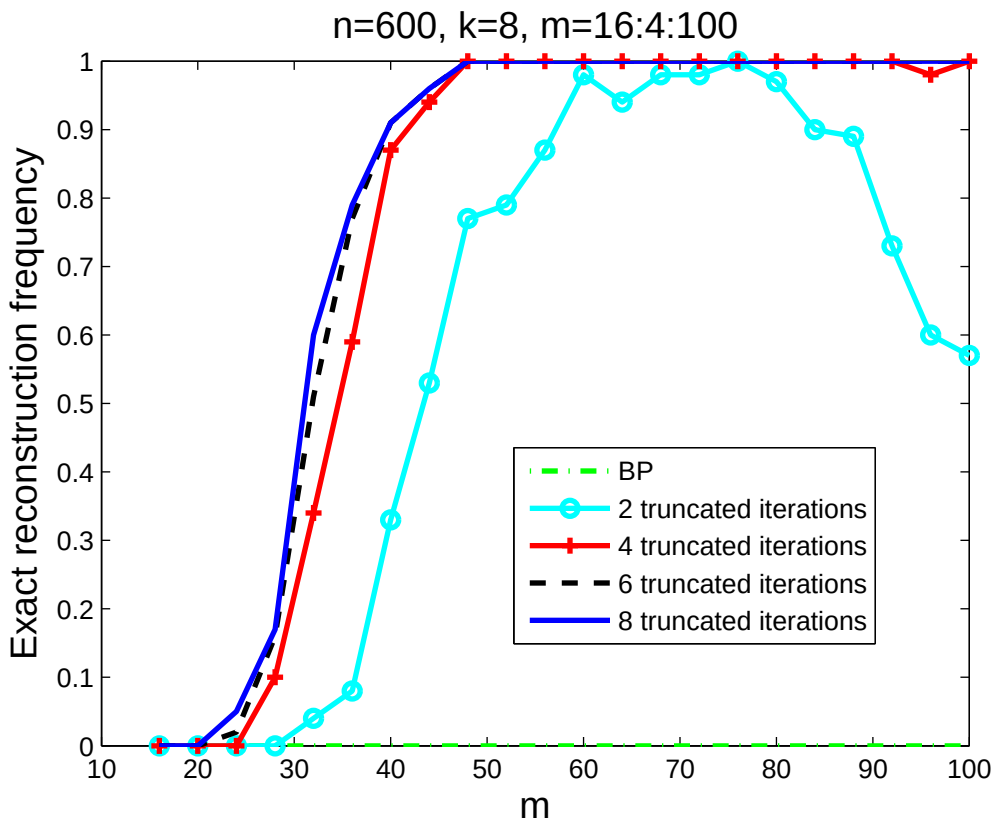

n= 100, m= 37, k= 16, Error = 9.13e−016
true signal
2 reweighted iterations
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
20
40
60
80
100

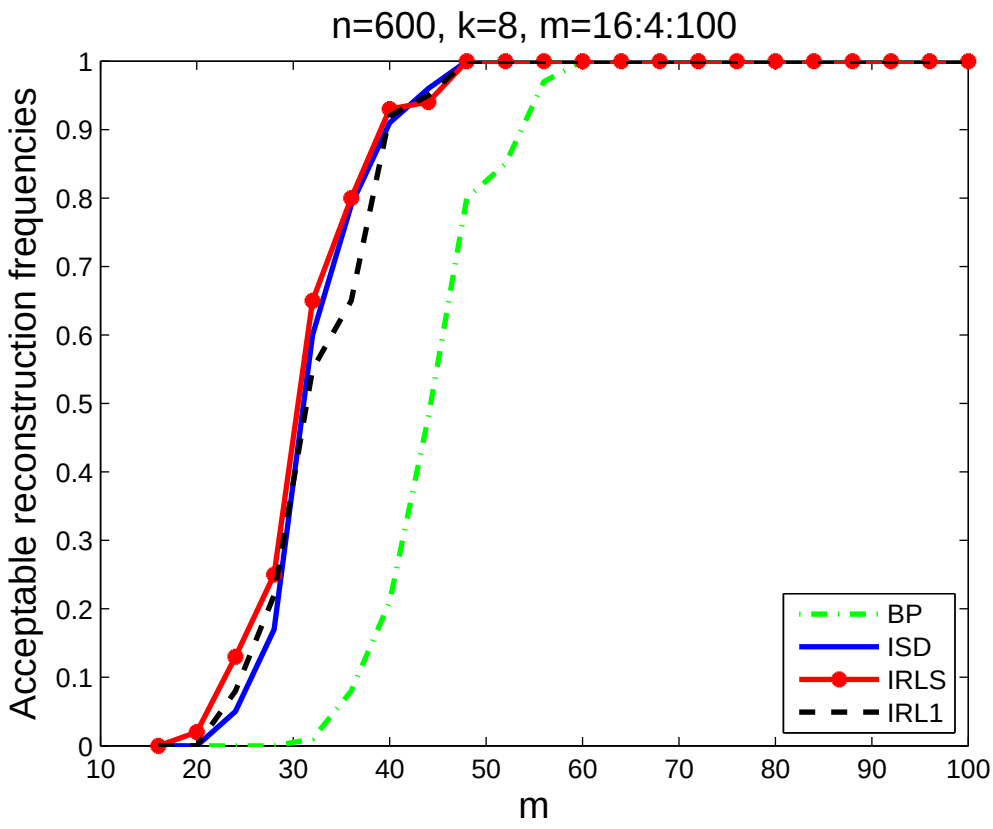

n= 100, m= 40, k= 16, Error = 2.29e−011
true signal
2 reweighted iterations
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
20
40
60
80
100

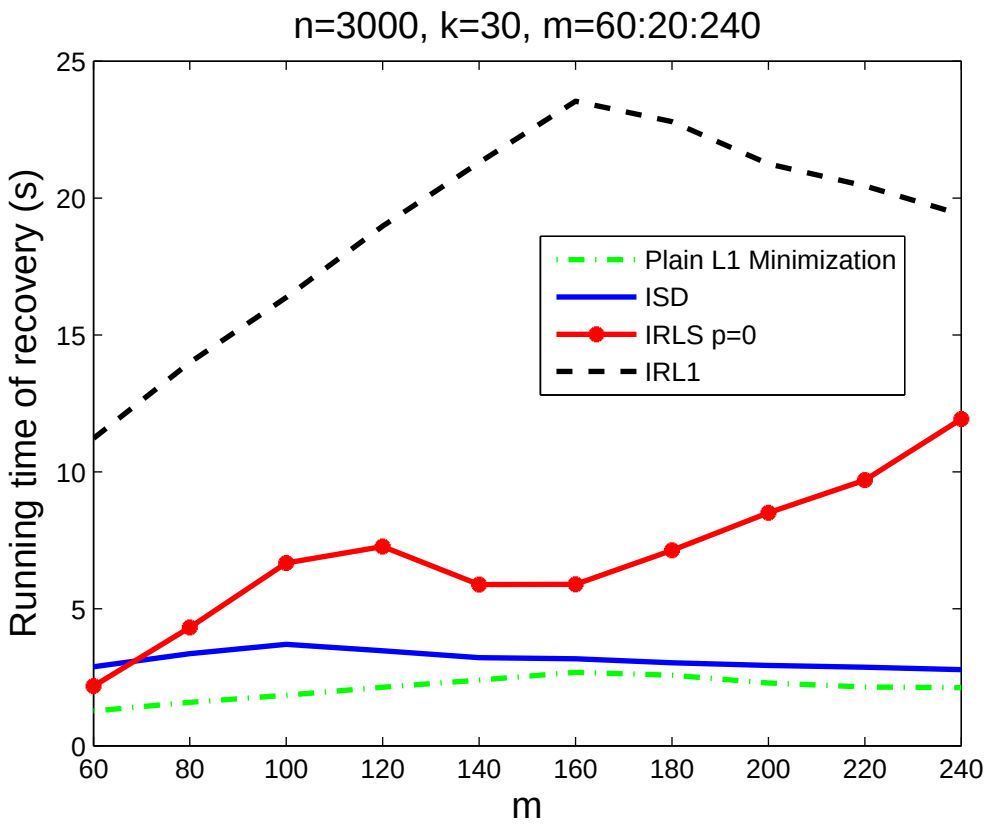

2−th iteration. (total,det,good,bad)=(25,23,23,0), Err = 7.06e−016
true signal
correct recovery
false recovery
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
50
100
150
200

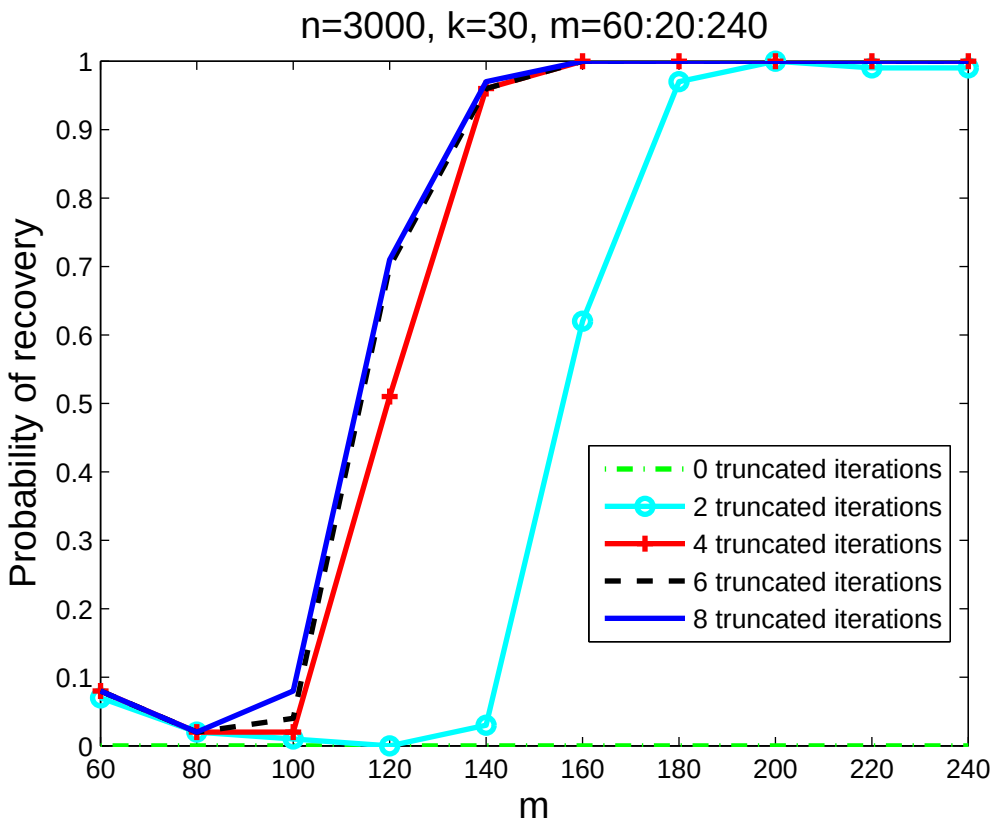

Error = 5.31e−001
original signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
50
100
150
200

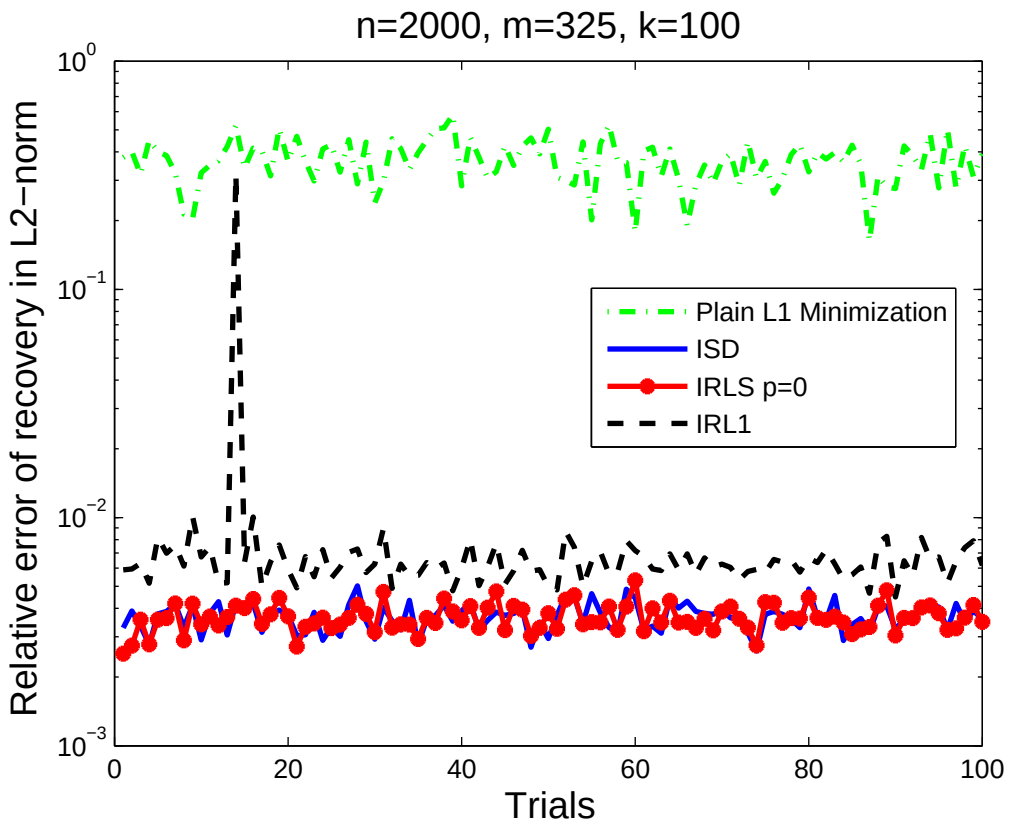

Nonzeros(total,pred,good,bad)=(13,10,8,2), Err = 5.01e−001
true signal
2 reweighted iterations
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
10
20
30
40
50
60

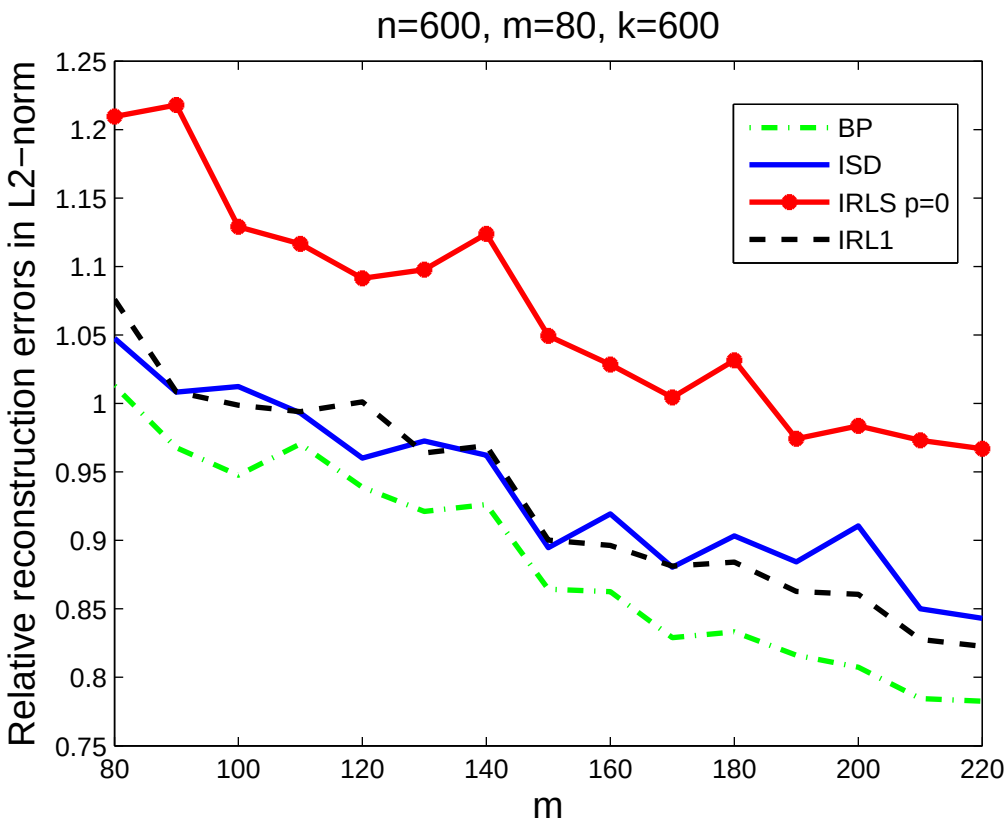

n=600, k=8, m=16:4:100
Exact reconstruction frequency
m
BP
2 truncated iterations
4 truncated iterations
6 truncated iterations
8 truncated iterations
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
10
20
30
40
50
60
70
80
90
100

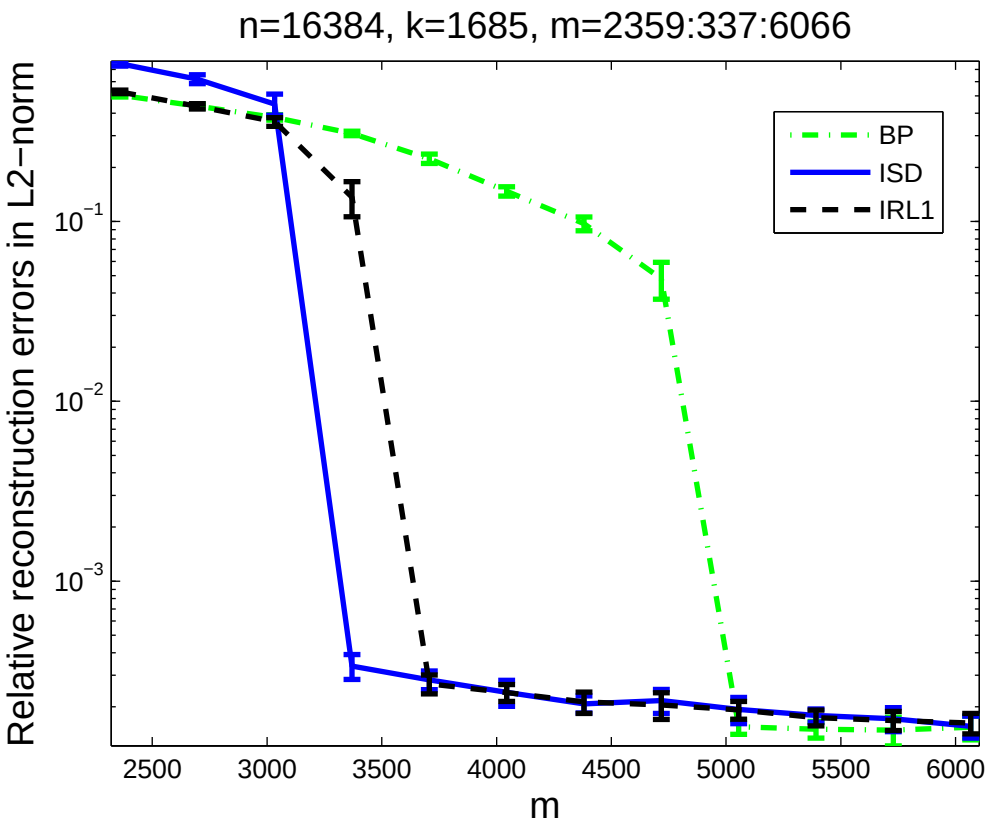

n=600, k=8, m=16:4:100
Acceptable reconstruction frequencies
m
BP
ISD
IRLS
IRL1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
10
20
30
40
50
60
70
80
90
100

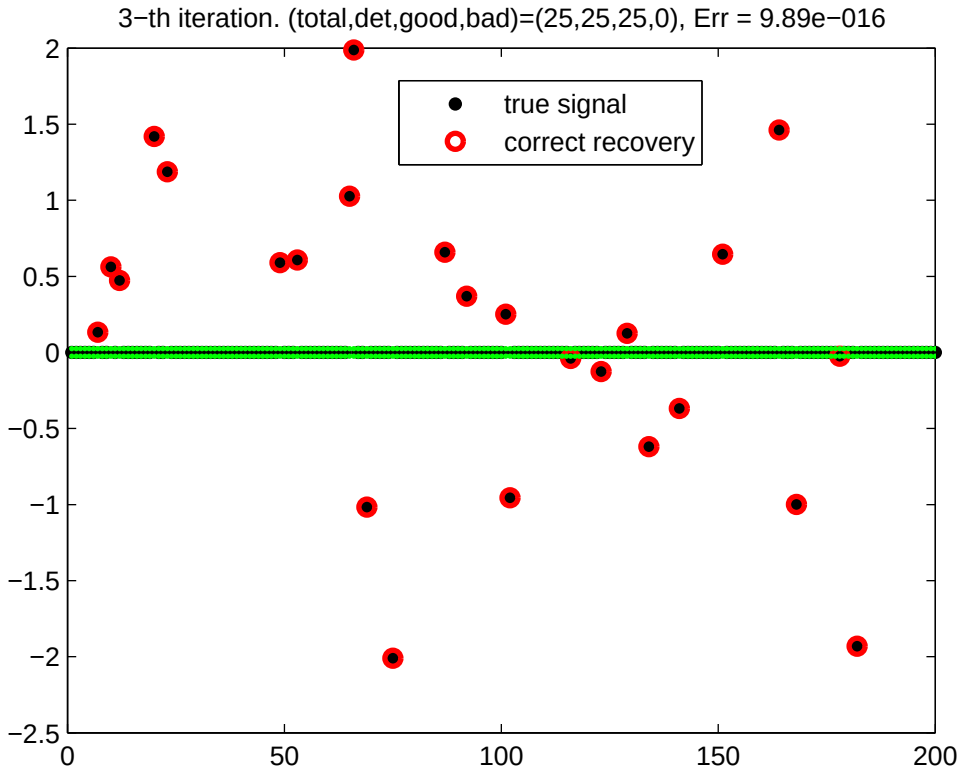

n=3000, k=30, m=60:20:240
Running time of recovery (s)
m
0
5
10
15
20
25
60
80
100
120
140
160
180
200
220
240
Plain L1 Minimization
ISD
IRLS p=0
IRL1

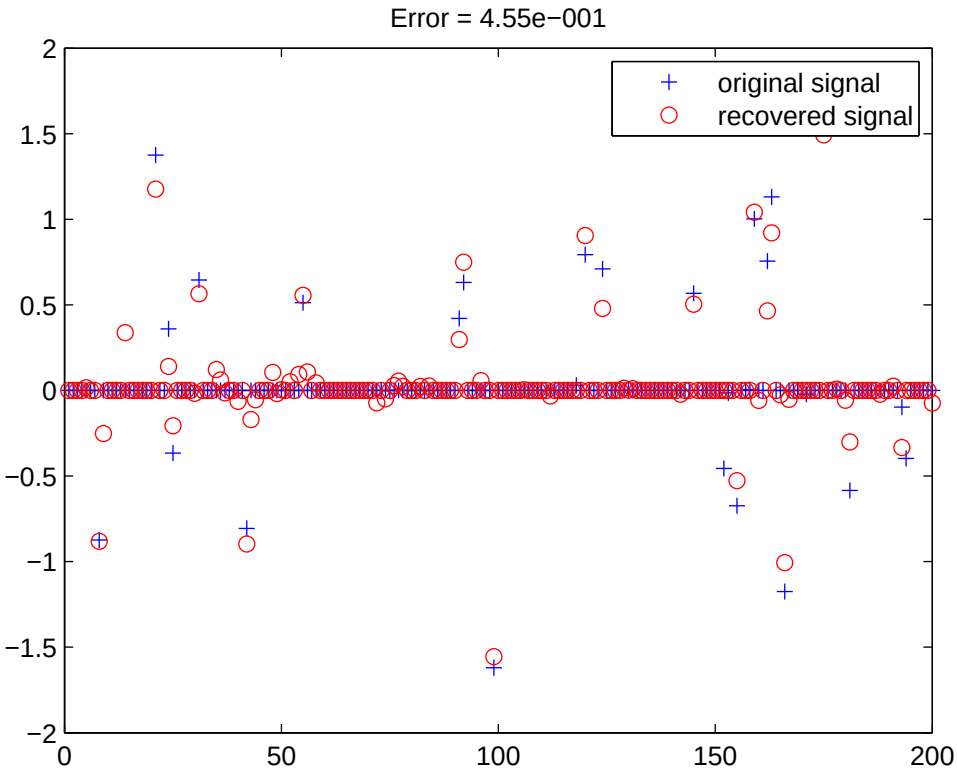

n=3000, k=30, m=60:20:240
Probability of recovery
m
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
60
80
100
120
140
160
180
200
220
240
0 truncated iterations
2 truncated iterations
4 truncated iterations
6 truncated iterations
8 truncated iterations

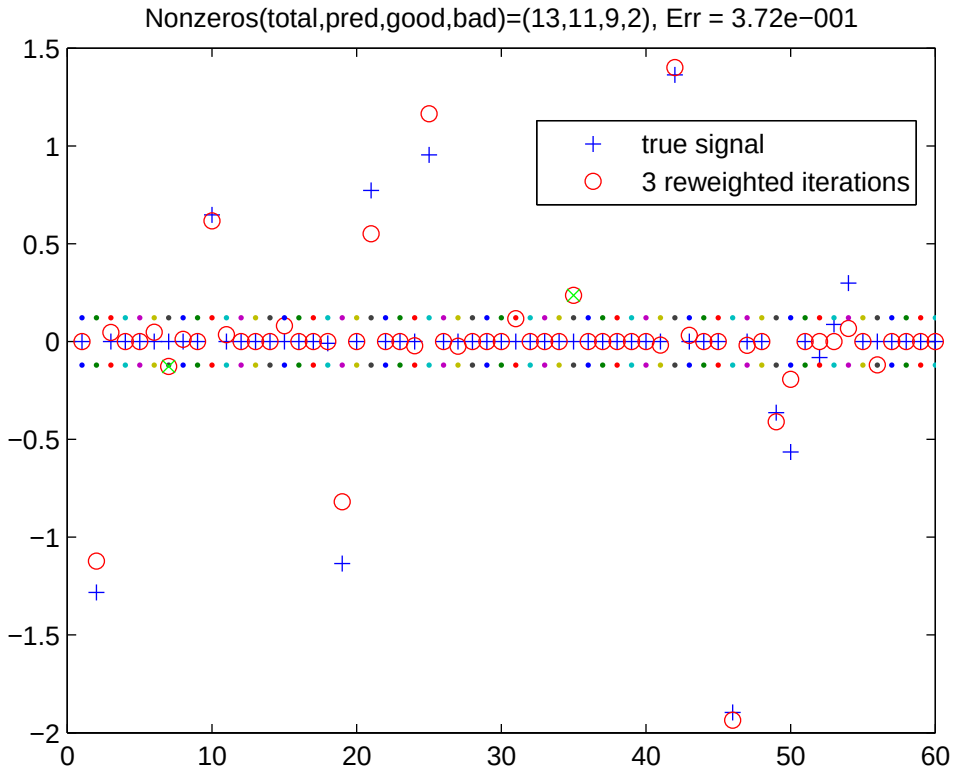

n=2000, m=325, k=100
Relative error of recovery in L2−norm
Trials
Plain L1 Minimization
ISD
IRLS p=0
IRL1
10^0
10^{-1}
10^{-2}
10^{-3}
0
20
40
60
80
100

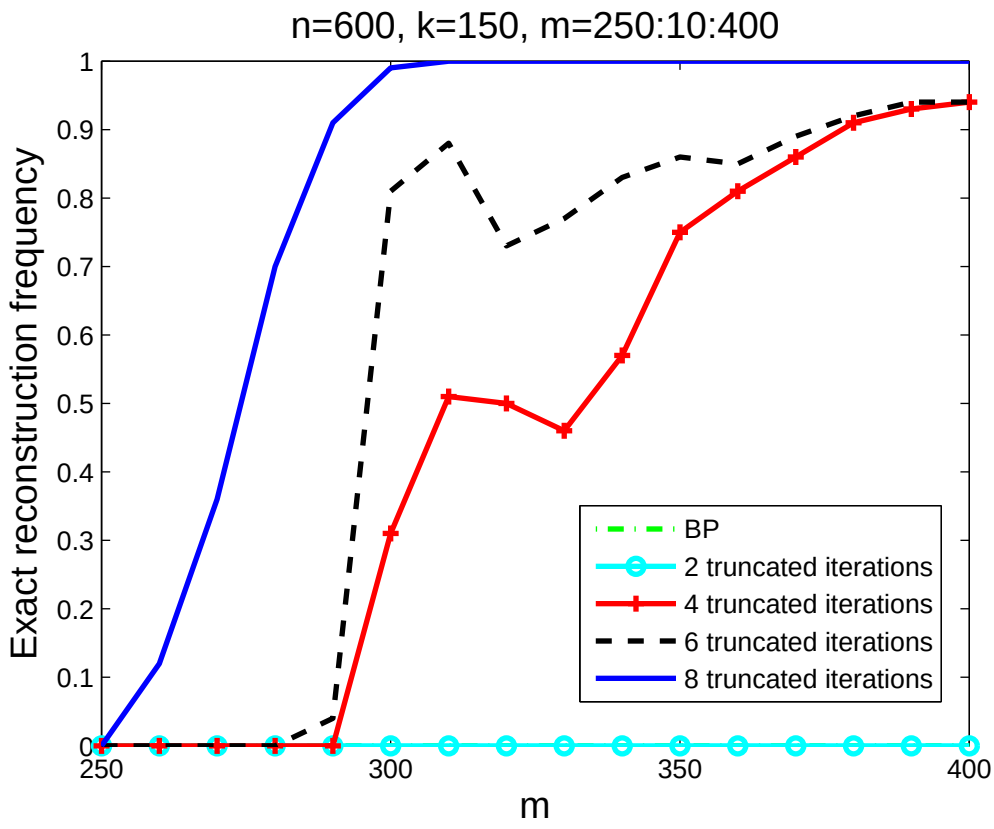

n=600, m=80, k=600
Relative reconstruction errors in L2−norm
m
BP
ISD
IRLS p=0
IRL1
1.25
1.2
1.15
1.1
1.05
1
0.95
0.9
0.85
0.8
0.75
80
100
120
140
160
180
200
220

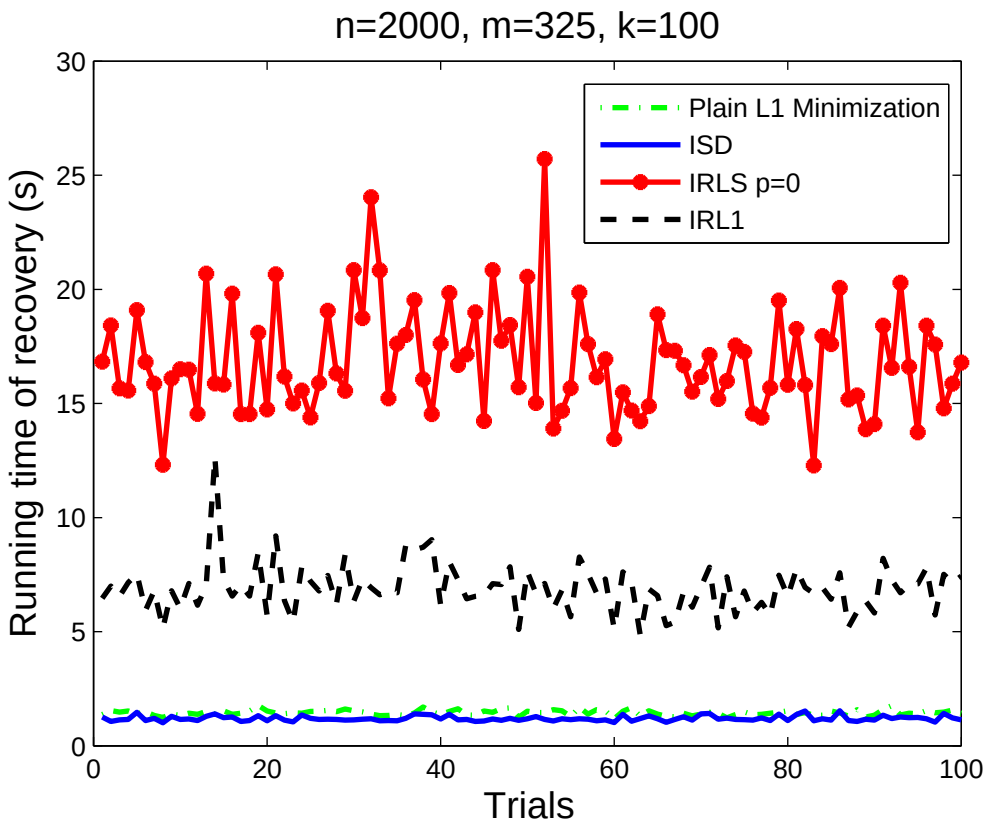

n=16384, k=1685, m=2359:337:6066
Relative reconstruction errors in L2−norm
m
BP
ISD
IRL1
10^{-1}
10^{-2}
10^{-3}
2500
3000
3500
4000
4500
5000
5500
6000

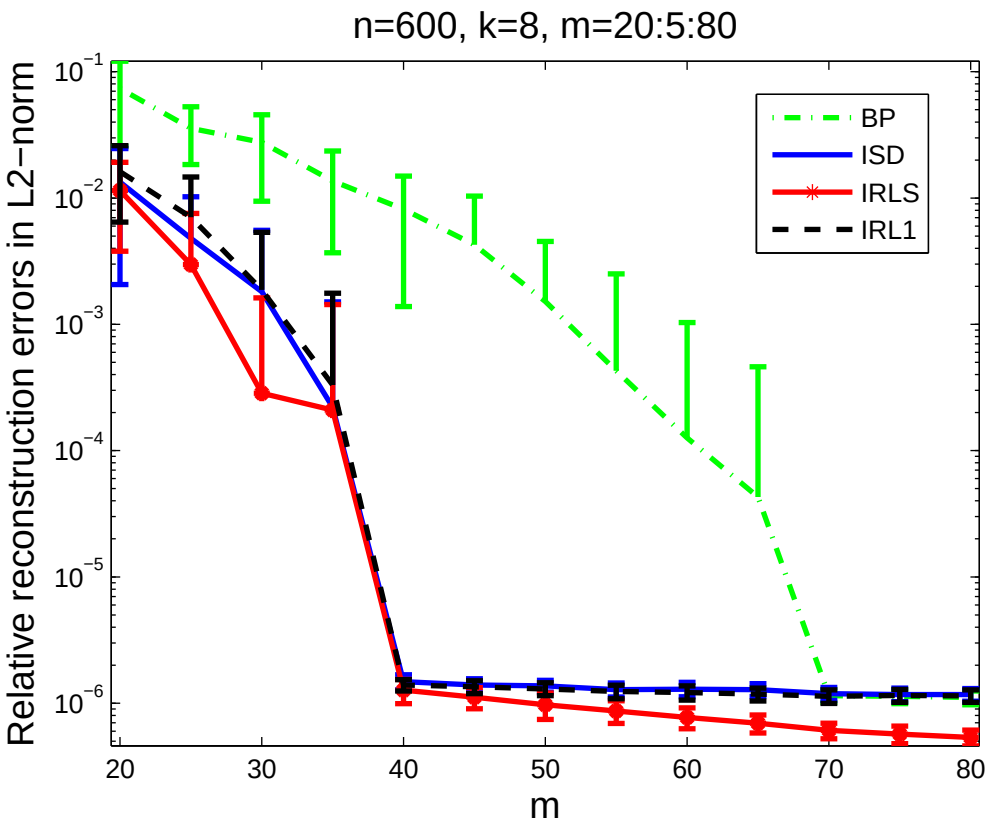

3−th iteration. (total,det,good,bad)=(25,25,25,0), Err = 9.89e−016
true signal
correct recovery
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
50
100
150
200

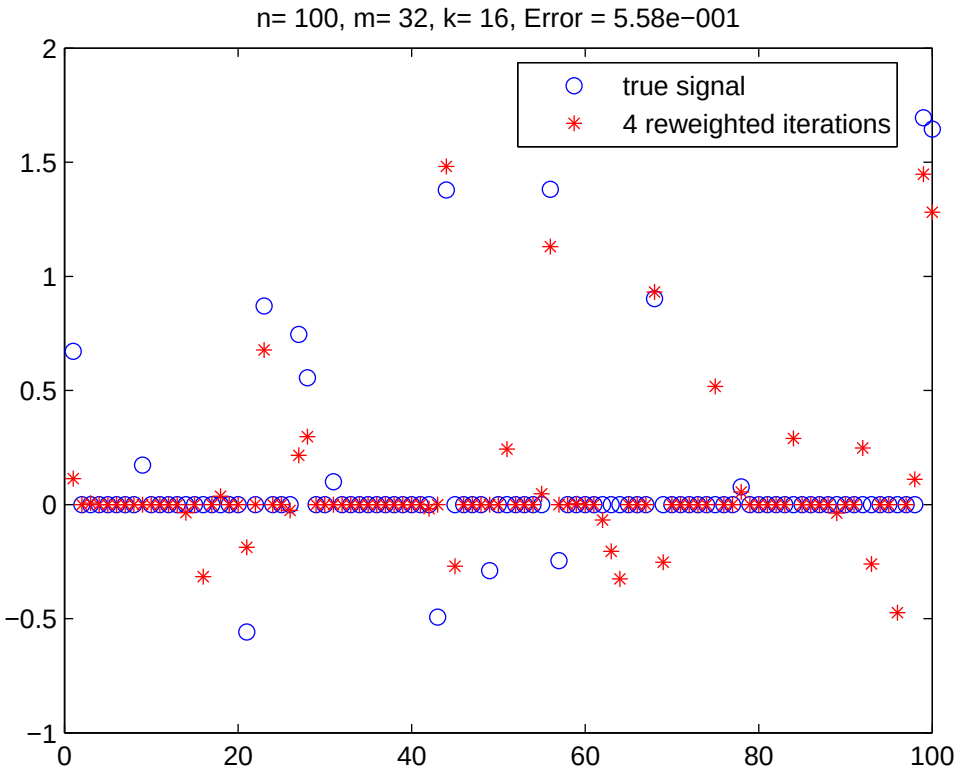

Error = 4.55e−001
original signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
50
100
150
200

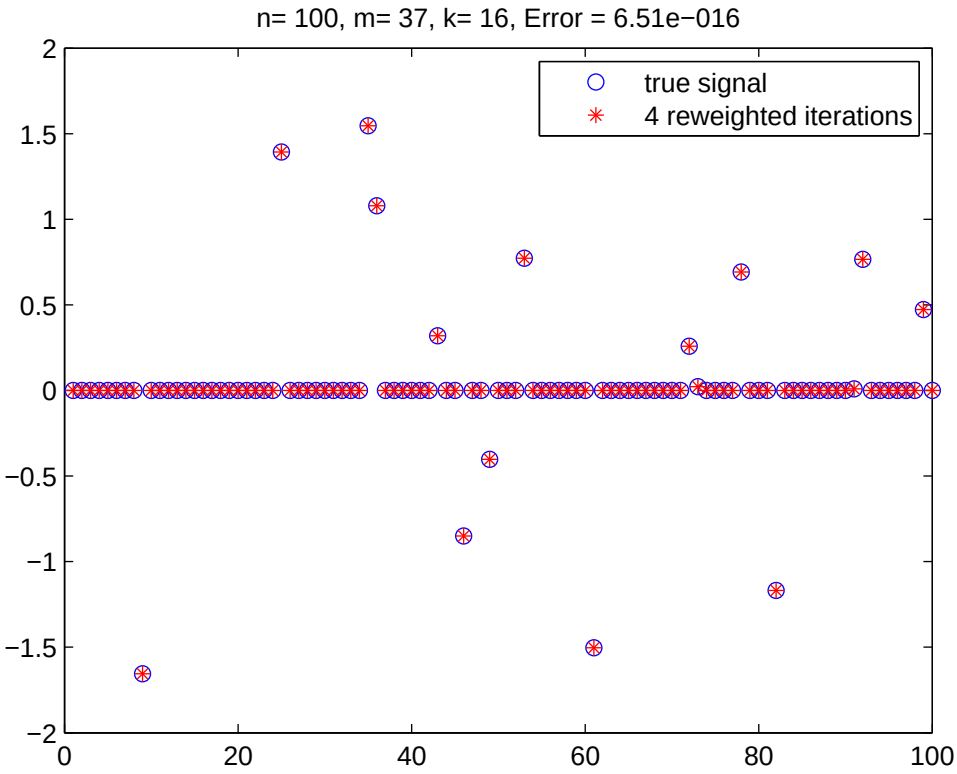

Nonzeros(total,pred,good,bad)=(13,11,9,2), Err = 3.72e−001
true signal
3 reweighted iterations
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
10
20
30
40
50
60

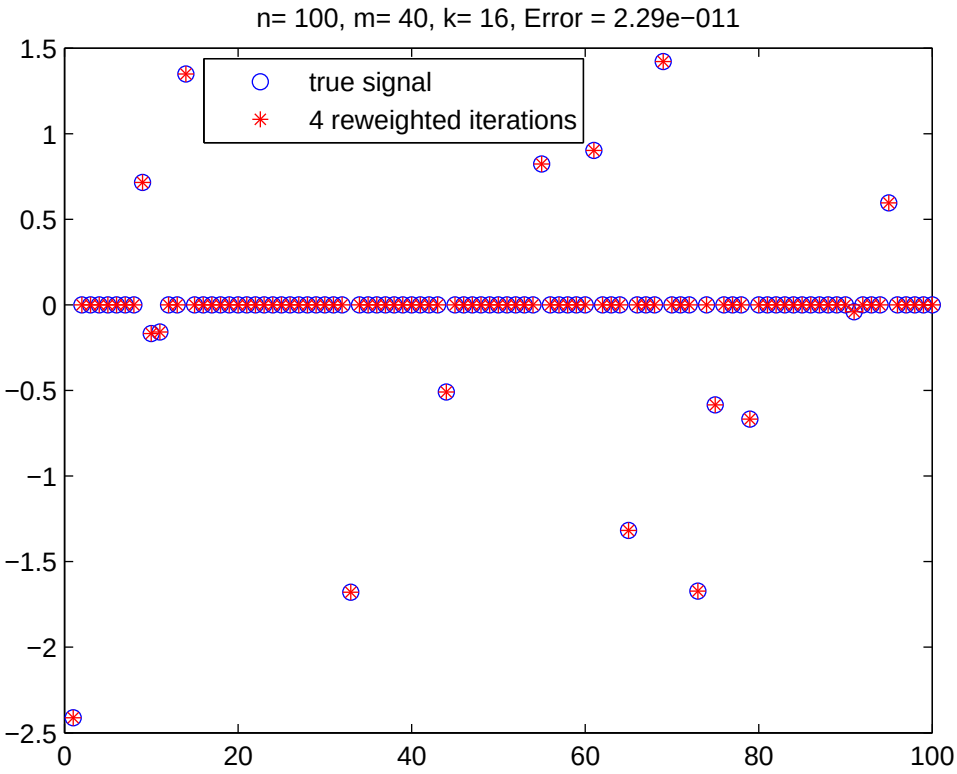

n=600, k=150, m=250:10:400
Exact reconstruction frequency
m
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
250
300
350
400
BP
2 truncated iterations
4 truncated iterations
6 truncated iterations
8 truncated iterations

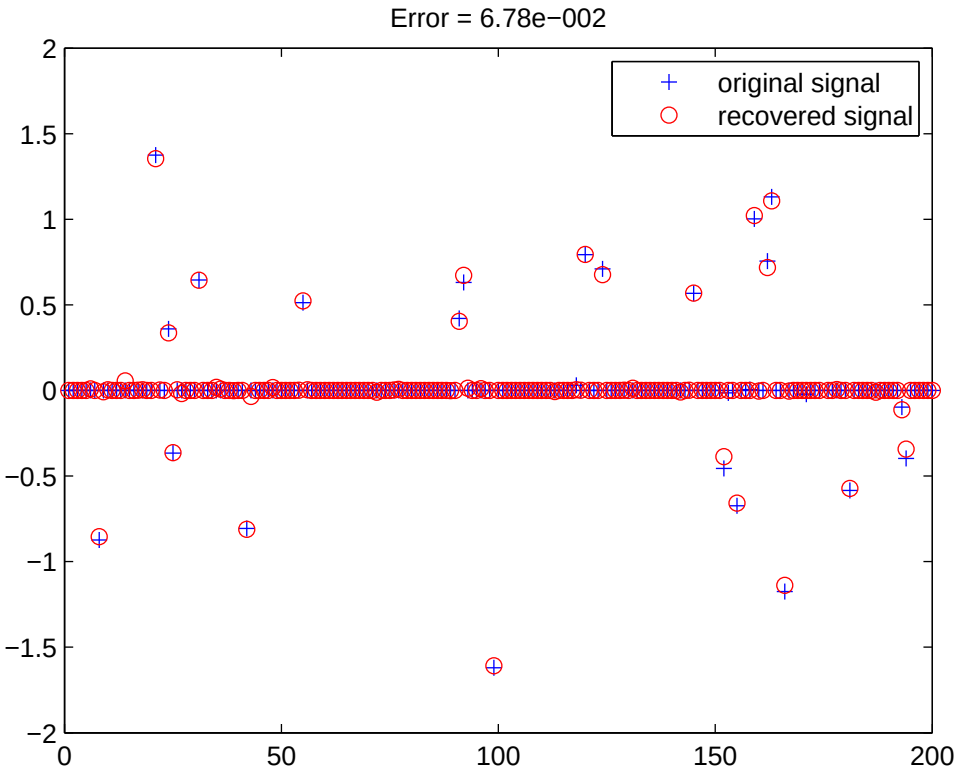

n=2000, m=325, k=100
Running time of recovery (s)
Trials
Plain L1 Minimization
ISD
IRLS p=0
IRL1
0
5
10
15
20
25
30
0
20
40
60
80
100

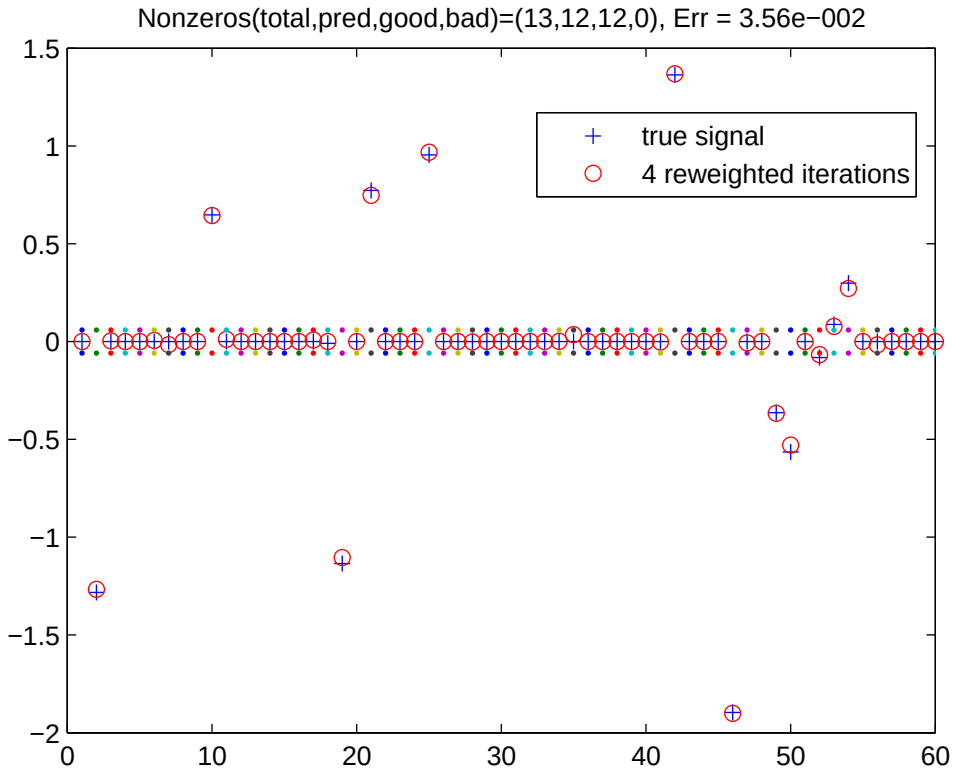

n=600, k=8, m=20:5:80
Relative reconstruction errors in L2−norm
m
BP
ISD
IRLS
IRL1
10^−1
10^−2
10^−3
10^−4
10^−5
10^−6
20
30
40
50
60
70
80

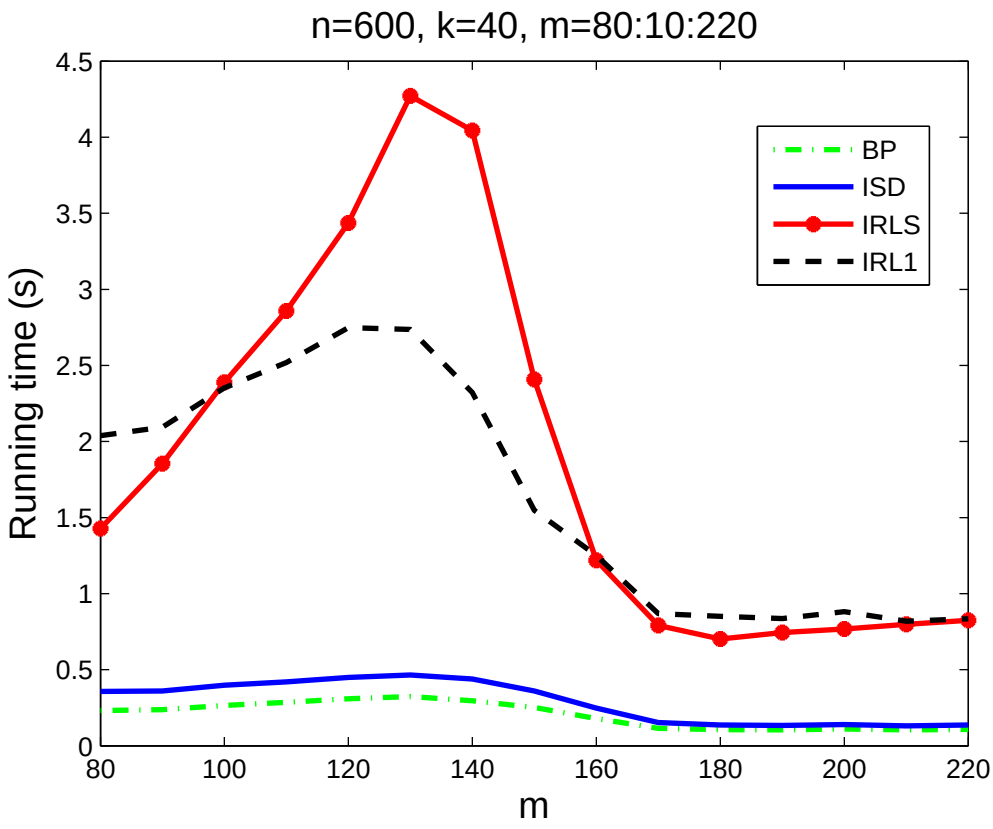

n= 100, m= 32, k= 16, Error = 5.58e−001
true signal
4 reweighted iterations
2
1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

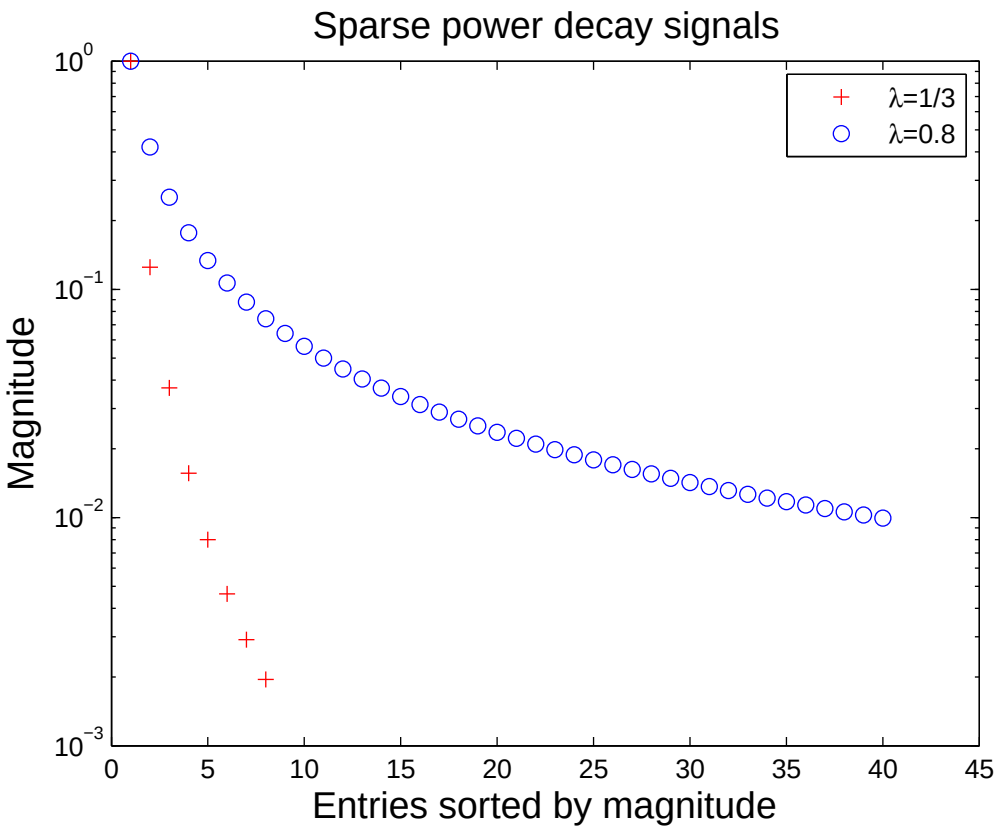

n= 100, m= 37, k= 16, Error = 6.51e−016
true signal
4 reweighted iterations
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
20
40
60
80
100

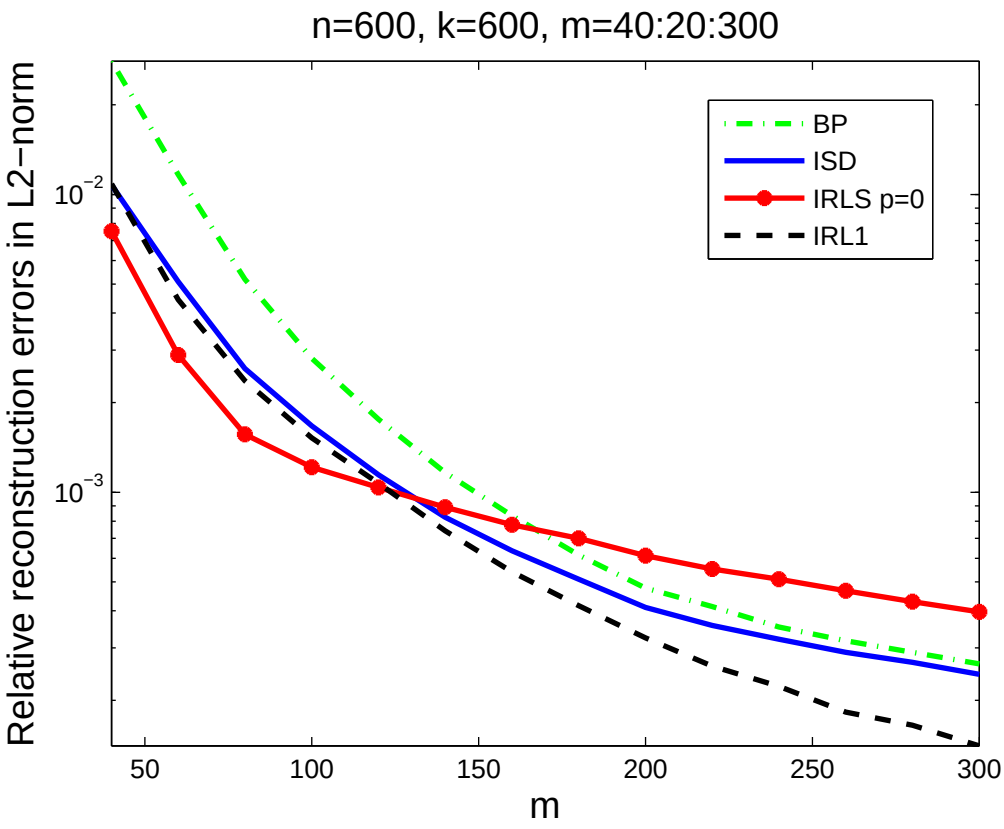

n= 100, m= 40, k= 16, Error = 2.29e−011
true signal
4 reweighted iterations
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
20
40
60
80
100

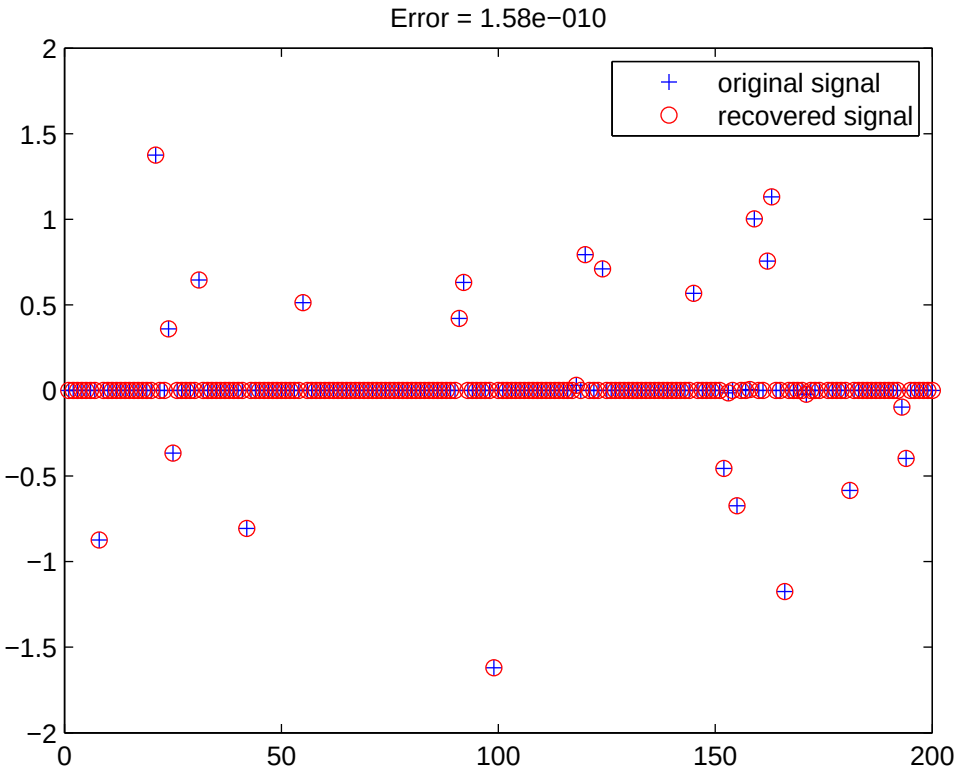

Error = 6.78e−002
original signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
50
100
150
200

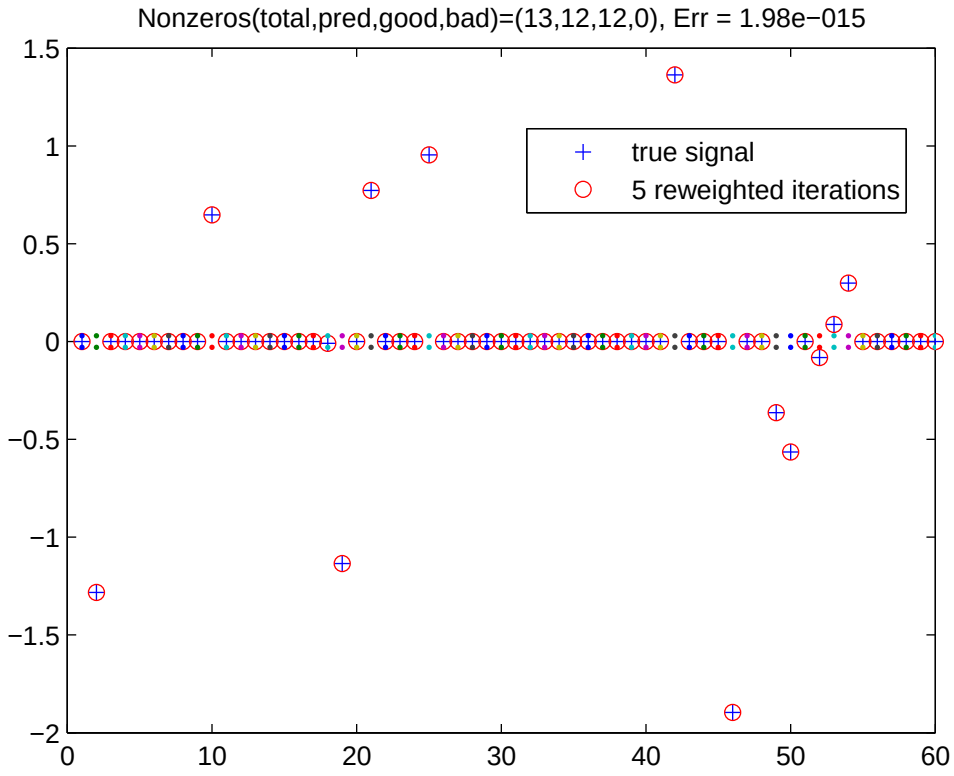

Nonzeros(total,pred,good,bad)=(13,12,12,0), Err = 3.56e−002
true signal
4 reweighted iterations
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
10
20
30
40
50
60


n=600, k=40, m=80:10:220
Running time (s)
m
BP
ISD
IRLS
IRL1
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
80
100
120
140
160
180
200
220


Sparse power decay signals
Magnitude
Entries sorted by magnitude
λ=1/3
λ=0.8
10^0
10^-1
10^-2
10^-3
0
5
10
15
20
25
30
35
40
45

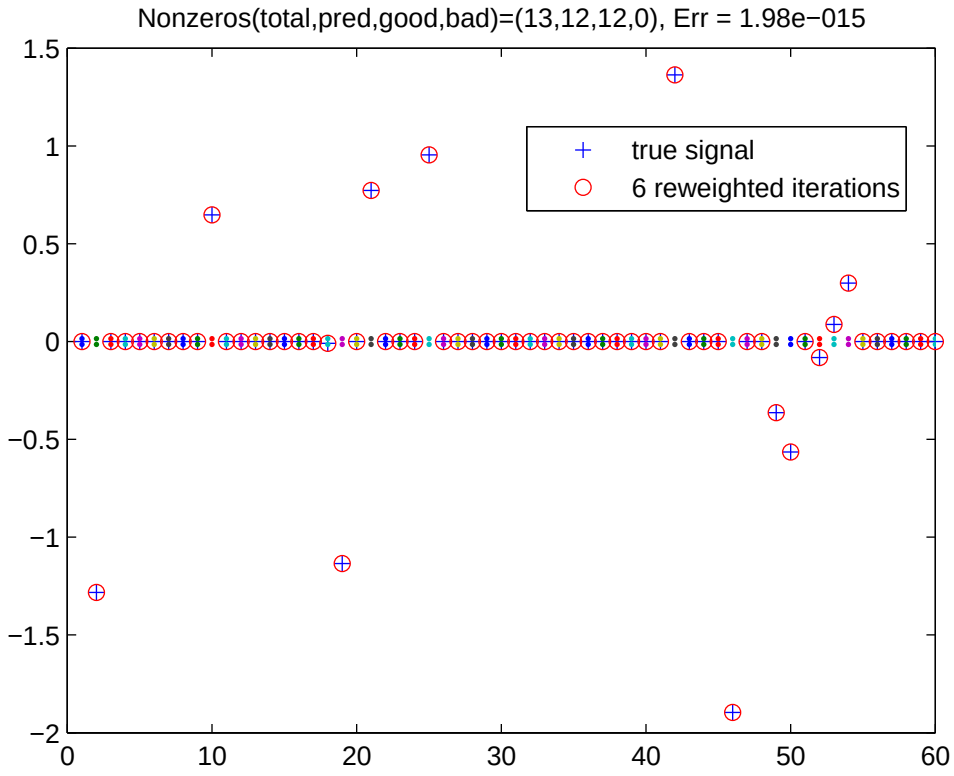

n=600, k=600, m=40:20:300
Relative reconstruction errors in L2−norm
10^{-2}
10^{-3}
50
100
150
200
250
300
m
BP
ISD
IRLS p=0
IRL1

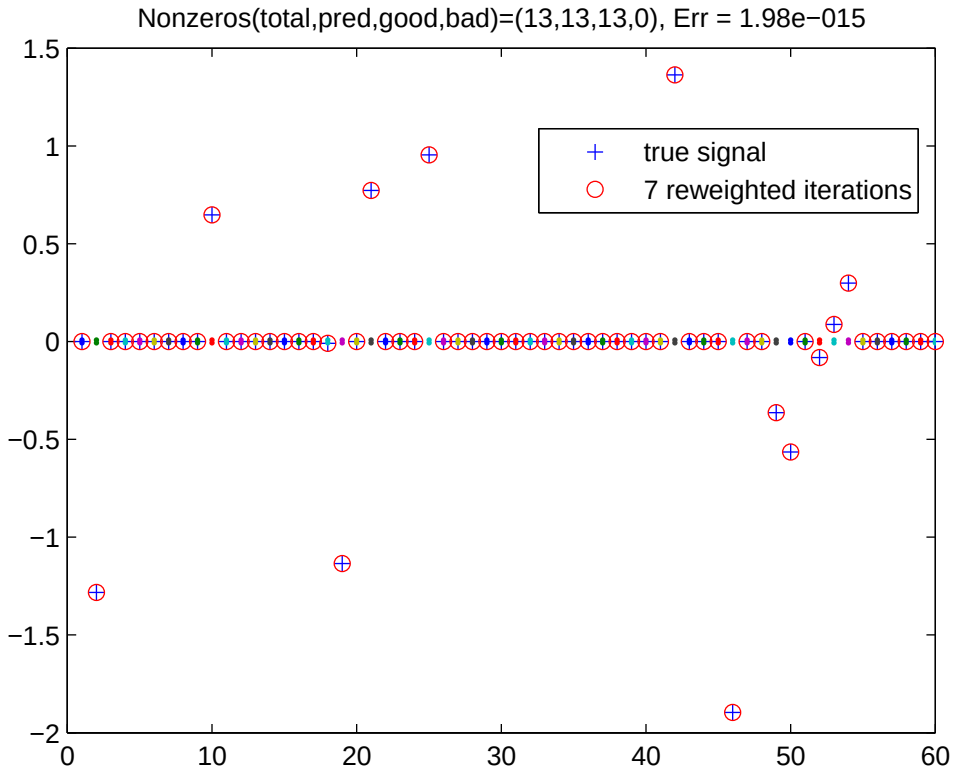

Error = 1.58e−010
original signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
50
100
150
200

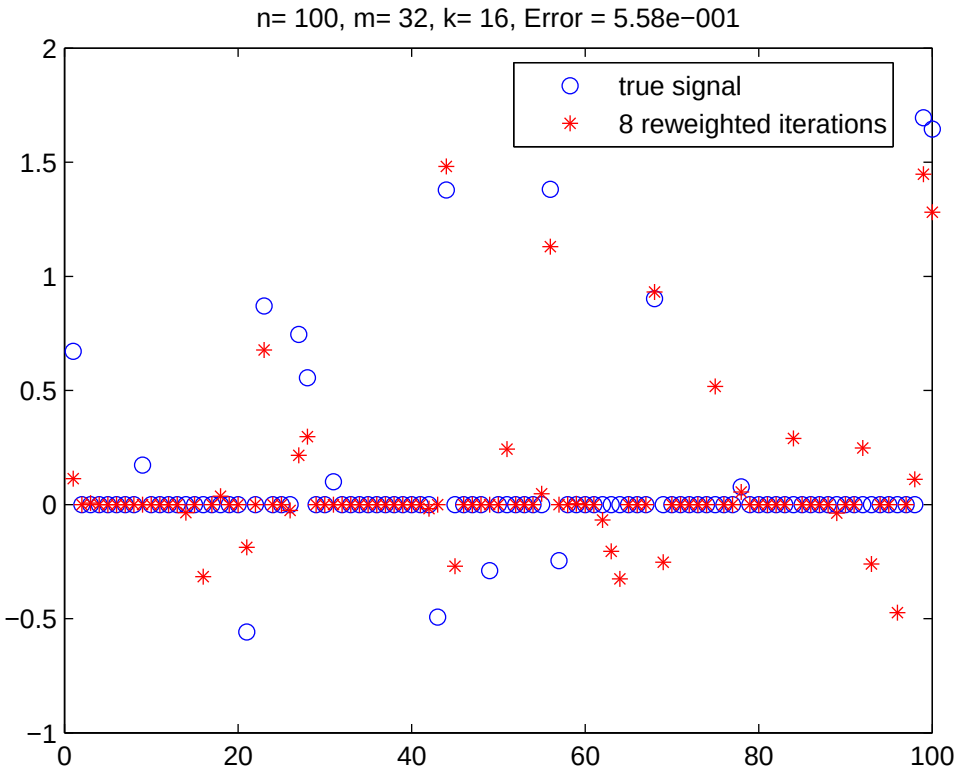

Nonzeros(total,pred,good,bad)=(13,12,12,0), Err = 1.98e−015
true signal
5 reweighted iterations
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
10
20
30
40
50
60

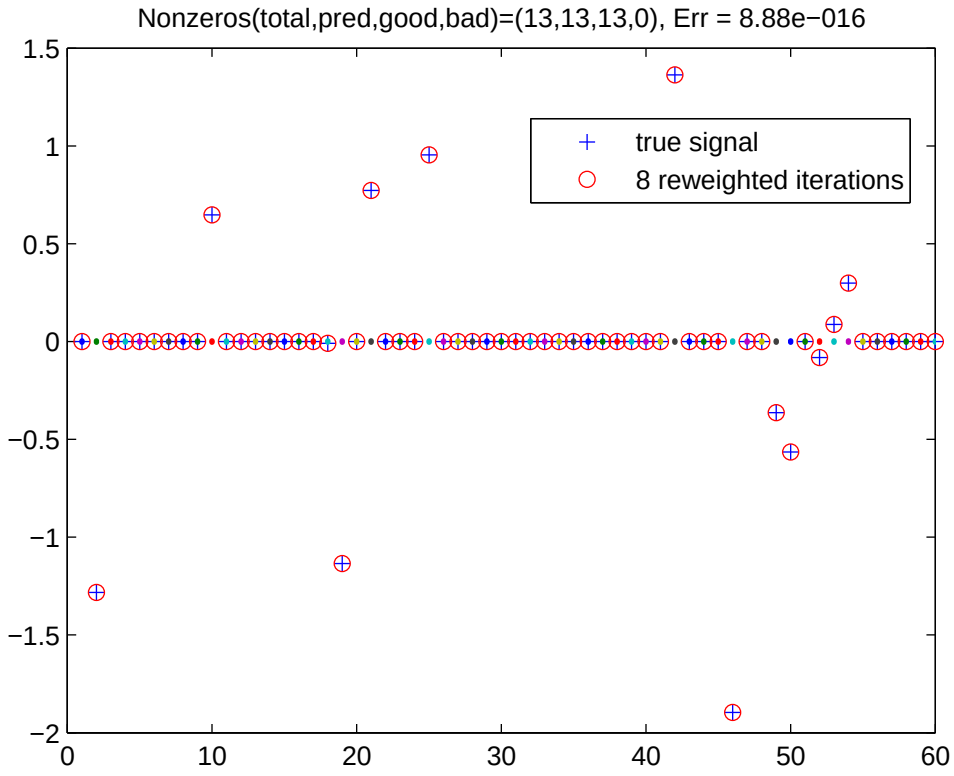
ISD: SNR=42.63dB, Err=6.40e−03, CPU time=80.00 s

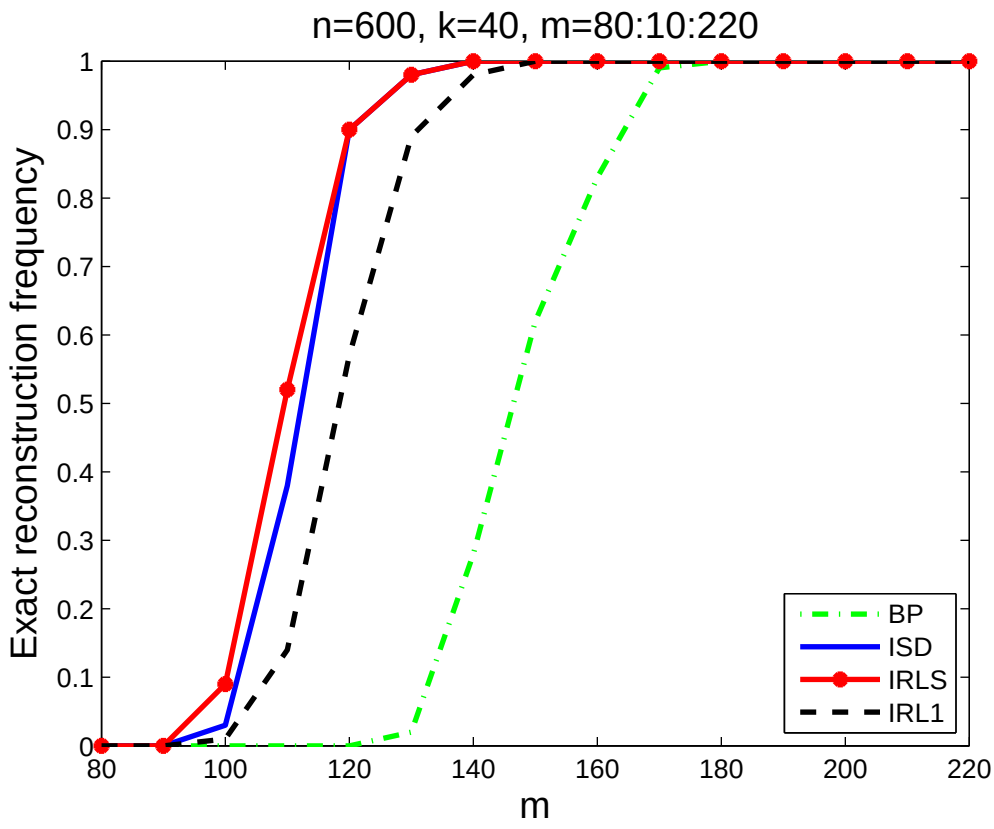
18% data

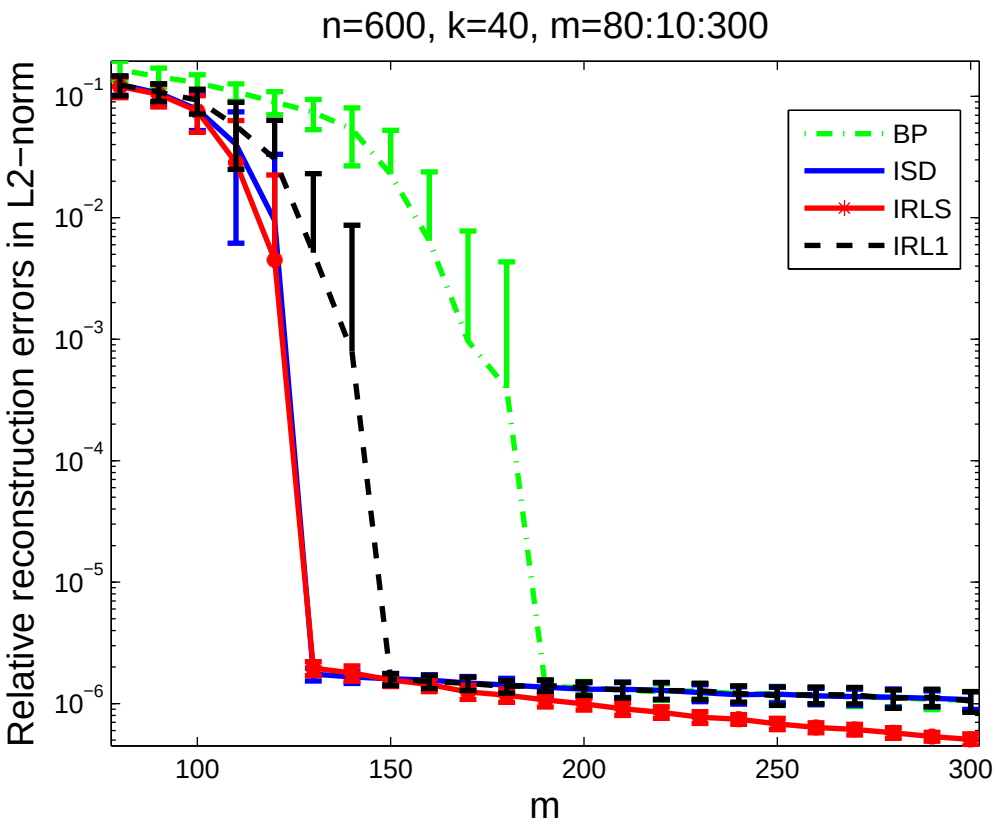

Nonzeros(total,pred,good,bad)=(13,12,12,0), Err = 1.98e−015
true signal
6 reweighted iterations
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
10
20
30
40
50
60

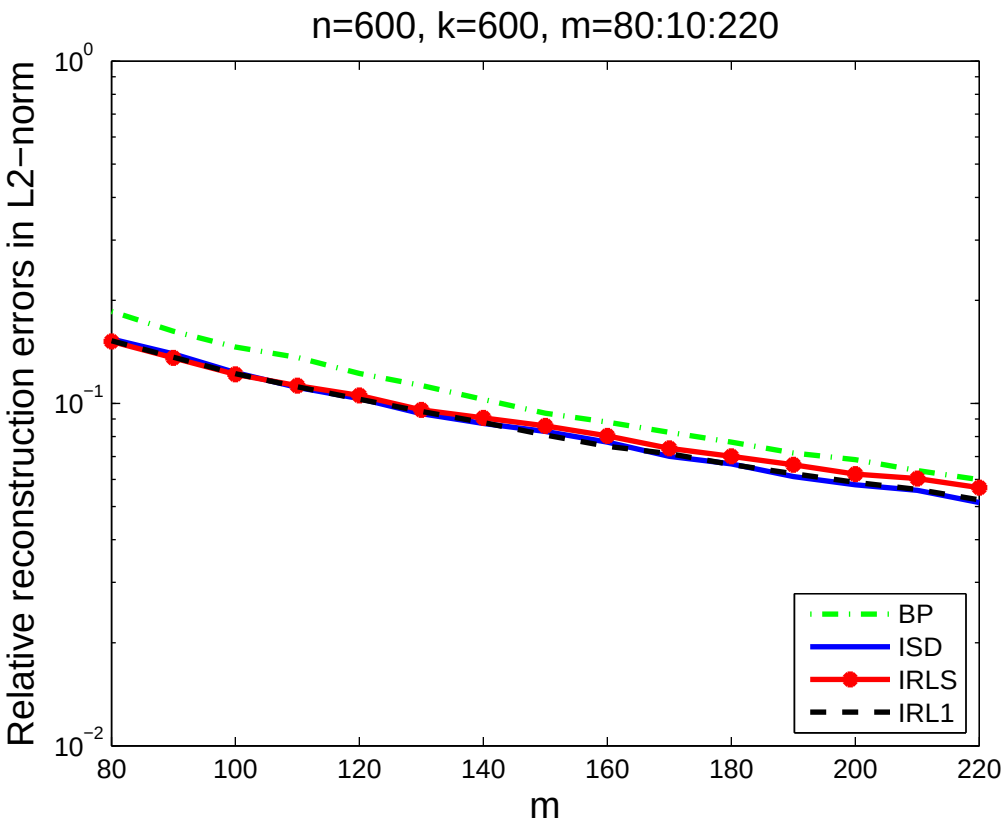

Nonzeros(total,pred,good,bad)=(13,13,13,0), Err = 1.98e−015
true signal
7 reweighted iterations
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
10
20
30
40
50
60

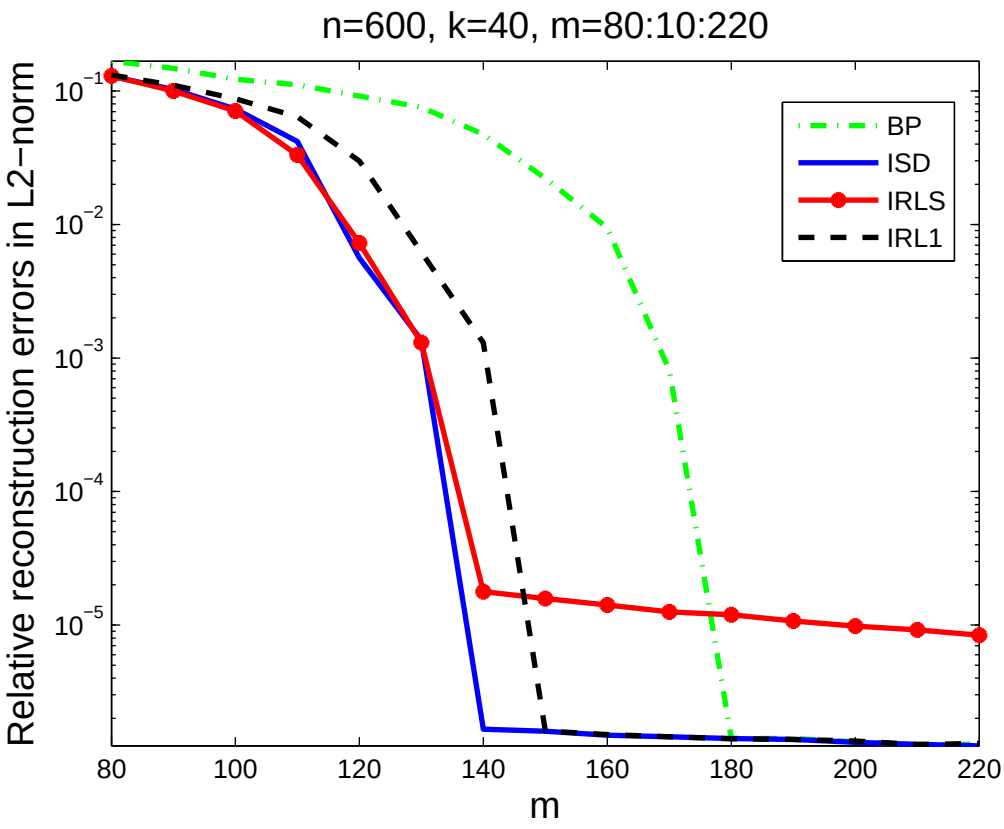

n= 100, m= 32, k= 16, Error = 5.58e−001
true signal
8 reweighted iterations
2
1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

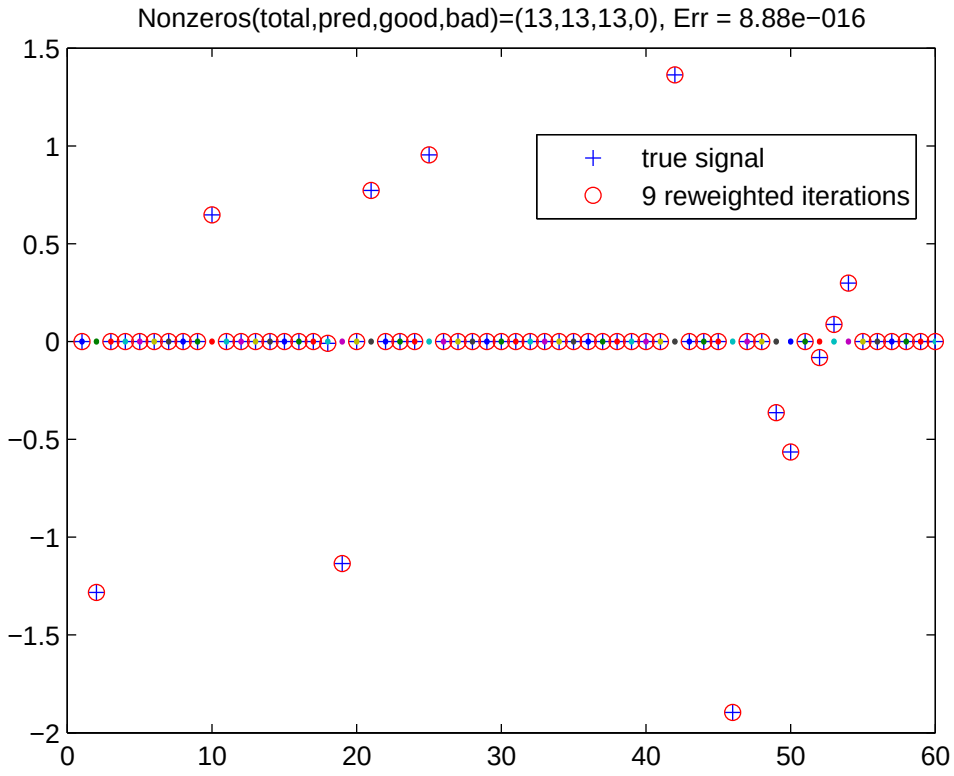

Nonzeros(total,pred,good,bad)=(13,13,13,0), Err = 8.88e−016
true signal
8 reweighted iterations
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
10
20
30
40
50
60

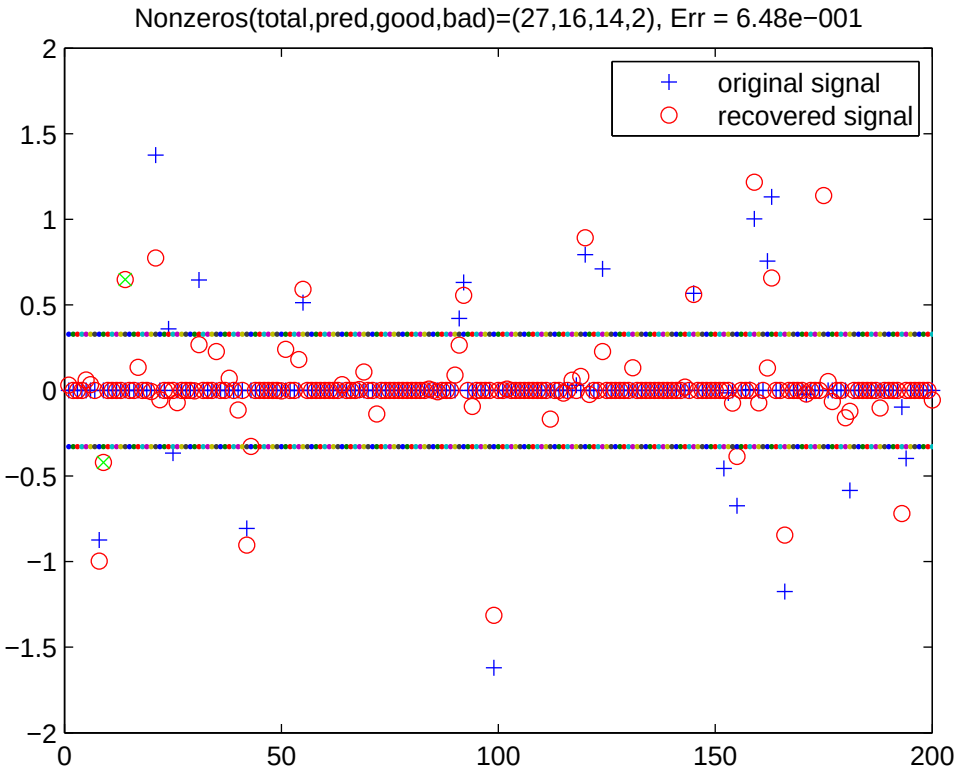

n=600, k=40, m=80:10:220
Exact reconstruction frequency
m
BP
ISD
IRLS
IRL1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
80
100
120
140
160
180
200
220

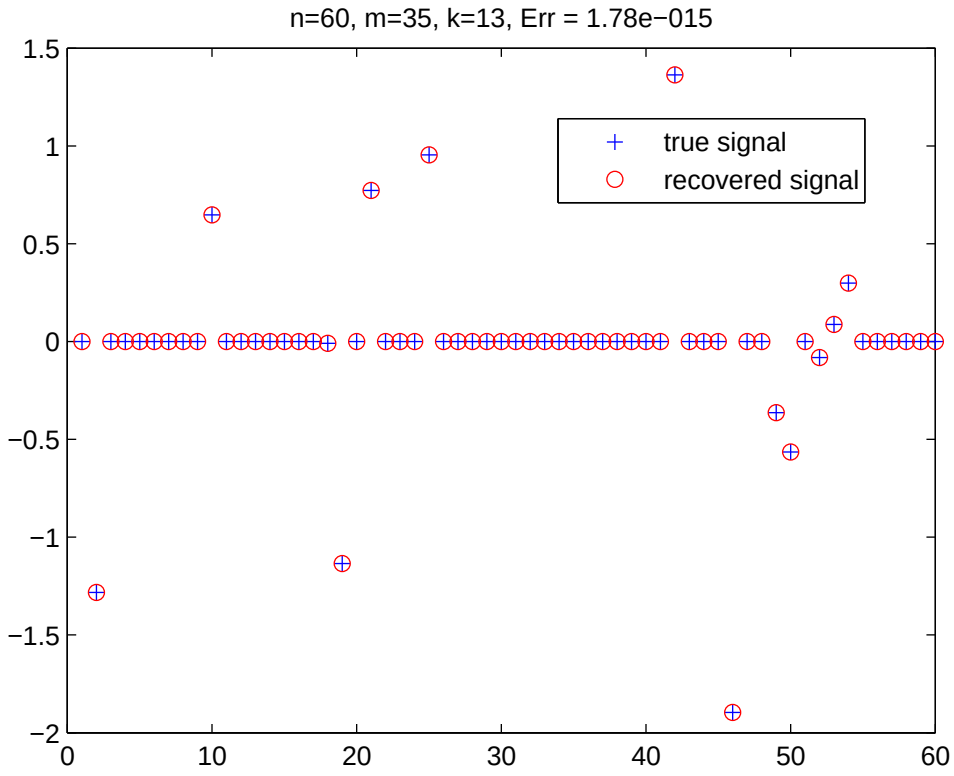

n=600, k=40, m=80:10:300
Relative reconstruction errors in L2−norm
m
BP
ISD
IRLS
IRL1
10^−1
10^−2
10^−3
10^−4
10^−5
10^−6
100
150
200
250
300

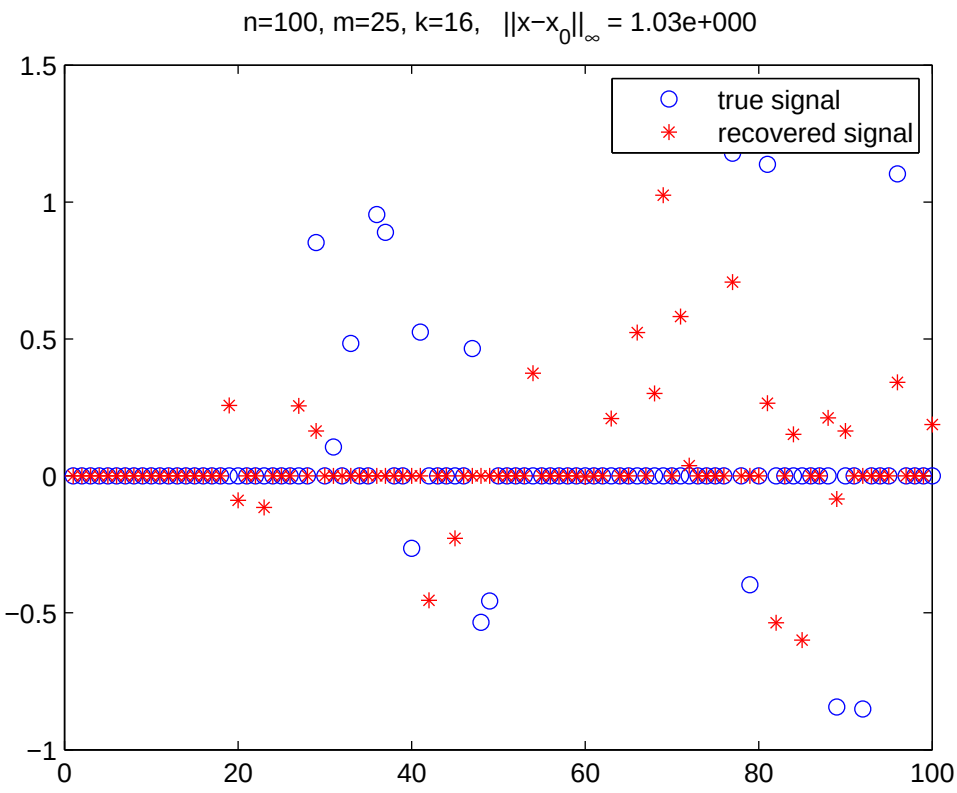

n=600, k=600, m=80:10:220
Relative reconstruction errors in L2−norm
m
BP
ISD
IRLS
IRL1
10^0
10^−1
10^−2
80
100
120
140
160
180
200
220

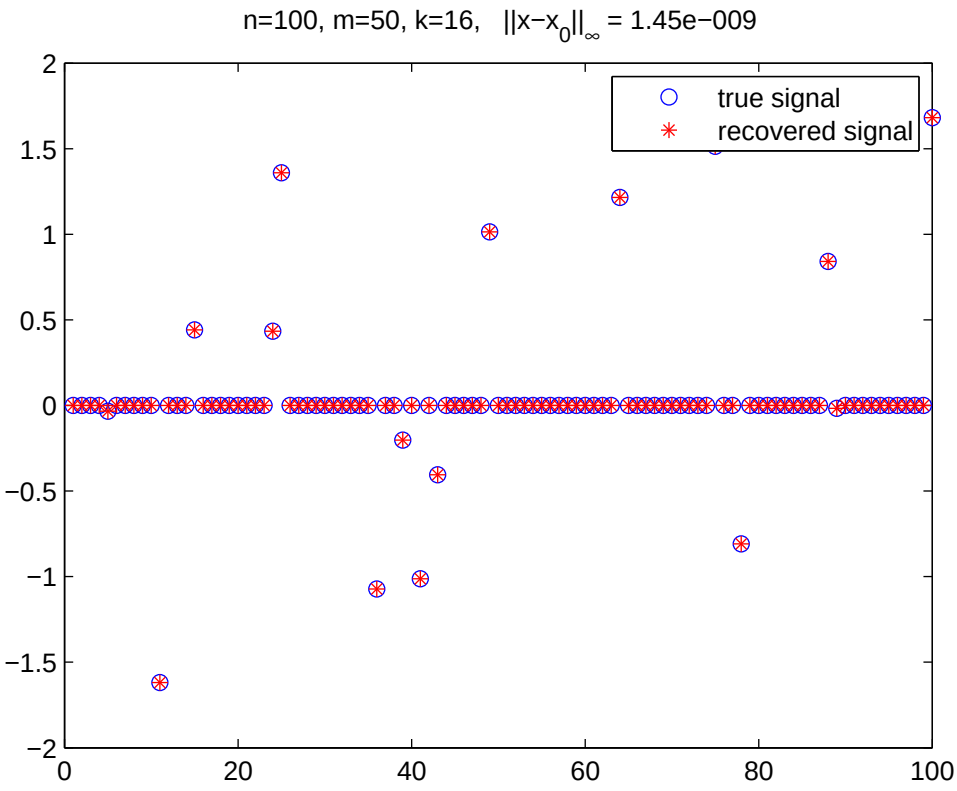

n=600, k=40, m=80:10:220
Relative reconstruction errors in L2−norm
m
BP
ISD
IRLS
IRL1
10^−1
10^−2
10^−3
10^−4
10^−5
80
100
120
140
160
180
200
220

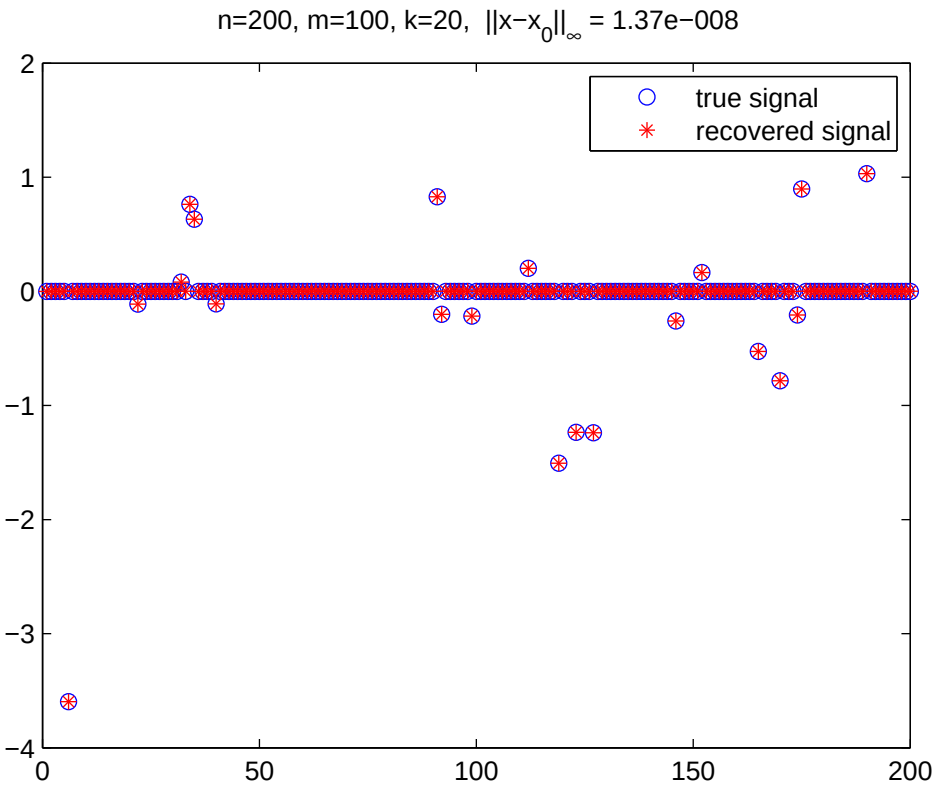

Nonzeros(total,pred,good,bad)=(13,13,13,0), Err = 8.88e−016
true signal
9 reweighted iterations
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
10
20
30
40
50
60

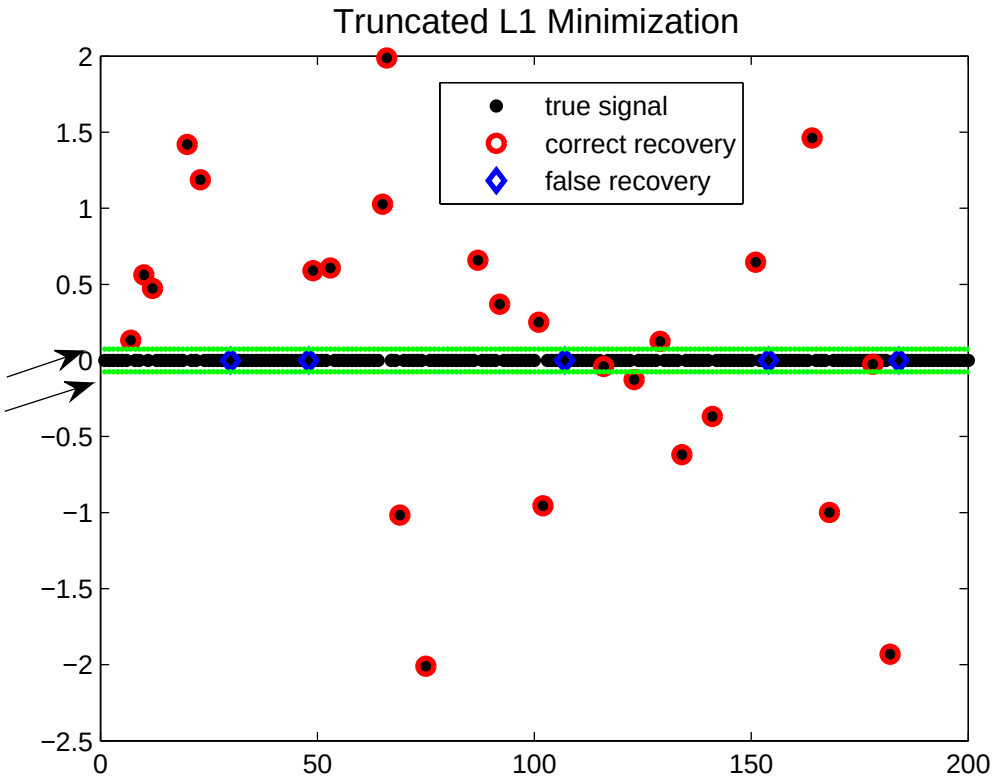

Nonzeros(total,pred,good,bad)=(27,16,14,2), Err = 6.48e−001
original signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
50
100
150
200

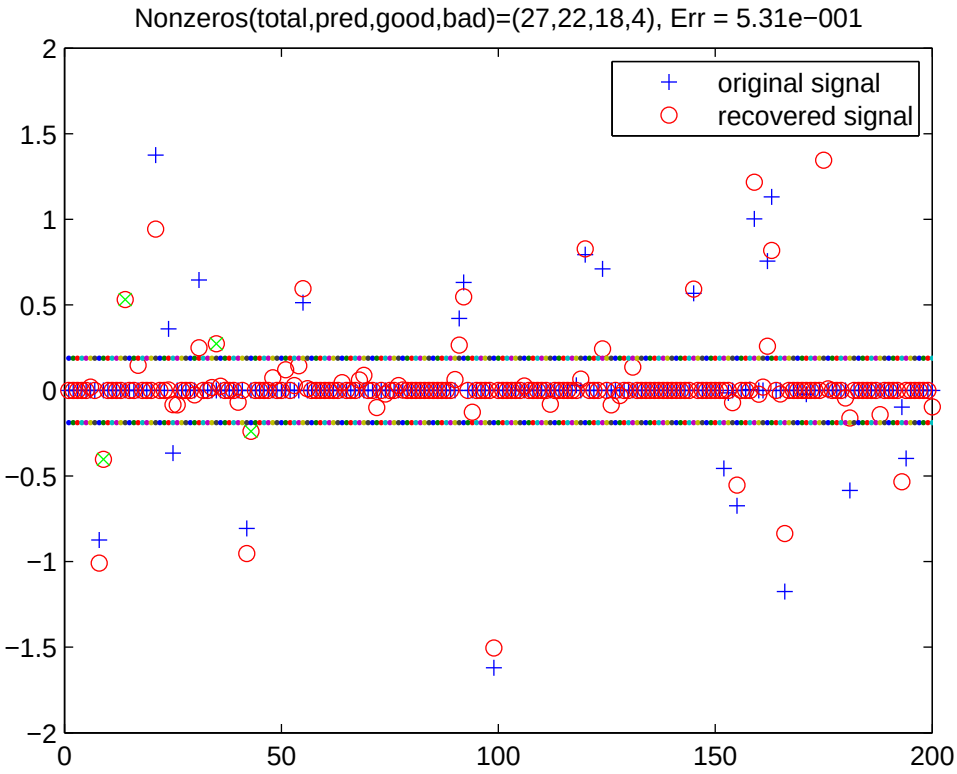

n=60, m=35, k=13, Err = 1.78e−015
true signal
recovered signal
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
10
20
30
40
50
60

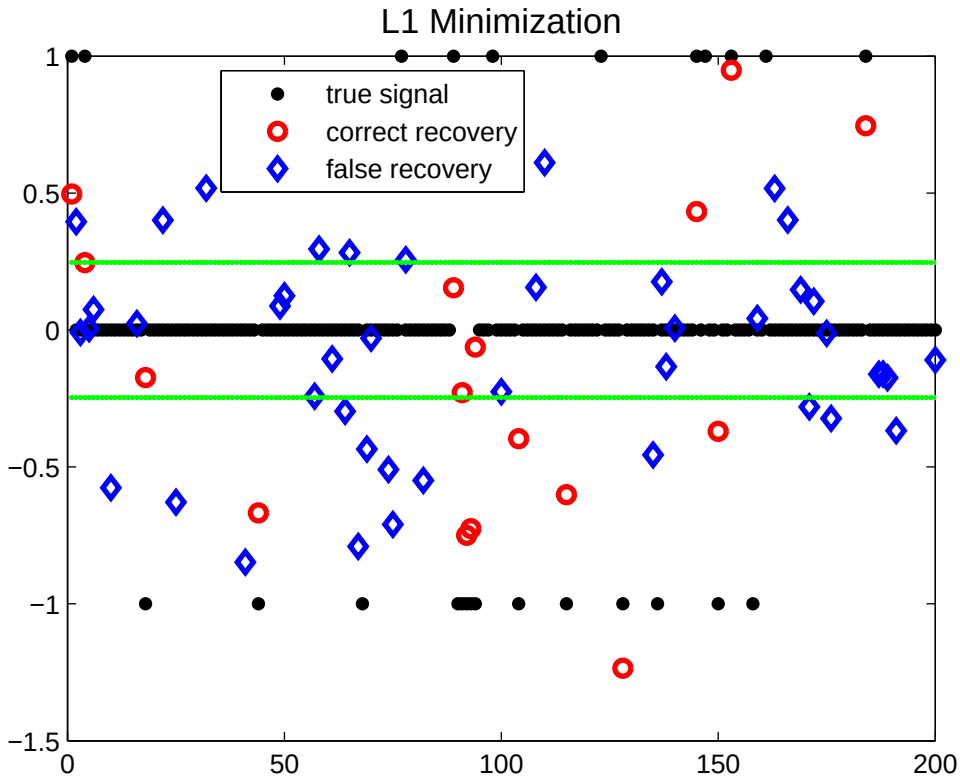

n=100, m=25, k=16, ||x−x0||∞ = 1.03e+000
true signal
recovered signal
1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

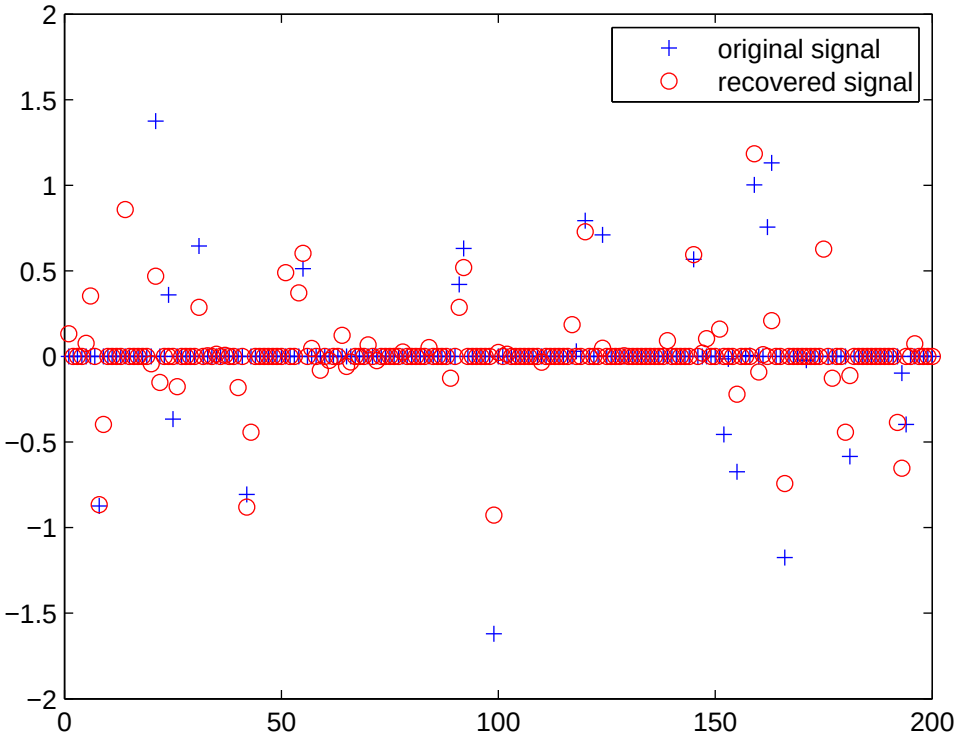

n=100, m=50, k=16, ||x−x0||∞ = 1.45e−009
true signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
20
40
60
80
100

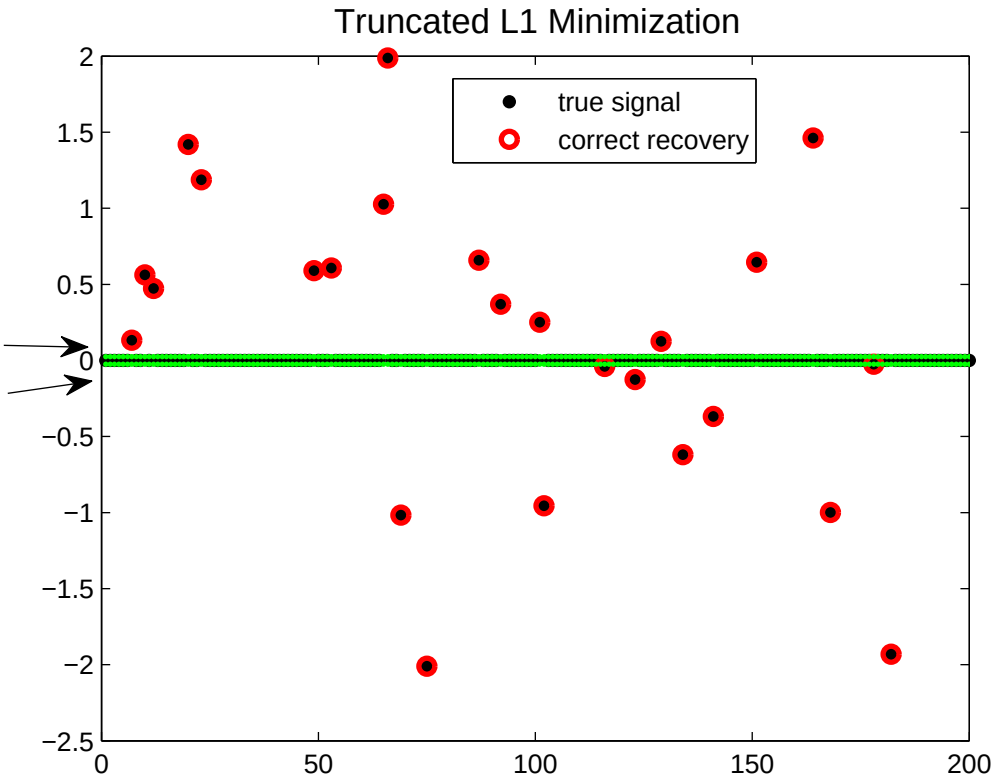

n=200, m=100, k=20, ||x−x0||∞ = 1.37e−008
true signal
recovered signal
2
1
0
−1
−2
−3
−4
0
50
100
150
200

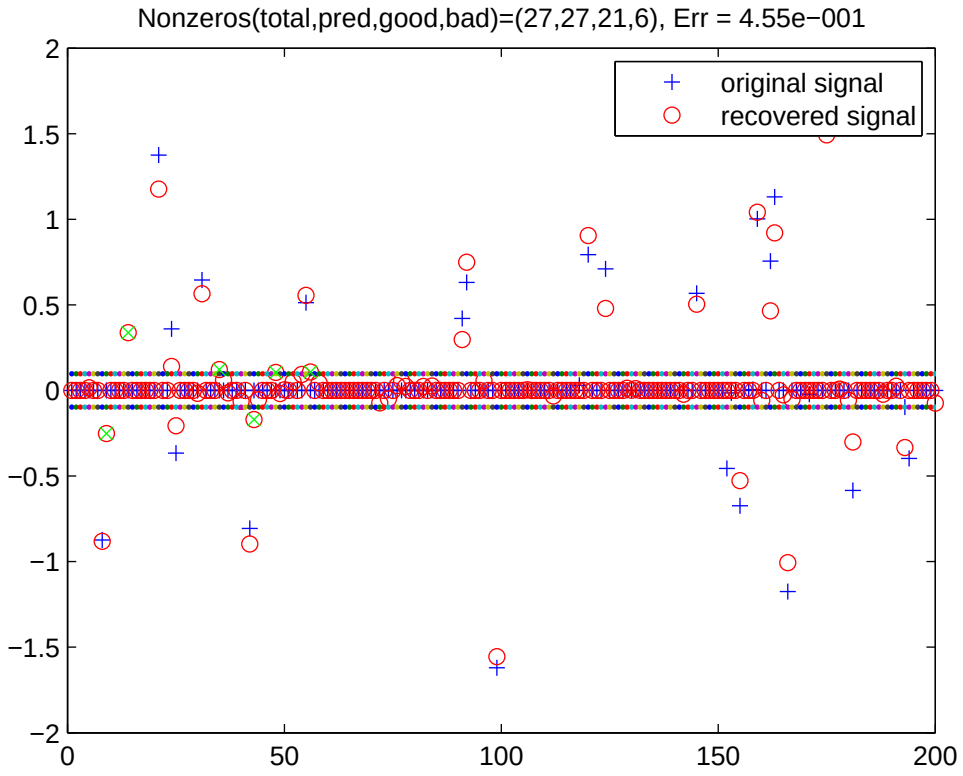

Truncated L1 Minimization
true signal
correct recovery
false recovery
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
50
100
150
200

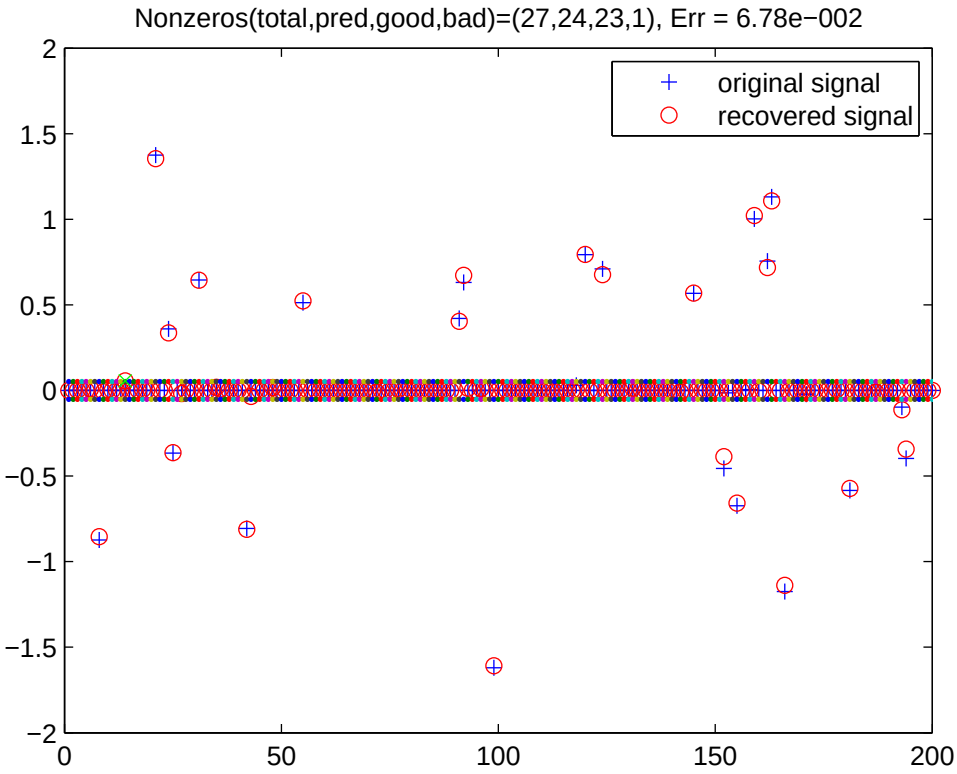

Nonzeros(total,pred,good,bad)=(27,22,18,4), Err = 5.31e−001
original signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
50
100
150
200

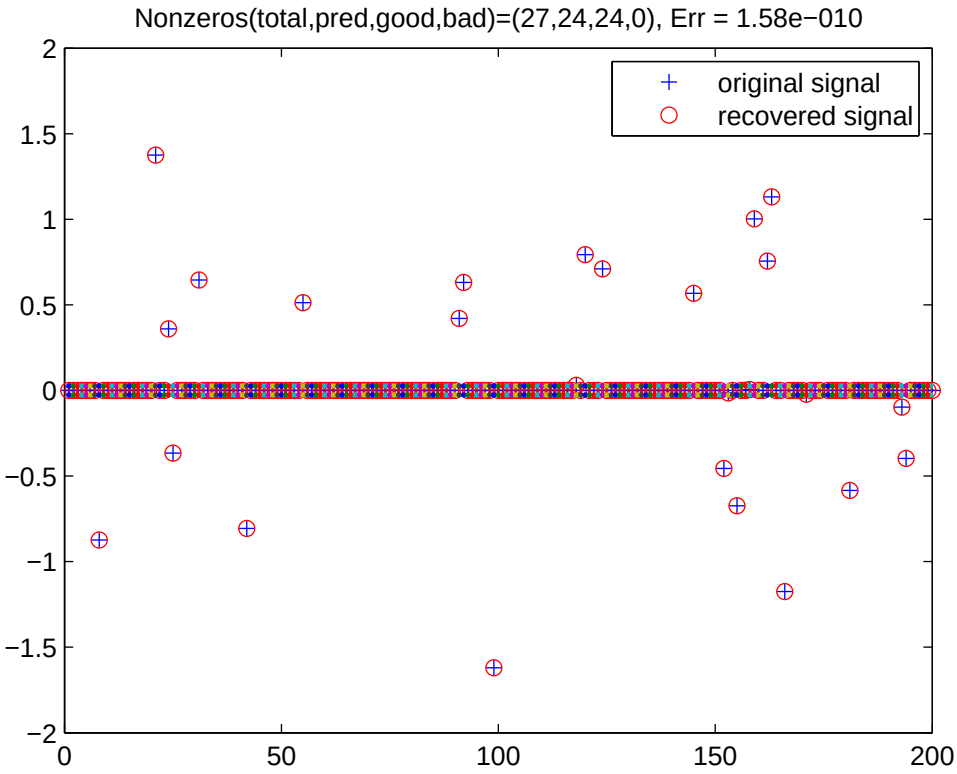

L1 Minimization
true signal
correct recovery
false recovery
1
0.5
0
−0.5
−1
−1.5
0
50
100
150
200

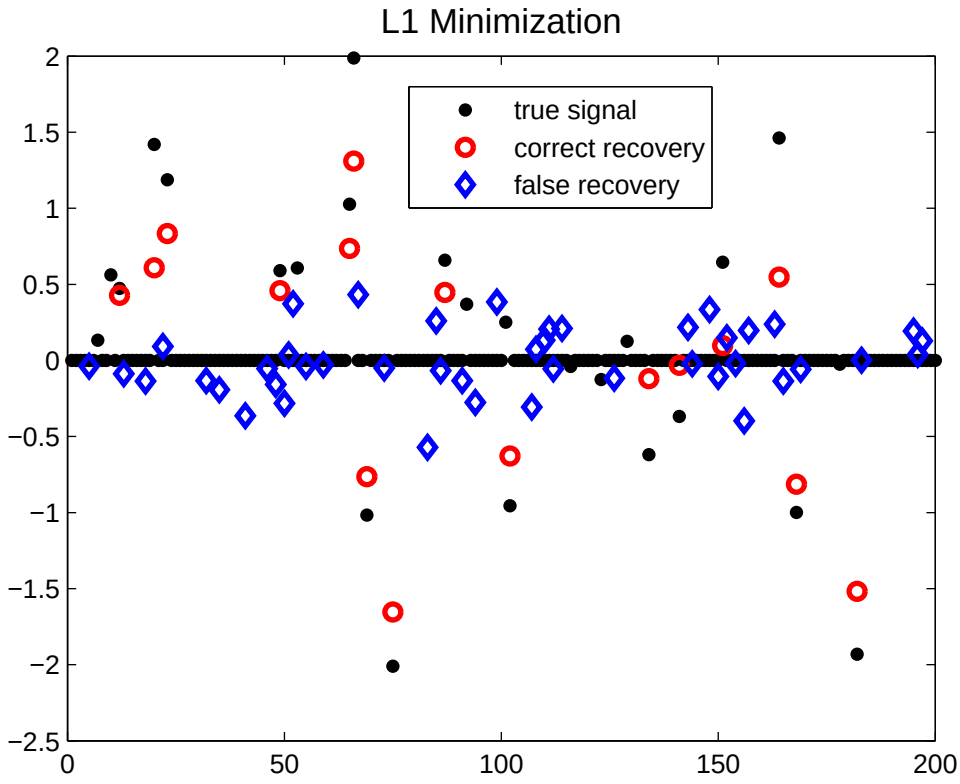

original signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
50
100
150
200

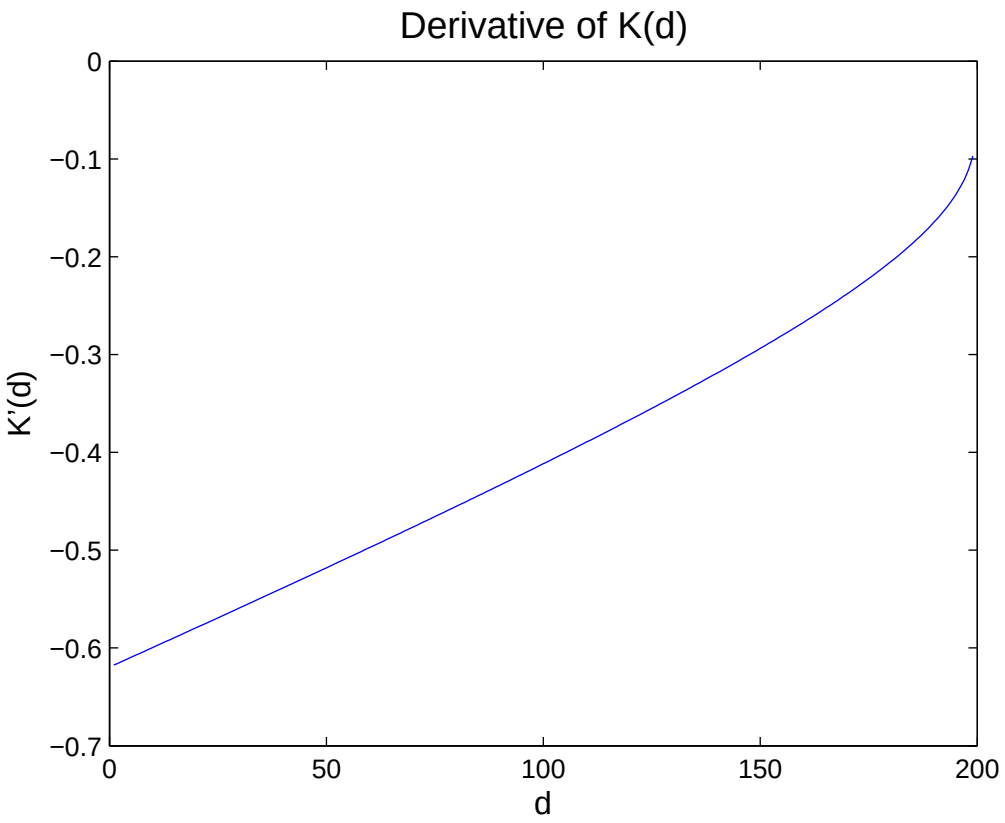

Truncated L1 Minimization
true signal
correct recovery
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
50
100
150
200

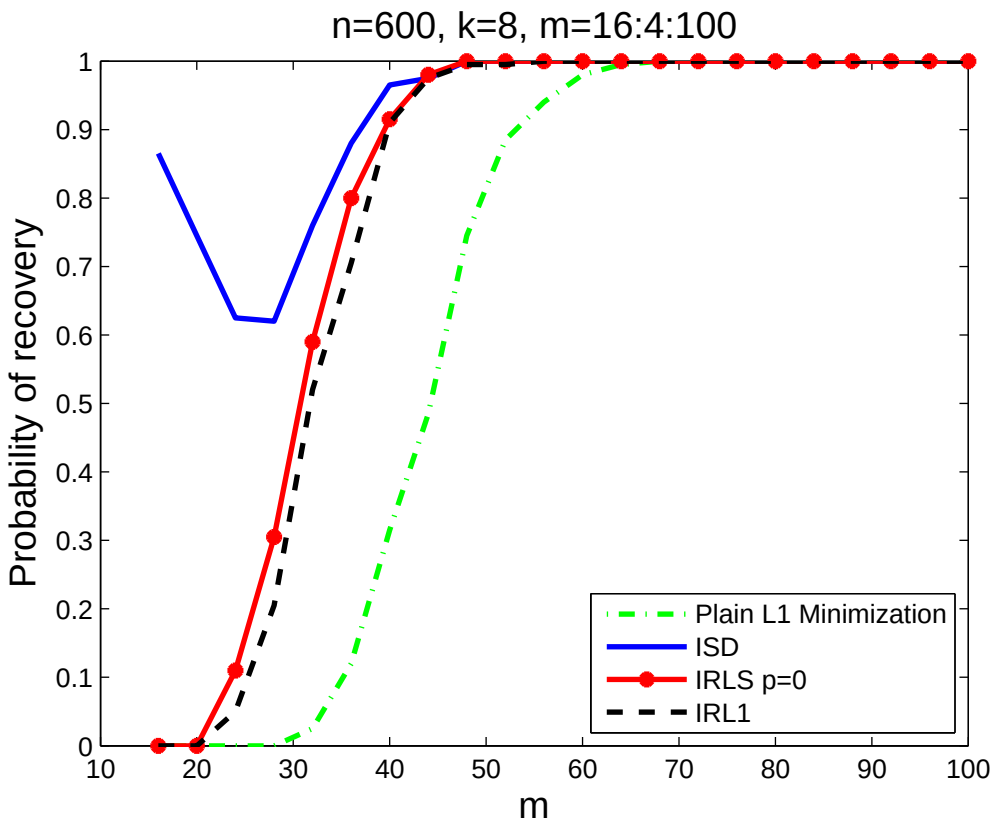

Nonzeros(total,pred,good,bad)=(27,27,21,6), Err = 4.55e−001
original signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
50
100
150
200

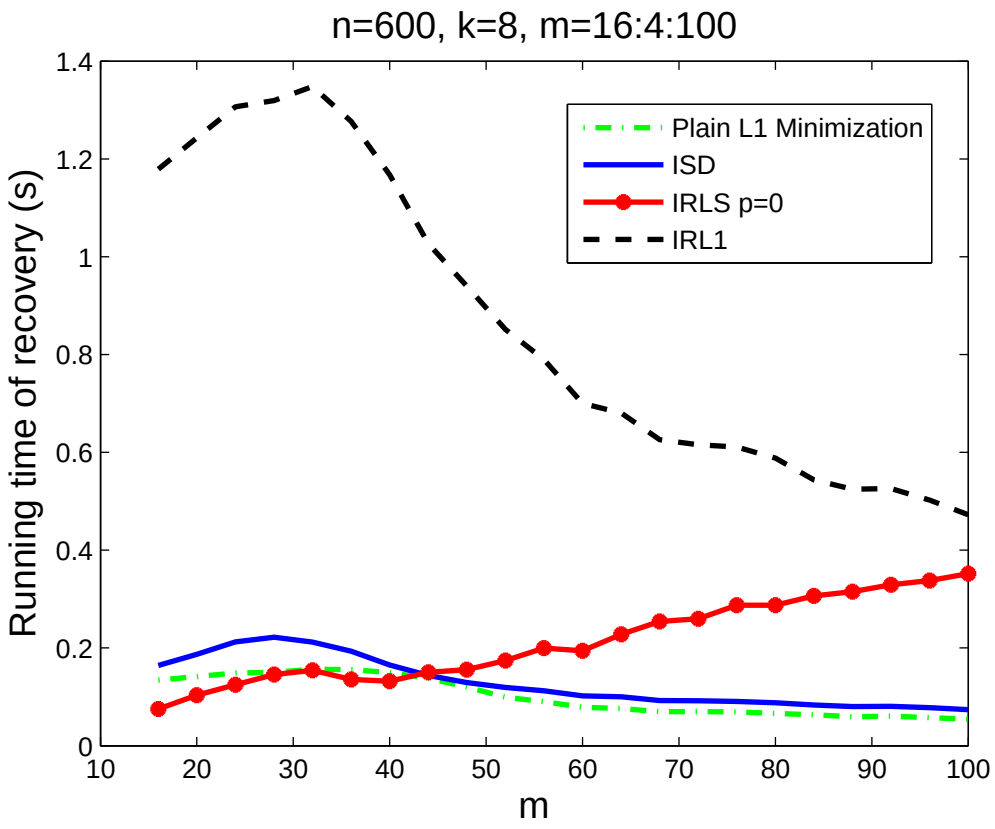

Nonzeros(total,pred,good,bad)=(27,24,23,1), Err = 6.78e−002
original signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
50
100
150
200

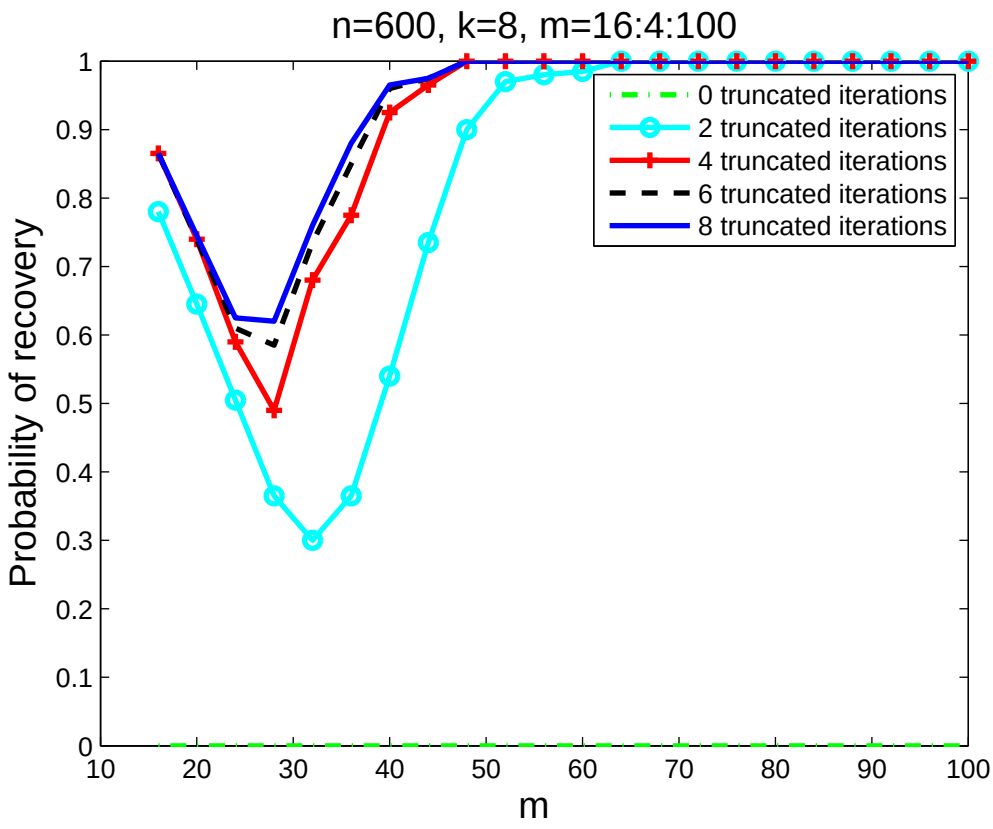

Nonzeros(total,pred,good,bad)=(27,24,24,0), Err = 1.58e−010
original signal
recovered signal
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
50
100
150
200

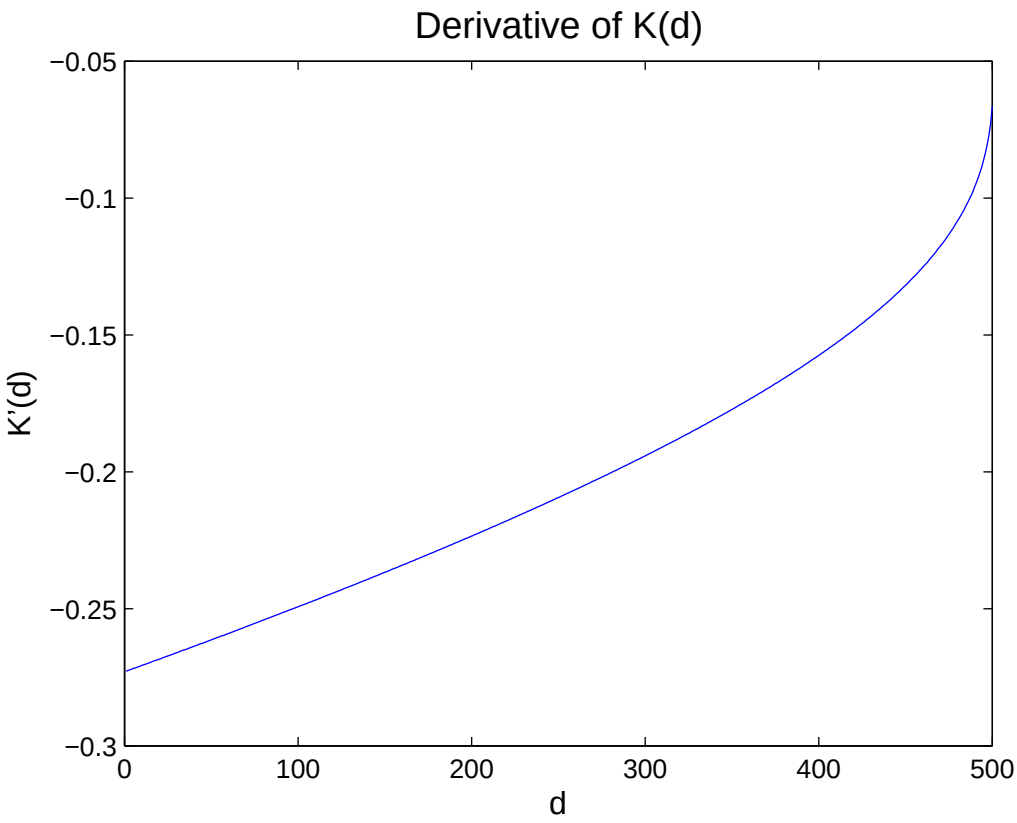

L1 Minimization
true signal
correct recovery
false recovery
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
50
100
150
200

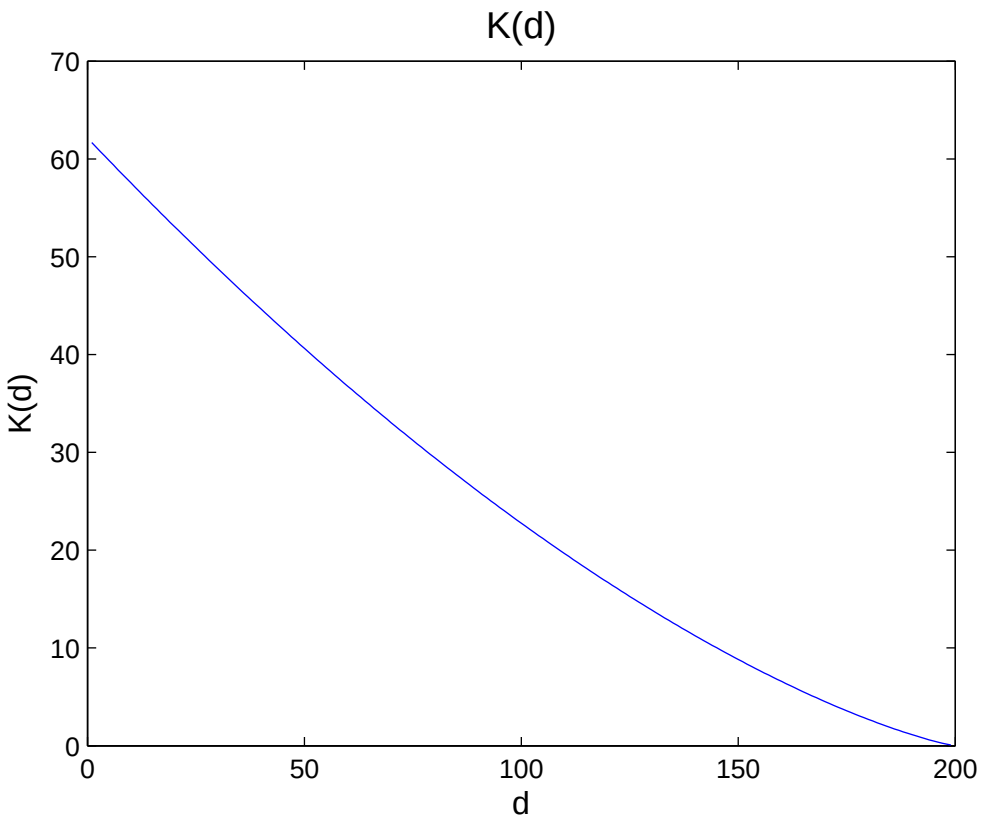

Derivative of K(d)
K'(d)
d
0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.6
−0.7
0
50
100
150
200


n=600, k=8, m=16:4:100
Probability of recovery
m
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
10
20
30
40
50
60
70
80
90
100
Plain L1 Minimization
ISD
IRLS p=0
IRL1


n=600, k=8, m=16:4:100
Running time of recovery (s)
m
Plain L1 Minimization
ISD
IRLS p=0
IRL1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
10
20
30
40
50
60
70
80
90
100

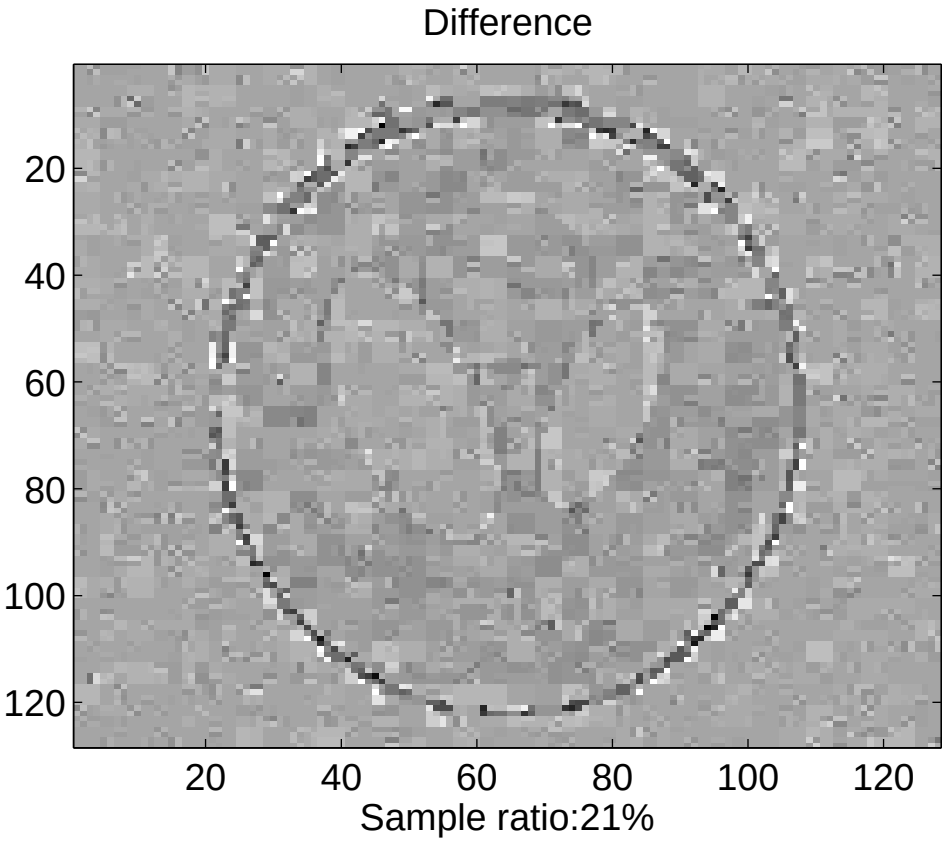

n=600, k=8, m=16:4:100
Probability of recovery
m
0 truncated iterations
2 truncated iterations
4 truncated iterations
6 truncated iterations
8 truncated iterations
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
10
20
30
40
50
60
70
80
90
100


Derivative of K(d)
−0.05
−0.1
−0.15
−0.2
−0.25
−0.3
K′(d)
0
100
200
300
400
500
d


K(d)
70
60
50
40
30
20
10
0
K(d)
0
50
100
150
200
d

BP: SNR=9.08dB, Err=3.05e−001, CPU time=30.60 s

Sample ratio:21%

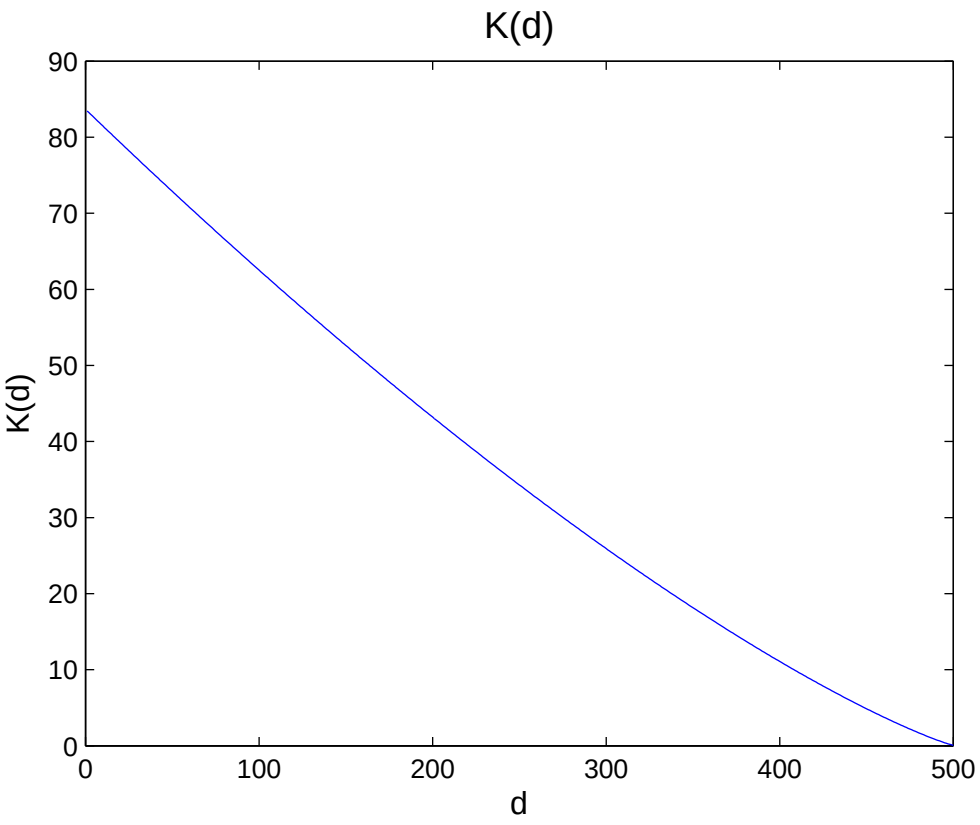

Difference
20
40
60
80
100
120
20
40
60
80
100
120
Sample ratio:21%

ISD: SNR=69.09dB, Err=3.05e−004, CPU time=30.43 s

Sample ratio:21%

Subtraction

Sample ratio:21%


K(d)
90
80
70
60
50
40
30
20
10
0
K(d)
0
100
200
300
400
500
d