Exact Sparse Recovery with L0 Projections

Ping Li
Deptartment of Statistical Science
Cornell University
Ithaca, NY 14853
pingli@cornell.edu

Cun-Hui Zhang
Department of Statistics and Biostatistics
Rutgers University
New Brunswick, NJ 08901
cunhui@stat.rutgers.edu

ABSTRACT

Many applications (e.g., anomaly detection) concern sparse signals. This paper focuses on the problem of recovering a K-sparse signal $\mathbf{x} \in \mathbb{R}^{1\times N}$, i.e., $K \ll N$ and $\sum_{i=1}^N 1\{x_i \neq 0\} = K$. In the mainstream framework of compressed sensing (CS), $\mathbf{x}$ is recovered from M linear measurements $\mathbf{y} = \mathbf{xS} \in \mathbb{R}^{1\times M}$, where $\mathbf{S} \in \mathbb{R}^{N\times M}$ is often a Gaussian (or Gaussian-like) design matrix.

In our proposed method, the design matrix $\mathbf{S}$ is generated from an α-stable distribution with $\alpha \approx 0$. Our decoding algorithm mainly requires one linear scan of the coordinates, followed by a few iterations on a small number of coordinates which are "undetermined" in the previous iteration. Our practical algorithm consists of two estimators. In the first iteration, the *(absolute) minimum estimator* is able to filter out a majority of the zero coordinates. The *gap estimator*, which is applied in each iteration, can accurately recover the magnitudes of the nonzero coordinates. Comparisons with linear programming (LP) and orthogonal matching pursuit (OMP) demonstrate that our algorithm can be significantly faster in decoding speed and more accurate in recovery quality, for the task of exact spare recovery. Our procedure is robust against measurement noise. Even when there are no sufficient measurements, our algorithm can still reliably recover a significant portion of the nonzero coordinates.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

General Terms

Algorithms, Performance, Theory

Keywords

Compressed Sensing, L0 Projections, Stable Distributions

1. INTRODUCTION

The goal of *Compressed Sensing (CS)* [7, 2] is to recover a sparse signal $\mathbf{x} \in \mathbb{R}^{1\times N}$ from a small number of non-adaptive linear measurements $\mathbf{y} = \mathbf{xS}$, (typically) by convex optimization (e.g., linear programming). Here, $\mathbf{y} \in \mathbb{R}^{1\times M}$ is the vector of measurements and $\mathbf{S} \in \mathbb{R}^{N\times M}$ is the design matrix. In classical settings, entries of $\mathbf{S}$ are i.i.d. samples from the Gaussian distribution $N(0,1)$, or a Gaussian-like distribution (e.g., a distribution with finite variance).

In this paper, we sample $\mathbf{S}$ from a heavy-tailed distribution. Strikingly, using such a design matrix turns out to result in a simple and powerful solution to the problem of exact K-sparse recovery, i.e., $\sum_{i=1}^N 1\{x_i \neq 0\} = K$.

1.1 Compressed Sensing

Sparse recovery can be naturally suitable for: (i) the "single pixel camera" type of applications; and (ii) the "data streams" type of applications. The idea of compressed sensing may be traced back to prior papers such as [10, 8, 5].

It has been realized (and implemented by hardware) that collecting a linear combination of a sparse vector, i.e., $\mathbf{y} = \mathbf{xS}$, can be more advantageous than sampling the vector itself. This is the foundation of the "single pixel camera" proposal. See the site `https://sites.google.com/site/igorcarron2/compressedsensinghardware` for a list of single-pixel-camera type of applications. Fig. 1 provides an illustrative example.

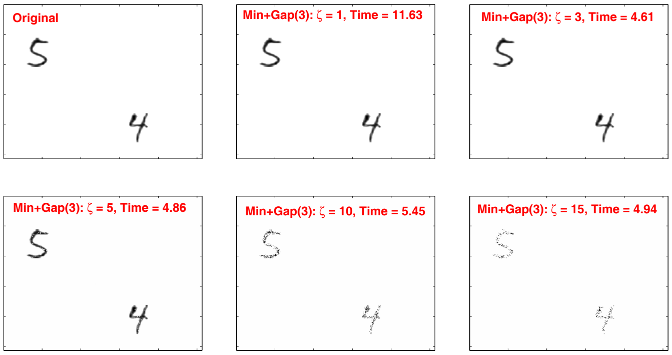
Original
Min+Gap(3): ζ = 1, Time = 11.63
Min+Gap(3): ζ = 3, Time = 4.61
Min+Gap(3): ζ = 5, Time = 4.86
Min+Gap(3): ζ = 10, Time = 5.45
Min+Gap(3): ζ = 15, Time = 4.94

Figure 1: We reconstruct a 256×256 image (i.e., $N = 65536$) with $K = 852$ nonzero pixels, using our proposed method with $M = K\log((N-K)/0.01)/\zeta)$ measurements, for $\zeta = 1, 3, 5, 10, 15$. Our method, "Min+Gap(3)", will be explained in the paper. For $\zeta = 1$, our method is able to exactly reconstruct the image, using 11.63 seconds. Strikingly, even with $\zeta = 15$ (i.e., $M = 891$ measurements), the reconstructed image could still be informative.

Natural images are in general not as sparse as the example in Fig. 1. We nevertheless expect that in many practical scenarios, the sparsity assumption can be reasonable. For

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

example, the differences between consecutive image/video frames taken by surveillance cameras are usually very sparse because the background remains still. In general, anomaly detection problems are often very sparse.

Another line of applications concerns data streams, which can be viewed as sparse dynamic vectors with entries rapidly varying over time. Due to the dynamic nature, it is nontrivial to know where the nonzero coordinates are, since the history of streaming is usually not stored. Many problems can be formulated as sparse data streams. For example, video data are naturally streaming. A common task in databases is to find the "heavy-hitters" [23], e.g., product items with the highest total sales. Also, see some recent papers on compressed sensing for network applications [21, 28, 29].

For data stream applications, entries of the signals $\mathbf{x}$ are (rapidly) updated over time (by addition and deletion). At a time t, the i_t-th entry is updated by I_t, i.e., $x_{i_t} \to x_{i_t} + I_t$. This is often referred to as the *turnstile model* [23]. As the projection operation is linear, i.e., $\mathbf{y} = \mathbf{xS}$, we can (re)generate corresponding entries of $\mathbf{S}$ on-demand whenever one entry of $\mathbf{x}$ is altered, to update all entries of the measurement vector $\mathbf{y}$. The use of stable random projections for estimating the α-th frequency moment $\sum_{i=1}^N |x_i|^\alpha$ (instead of the individual terms x_i) was studied in [14]. [16] proposed the use of geometric mean estimator for stable random projections, for estimating $\sum_{i=1}^N |x_i|^\alpha$ as well as the harmonic mean estimator for estimating $\sum_{i=1}^N |x_i|^\alpha$ when $\alpha \approx 0$. When the data are nonnegative, the method named *compressed counting* [17, 18] based on skewed-stable distributions becomes particularly effective.

1.2 Review of α-Stable Distribution

A random variable Z follows an α-stable distribution with unit scale, $S(\alpha, 1)$, if its characteristic function is [24]

$$E\left(e^{\sqrt{-1}Zt}\right) = e^{-|t|^\alpha}, \qquad 0 < \alpha \le 2 \tag{1}$$

When $\alpha = 2$ (or $\alpha = 1$), this is the normal (or Cauchy) distribution. To sample from $S(\alpha, 1)$, we sample independent exponential $w \sim exp(1)$ and uniform $u \sim unif(-\pi/2, \pi/2)$ variables, and then compute $Z \sim S(\alpha, 1)$ by [3]

$$Z = \frac{\sin(\alpha u)}{(\cos u)^{1/\alpha}}\left[\frac{\cos(u - \alpha u)}{w}\right]^{(1-\alpha)/\alpha} \sim S(\alpha, 1) \tag{2}$$

If $S_1, S_2 \sim S(\alpha, 1)$ i.i.d., then for any constants C_1, C_2, we have $C_1S_1 + C_2S_2 = S \times (|C_1|^\alpha + |C_2|^\alpha)^{1/\alpha}$, where $S \sim S(\alpha, 1)$. More generally, $\sum_{i=1}^N x_iS_i = S \times (\sum_{i=1}^N |x_i|^\alpha)^{1/\alpha}$.

In our numerical experiments with Matlab, α is taken to be 0.03 and no special data storage structure is needed. Our method can be intuitively illustrated by an "idealized" algorithm using the limit as $\alpha \to 0$.

1.3 The Proposed Practical Recovery Algorithm

We assume $\mathbf{x} \in \mathbb{R}^{1\times N}$ is K-sparse. We obtain M linear measurements $\mathbf{y} = \mathbf{xS} \in \mathbb{R}^{1\times M}$, where entries of $\mathbf{S} \in \mathbb{R}^{N\times M}$, denoted by s_{ij}, are i.i.d. samples from $S(\alpha, 1)$ with a small α (e.g., 0.03). That is, each measurement is $y_j = \sum_{i=1}^N x_is_{ij}$. Our algorithm, which consists of two estimators, utilizes the ratio statistics $z_{i,j} = y_j/s_{ij}$, $j = 1, 2, ..., M$, to recover x_i.

The **absolute minimum estimator** is defined as

$$\hat{x}_{i,min} = z_{i,t}, \quad \text{where } t = \underset{1\le j\le M}{\text{argmin}}\, |z_{i,j}|, \qquad z_{i,j} = \frac{y_j}{s_{ij}}$$

Algorithm 1 The proposed recovery algorithm.

Input: K-sparse signal $\mathbf{x} \in \mathbb{R}^{1\times N}$, threshold $\epsilon > 0$ (e.g., 10^{-5}), design matrix $\mathbf{S} \in \mathbb{R}^{N\times M}$ sampled from $S(\alpha, 1)$ with $\alpha \approx 0$ (e.g., 0.03). $\mathbf{S}$ can be generated on-demand in data streams.

Output: The recovered signal, denoted by $\hat{x}_i$, $i = 1$ to N.

Linear measurements: $\mathbf{y} = \mathbf{xS}$, which can be conducted incrementally if entries of $\mathbf{x}$ arrive in a streaming fashion.

Detection: For $i = 1$ to N, compute $\hat{x}_{i,min} = \text{argmin}_j |y_j/s_{ij}|$. If $|\hat{x}_{i,min}| \le \epsilon$, set $\hat{x}_i = 0$.

Estimation: If $|\hat{x}_{i,min}| > \epsilon$, compute the gaps for the sorted observations y_j/s_{ij} and estimate x_i using the gap estimator $\hat{x}_{i,gap}$. Let $\hat{x}_i = \hat{x}_{i,gap}$. See details below.

Iterations: If $|\hat{x}_{i,min}| > \epsilon$ and the minimum gap length $> \epsilon$, we call this i an "undetermined" coordinate and set $\hat{x}_i = 0$. Compute the residuals: $\mathbf{r} = \mathbf{y} - \hat{\mathbf{x}}\mathbf{S}$, and apply the gap estimator using the residual $\mathbf{r}$, only on these "undetermined" coordinates. Repeat the iterations a number of times until no changes are observed.

We prove that essentially $M_0 = K\log((N-K)/\delta)$ measurements are sufficient for detecting all zeros with probability at least $1-\delta$. The actual required measurements will be significantly lower than M_0 if we use the minimum algorithm together with the gap estimator and the iterative process.

When $|\hat{x}_{i,min}| > \epsilon$, the **gap estimator** is used to estimate the magnitude of x_i. We first sort $z_{i,j}$'s: $z_{i,(1)} \le z_{i,(2)} \le ... \le z_{i,(M)}$, and then compute the gaps: $z_{i,(j+1)} - z_{i,(j)}$, $1 \le j \le M-1$. The gap estimator is simply

$$\hat{x}_{i,gap} = \frac{1}{2}\left\{z_{i,(j_i)} + z_{i,(j_i+1)}\right\}, j_i = \underset{1\le j\le M-1}{\text{argmin}}\; z_{i,(j+1)} - z_{i,(j)}$$

We also derive the error bound $\mathbf{Pr}(|\hat{x}_{i,gap} - x_i| > \epsilon)$. When $M < M_0$, we discover that it is better to apply the gap estimator iteratively, each time using the residual measurements only on the "undetermined" coordinates; see Alg. 1.

2. INTUITION

Our procedure is intuitive from the ratio of two independent α-stable random variables, in the limit $\alpha \to 0$. Recall that, for each coordinate i, our observations are (y_j, s_{ij}), $j = 1$ to M. Naturally our first attempt was to use the joint likelihood of (y_j, s_{ij}). However, our proposed method only utilizes the ratio statistics y_j/s_{ij}. We first explain why.

2.1 Why Using the Ratio Statistics y_j/s_{ij}?

For convenience, we first define

$$\theta = \left(\sum_{i=1}^N |x_i|^\alpha\right)^{1/\alpha}, \qquad \theta_i = (\theta^\alpha - |x_i|^\alpha)^{1/\alpha} \tag{3}$$

Denote the density function of $S(\alpha, 1)$ by $f_S(s)$. By a conditional probability argument, the joint density of (y_j, s_{ij}) can be shown to be $\frac{1}{\theta_i}f_S(s_{ij})f_S\left(\frac{y_j - x_is_{ij}}{\theta_i}\right)$, from which we derive the joint log-likelihood of (y_j, s_{ij}), $j = 1$ to M, as

$$l(x_i, \theta_i) = \sum_{j=1}^M \log f_S(s_{ij}) + \sum_{j=1}^M \log f_S\left(\frac{y_j - x_is_{ij}}{\theta_i}\right) - M\log\theta_i$$

Closed-form expressions of f_S are in general not available (unless $\alpha = 1, 2$). Interestingly, from the procedure (2) for sampling $Z \sim S(\alpha, 1)$, we can guess that $1/|Z|^\alpha$ is approximately $w \sim exp(1)$ when $\alpha \approx 0$. Indeed, as shown by [6],

$1/|Z|^\alpha \rightarrow exp(1)$ in distribution. Using this limit, the density function $f_S(s)$ is approximately $\frac{\alpha}{2}\frac{e^{-|s|^{-\alpha}}}{|s|^{\alpha+1}}$, and hence the joint log-likelihood $l(x_i, \theta_i)$ is approximately

$$l\left(x_i,\theta_i\right) \approx \sum_{j=1}^{M} \log f_S(s_{ij}) + \alpha M \log \theta_i + M \log\left(\frac{\alpha}{2}\right)$$
$$+ \sum_{j=1}^{M}\left\{-\frac{\theta_i^\alpha}{|y_j - x_i s_{ij}|^\alpha} - (\alpha+1)\log|y_j - x_i s_{ij}|\right\}$$

which approaches infinity (i.e., the maximum likelihood) at the poles: $y_j - x_i s_{ij} = 0$, $j = 1$ to M. This is the reason why we use only the ratio statistics $z_{i,j} = y_j/s_{ij}$.

2.2 The Approximate Distribution of y_j/s_{ij}

Note that $\frac{y_j}{s_{ij}} = \frac{\sum_{t=1}^{N} x_t s_{tj}}{s_{ij}} = \frac{\sum_{t\neq i} x_t s_{tj}}{s_{ij}} + x_i = \theta_i \frac{S_2}{S_1} + x_i$, where $S_1, S_2 \sim S(\alpha, 1)$, i.i.d. Recall the definition $\theta_i = \left(\sum_{t\neq i}|x_t|^\alpha\right)^{1/\alpha}$. Thus, $\mathbf{Pr}\left(\frac{y_j}{s_{ij}} < t\right) = \mathbf{Pr}\left(\frac{S_2}{S_1} < \frac{t-x_i}{\theta_i}\right)$ and the problem boils down to finding the distribution of the ratio of two α-random variables with $\alpha \approx 0$. Using the limits: $1/|S_1|^\alpha \rightarrow exp(1)$, $1/|S_2|^\alpha \rightarrow exp(1)$, the approximate cumulative distribution function (CDF) of y_j/s_{ij} is

$$\mathbf{Pr}\left(\frac{y_j}{s_{ij}} < t\right) = \mathbf{Pr}\left(\frac{S_2}{S_1} < \frac{t-x_i}{\theta_i}\right)$$
$$\approx \begin{cases} \frac{1}{2\left(1+\left|\frac{t-x_i}{\theta_i}\right|^\alpha\right)} & t < x_i \\ 1 - \frac{1}{2\left(1+\left|\frac{t-x_i}{\theta_i}\right|^\alpha\right)} & t \geq x_i \end{cases} \tag{4}$$

The CDF of S_2/S_1 is also given by (4) with $x_i = 0$, $\theta_i = 1$.

Fig. 2 plots the approximate CDFs (4) for S_2/S_1 (left panel) and y_j/s_{ij} (right panel, with $x_i = 0$ and three values of θ^α). While the distribution of S_2/S_1 is extremely heavy-tailed, about half of the probability mass concentrated near 0. This means, as $\alpha \rightarrow 0$, samples of $|S_2/S_1|$ are equal likely to be either very close to zero or very large. Since (4) is only approximate, we also provide the simulations of S_2/S_1 in Fig. 3 to help verify the approximate CDF in Fig. 2.

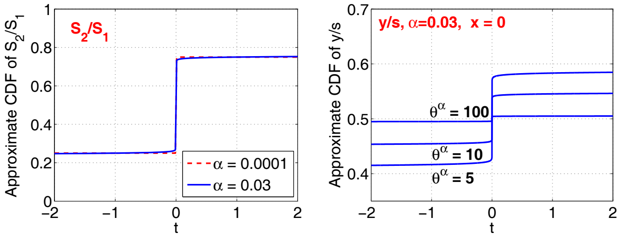

Figure 2: Approximate CDFs of S_2/S_1 (left panel) and $\frac{y_j}{s_{ij}}$ (right panel) as in (4). The CDF of S_2/S_1 is heavy-tailed with an essentially vertical jump around 0, i.e., samples of S_2/S_1 are likely to be either close 0 or large. The CDF of y_j/s_{ij} is a scaled (and shifted) version of the CDF of S_2/S_1, with an almost vertical jump around x_i. This motivates us to develop the gap estimator. Given M observations: y_j/s_{ij}, $j = 1$ to M, observations outside $(x_i - e, x_i + e)$ for very small e will likely be far away from each other. Observations within $(x_i - e, x_i + e)$ are very close to each other (i.e., a cluster around x_i).

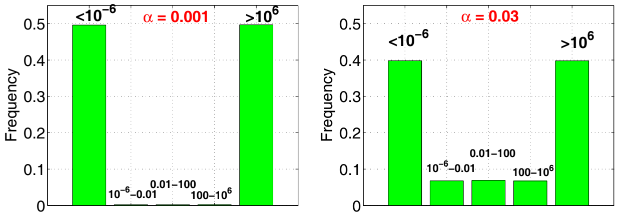

Figure 3: Simulations of $|S_2/S_1|$ directly using the formula (2) for generating two independent α-stable variables S_1 and S_2. With $\alpha \rightarrow 0$, most of the samples are either close to 0 ($< 10^{-6}$) or large ($> 10^6$).

2.3 The "Idealized" Algorithm with $K = 2$

We consider $K = 2$, to illustrate the iterative process in Alg. 1. For simplicity, let $x_1 = x_2 = 1$, $x_i = 0, \forall\ 3 \leq i \leq N$. This way, the observations become $y_j = x_1 s_{1j} + x_2 s_{2j} = s_{1j} + s_{2j}$, for $j = 1$ to M. The ratio statistics are

$$z_{1,j} = y_j/s_{1j} = 1 + \frac{s_{2j}}{s_{1j}}, \qquad z_{2,j} = y_j/s_{2j} = 1 + \frac{s_{1j}}{s_{2j}}$$
$$z_{i,j} = y_j/s_{ij} = \frac{s_{1j}}{s_{ij}} + \frac{s_{2j}}{s_{ij}}, \quad i \geq 3$$

We assume an "idealized" algorithm, which allows us to use an extremely small α. As $\alpha \rightarrow 0$, $\frac{s_{2j}}{s_{1j}}$ is either (virtually) 0 or $\pm\infty$. Note that $\frac{s_{1j}}{s_{2j}} \approx 0 \Longleftrightarrow \frac{s_{2j}}{s_{1j}} \approx \pm\infty$. Suppose, with $M = 3$ observations, the ratio statistics, for $i = 1, 2$, are: $(z_{1,1}, z_{2,1}) = (1, \pm\infty)$, $(z_{1,2}, z_{2,2}) = (\pm\infty,\ 1)$, $(z_{1,3}, z_{2,3}) = (1, \pm\infty)$. Then we have seen $z_{1,j} = 1$ twice and this "idealized" algorithm is able to correctly estimate $\hat{x}_1 = 1$, as there is a "cluster" of 1's. After we have estimated x_1, we compute the residual $r_j = y_j - \hat{x}_1 s_{1j} = s_{2j}$. In the **second iteration**, the ratio statistics become

$$r_j/s_{2j} = 1, \qquad r_j/s_{ij} = \frac{s_{2j}}{s_{ij}}, \quad i \geq 3$$

This means we know $x_2 = 1$. We again compute the residuals, which become zero. In the **third iteration**, all zero coordinates can be recovered. The most exciting part of this example is that, with $M = 3$ measurements, we can recovery a signal with $K = 2$, regardless of N. We hope this example helps understand why our algorithm performs so well empirically. We summarize the "idealized" algorithm:

1. The algorithm assumes $\alpha \rightarrow 0$, or as small as necessary.
2. As long as there are two observations y_j/s_{ij} in the extremely narrow interval $(x_i - e, x_i + e)$ with e very close to 0, the algorithm is able to correctly recover x_i. We assume e is so small that it is outside the required precision range of x_i. Here we use e instead of ϵ to differentiate it from the ϵ in our Alg. 1.

This "idealized" algorithm can not be strictly implemented. When we use a small α instead of $\alpha = 0$, the observations $|y_j/s_{ij}|$ will be between 0 and ∞, and we will not be able to identify the true x_i with high confidence unless we see **two** essentially identical observations. As analyzed in Sec. 5, the proposed gap estimator is a practical surrogate.

2.4 The Intuition for the Two Estimators

Fig. 2 (right panel) shows that the distribution of y_j/s_{ij} is heavy-tailed, with a jump very near x_i in the CDF. This means more than one observations (among M observations) will likely lie in the extremely narrow interval around x_i, depending on the value of θ^α (which is essentially K). We are able to detect whether $x_i = 0$ if there is just one observation near x_i. To estimate the magnitude of x_i, however, we need to see a "cluster" of observations, e.g., two or more observations which are essentially identical. This is the intuition for the minimum estimator and the gap estimator. Also, as one would expect, Fig. 2 shows that the performance will degrade (i.e., more observations are needed) as θ_i^α increases.

The gap estimator is a practical surrogate for the "idealized" algorithm. Basically, for each i, if we sort the observations: $z_{i,(1)} \leq z_{i,(2)} \leq ... \leq z_{i,(M)}$, the two neighboring observations corresponding to the minimum gap will be likely lying in a narrow neighborhood of x_i, provided that the length of the minimum gap is very small, due to the heavy concentration of the probability mass about x_i.

If the observed minimum gap is not small, we give up estimating this ("undetermined") coordinate in the current iteration. After we remove the (reliably) estimated coordinates, we may have a better chance of successfully recovering some of these undetermined coordinates because the effective "K" and the effective "N" are significantly reduced.

3. TWO BASELINES: LP AND OMP

Both linear programming (LP) and orthogonal matching pursuit (OMP) utilize a design matrix sampled from Gaussian (i.e., α-stable with $\alpha = 2$) or Gaussian-like distribution. Here, we use $\mathbf{S}_{(2)}$ to denote such a design matrix.

The well-known LP algorithm recovers the signal $\mathbf{x}$ by solving the following l_1 optimization problem:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{y} = \mathbf{x}\mathbf{S}_{(\mathbf{2})} \tag{5}$$

which is also commonly known as *Basis Pursuit* [4]. It has been proved that LP can recover $\mathbf{x}$ using $M = O\left(K \log(N/K)\right)$ measurements [11]. This procedure is computationally prohibitive for large N (e.g., $N = 10^9$). When there are measurement noises, the LP algorithm can be modified as other convex optimization problems, for example, the *Lasso* [25].

The orthogonal matching pursuit (OMP) algorithm [22] is a greedy iterative procedure. It typically proceeds with K iterations. At each iteration, it conducts univariate least squares for all the coordinates on the residuals, and chooses the coordinate which maximally reduces the overall square errors. At the end of each iteration, all chosen coordinates are used to update the residuals via a multivariate least square. [30, 12] showed that, under appropriate conditions, the required number of measurements of OMP is essentially $O(K\log(N-K))$, which improved the prior result in [26]. Our experimental study will focus on the comparisons with OMP and LP, as they are the basic and strong baselines. We are aware of other methods such as the "message-passing" algorithm [9] and the "sparse matrix" algorithm [13].

In parallel to this paper, we develop two other algorithms concurrently: (i) *sparse recovery with compressed counting* [19], by using skewed projections [17, 18], and (ii) *sparse recovery with very sparse matrices* [20], by using an idea similar to *very sparse stable random projections* [15] in KDD'07.

4. SIMULATIONS

To validate the procedure in Alg. 1, we provide some simulations (and comparisons with LP and OMP), before presenting the theory. In each simulation, we randomly select K coordinates from a total of N coordinates. We set the magnitudes (and signs) of these K coordinates according to one of the two mechanisms. (i) **Gaussian signals**: the values are sampled from $Normal(0, 5^2)$. (ii) **Sign signals**: we simply take the signs, i.e., $\{-1, 0, 1\}$, of the generated Gaussian signals. The number of measurements M is chosen by

$$M = M_0/\zeta, \qquad M_0 = K\log\left((N-K)/\delta\right) \tag{6}$$

where $\delta = 0.01$ and $\zeta \in \{1, 1.3, 2, 3, 4, 5\}$.

4.1 Sample Instances of Simulations

Fig. 4 to Fig. 6 present instances of simulations, for sign signals, $N = 100000$ and $K = 30$. In each simulation (each figure), we generate the heavy-tailed design matrix $\mathbf{S}$ (with $\alpha = 0.03$) and the Gaussian design matrix $\mathbf{S}_{(2)}$ (with $\alpha = 2$), using the same random variables (w's and u's) as in (2). This provides shoulder-by-shoulder comparisons of our method with LP and OMP. We use the popular l_1-magic package [1].

In Fig. 4, we let $M = M_0$ (i.e., $\zeta = 1$). For this M, all methods perform well. The **left-top** panel of Fig. 4 shows that the minimum estimator $\hat{x}_{i,min}$ can precisely identify all the nonzero coordinates. The **right-top** panel shows that the gap estimator $\hat{x}_{i,gap}$ applied on the coordinates identified by $\hat{x}_{i,min}$, can accurately estimate the magnitudes. The label "**min+gap(1)**" means only one iteration is performed (which is good enough for $\zeta = 1$).

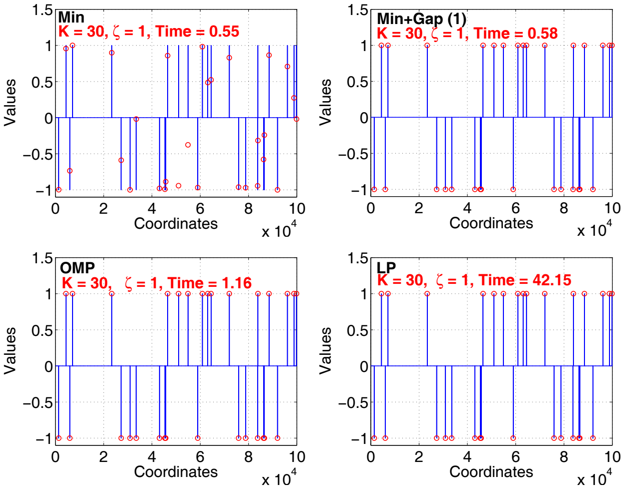

Figure 4: Reconstruction results from one simulation, using $N = 100000$, $K = 30$, $M = M_0$ (i.e., $\zeta = 1$), and sign signals. The reconstructed signals are denoted by (red) circles. The minimum estimator (left-top) is able to identify all nonzero coordinates with no false positives, using only 0.55 seconds. With one iteration of the gap estimator (right-top), we can perfectly reconstruct the signal using additional 0.03 seconds (so the total time is 0.58 seconds). Both OMP and LP (bottom panels) also perform well at significantly higher computational costs.

The **bottom** panels of Fig. 4 show that both OMP and LP also perform well when $\zeta = 1$. OMP is noticeably more costly than our method (even though K is small) while LP is significantly much more expensive than all other methods.

We believe these plots of sample instances provide useful information, especially when $M \ll M_0$. If M is too small, then all methods will ultimately fail, but the failure patterns are important, for example, a "catastrophic" failure such that none of the reported nonzeros is correct will be very undesirable. Our method does not have such failures.

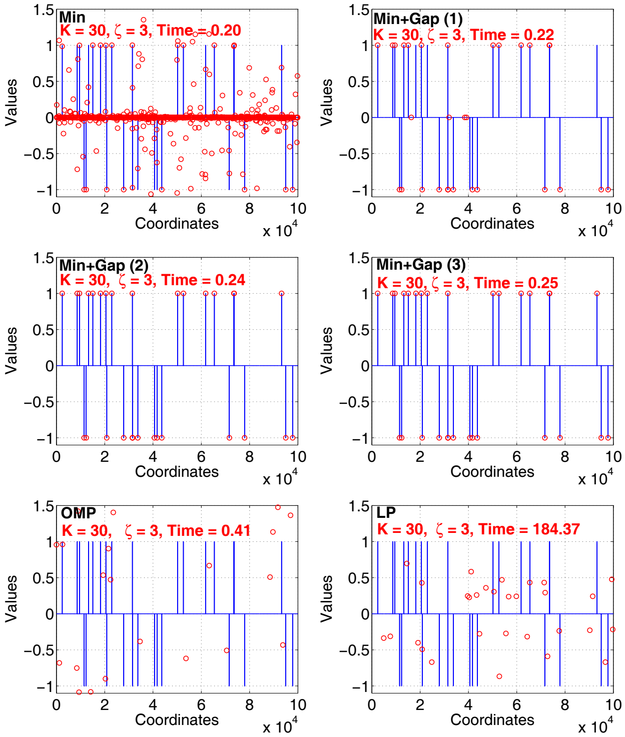

Figure 5: Reconstruction results from one simulation, with $N = 100000$, $K = 30$, $M = M_0/3$ (i.e., $\zeta = 3$), and sign signals. Many of the false positives produced by the min estimator are removed by the gap estimator after 1 iteration. The signal is perfectly reconstructed after the second iteration. Both OMP and LP perform poorly.

Simulations in Fig. 5 use $M = M_0/3$ (i.e., $\zeta = 3$). The minimum estimator $\hat{x}_{i,min}$ outputs a significant number of false positives but our method can still perfectly reconstruct signal using the gap estimator with one additional iteration (i.e., **Min+Gap(2)**). In comparisons, both LP and OMP perform poorly and exhibit catastrophic failures.

Fig. 6 uses $M = M_0/5$ (i.e., $\zeta = 5$) to further demonstrate the robustness of our algorithm. As M is not large enough, a small fraction of nonzero coordinates are not recovered by our method, but there are no catastrophic failures. This point is of course already illustrated in Fig. 1 (with $M \approx K$).

4.2 Summary Statistics from Simulations

We also report the aggregated reconstruction errors and run times, using $M = M_0/\zeta$ with $\zeta \in \{1, 1.3, 2, 3, 4, 5\}$, and $(N,\ K)$ from $\{(10000, 50), (10000, 100), (100000, 100)\}$, for both Gaussian $Normal(0, 5^2)$ signals and sign signals. For each setting, we repeat the simulations 1000 times, except $(N, K) = (100000, 100)$, for which we only repeat 100 times.

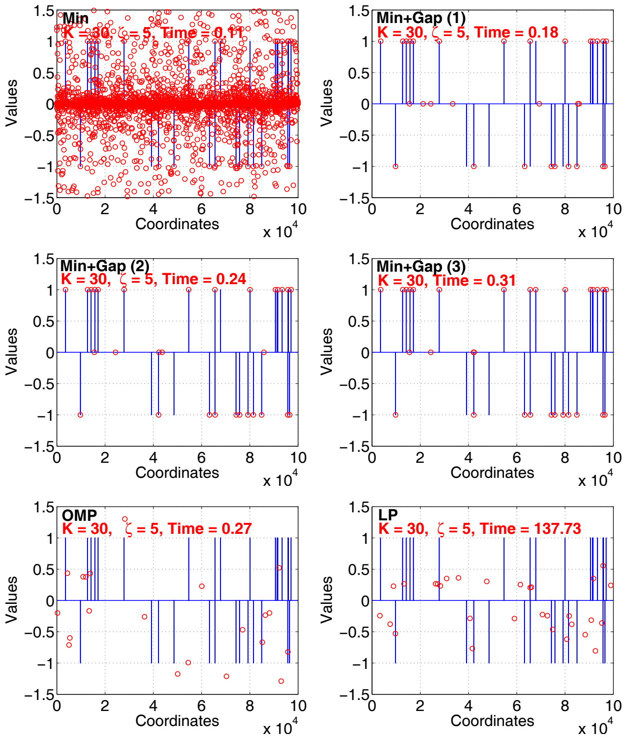

Figure 6: Reconstruction results from one simulation, with $N = 100000$, $K = 30$, $M = M_0/5$ (i.e., $\zeta = 5$), and sign signals. A small fraction of the nonzero coordinates are not reconstructed by our method. In comparisons, both OMP and LP perform very poorly in that none of the reported nonzero coordinates is correct.

4.2.1 Precision and Recall

For sparse recovery, it is crucial to correctly recover the nonzero locations. Here we use precision and recall

$$\text{Precision} = \frac{\#\text{ True Nonzeros}}{\#\text{ Returned Nonzeros}} = \frac{tp}{tp + fp},$$
$$\text{Recall} = \frac{\#\text{ True Nonzeros}}{\#\text{ Total True Nonzeros}} = \frac{tp}{tp + fn}$$

to compare the proposed absolute minimum estimator with LP decoding. Here, we view nonzero coordinates as "positives" (p) and zero coordinates as "negatives" (n). Ideally, we hope to maximize "true positives" (tp) and minimize "false positives" (fp) and "false negatives" (fn). In reality, we usually hope to achieve at least perfect recalls so that the retrieved set of coordinates contain all the true nonzeros.

Fig. 7 presents the (median) precision-recall curves. Our minimum estimator always produces essentially 100% recalls, meaning that the true positives are always included for the next stage of reconstruction. In comparison, as M decreases, the recalls of LP decreases significantly.

4.2.2 Reconstruction Accuracy

The reconstruction accuracy is another useful measure of quality. We define the reconstruction error as

$$\text{Error} = \sqrt{\frac{\sum_{i=1}^{N}\left(x_i - \text{estimated } x_i\right)^2}{\sum_{i=1}^{N} x_i^2}} \qquad (7)$$

Fig. 8 presents the median reconstruction errors. At $M = M_0$ (i.e., $\zeta = 1$), all methods perform well. For sign signals, both OMP and LP perform poorly as soon as $\zeta > 1.3$ or 2

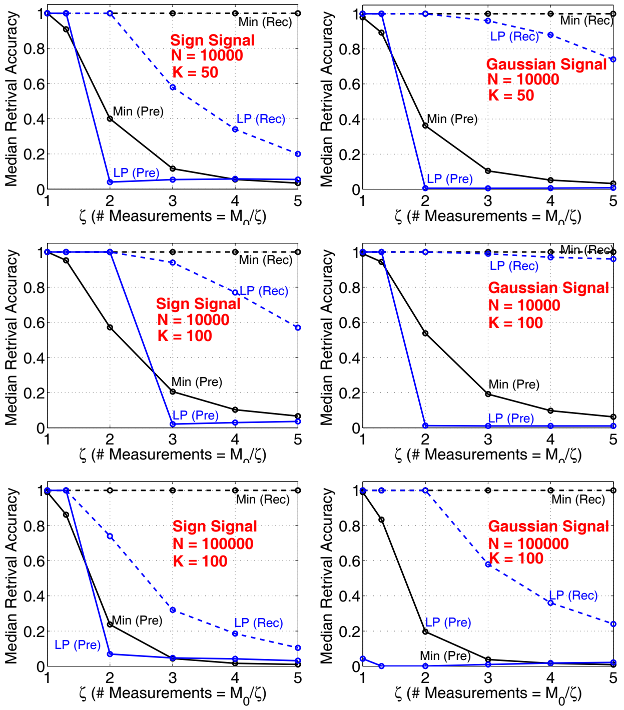

Figure 7: Median precision and recall curves, for comparing our proposed minimum estimator with LP decoding. The minimum estimator produces essentially 100% recalls even for M as small as $M_0/5$.

and OMP results are particularly bad. For Gaussian signals, OMP can produce good results even when $\zeta = 3$.

Our method performs well, and 2 or 3 iterations of the gap estimation procedure help noticeably. One should keep in mind that errors defined by (7) may not always be as informative. For example, with $M = M_0/5$, Fig. 6 shows that, even though our method fails to recover a small fraction of nonzero coordinates, the recovered coordinates are accurate. In comparison, for OMP and LP, essentially none of the nonzero coordinates in Fig. 6 could be identified when $M = M_0/5$. We have seen the stability and reliability of our method in Fig. 1. In that example, even with $M \approx K$, the reconstructed signal by our method is still informative.

4.2.3 Reconstruction Time

Fig. 9 confirms that LP is computationally expensive, using the l_1-magic package. In comparison, OMP is substantially more efficient than LP, although it is still much more costly than our algorithm, especially when K is not small.

In addition to the results reported in Fig. 9, we also experimented with the SPGL1 package [27] (the faster .mex version) and found our method (implemented in Matlab) is still substantially much faster than SPGL1.

5. THEORY

This section will develop the theoretical analysis of our method, including the minimum estimator and the gap estimator. The minimum estimator is not crucial once we have the gap estimator and the iterative process. We keep it in our procedure for two reasons. Firstly, it is faster than the gap estimator and is able to identify a majority of the zero coordinates in the first iteration. Secondly, even if we just use one iteration, the required sample size for the minimum estimator M is essentially $K \log N/\delta$, which already matches the complexity bounds in the compressed sensing literature.

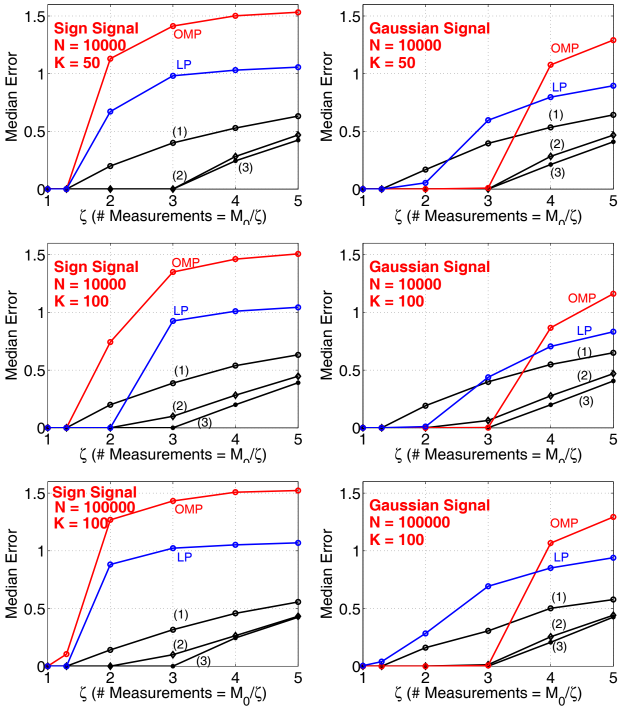

Figure 8: Median reconstruction errors (7). When $M = M_0$, all methods perform well. As M decreases, the advantage of our algorithm becomes more obvious, especially with 2 or 3 iterations. Note that, with $M < M_0/3$, the error of our method comes from the fraction of coordinates which our method "gives up", and the reported nonzero coordinates are still very accurate. See Fig. 6.

5.1 Probability Bounds

Our analysis uses the distribution of the ratio of two independent stable random variables, $S_1, S_2 \sim S(\alpha, 1)$. As a closed-form expression is not available, we compute the lower and upper bounds. First, we define

$$F_\alpha(t) = \mathbf{Pr}\left(|S_2/S_1|^{\alpha/(1-\alpha)} \leq t\right), \quad t \geq 0 \tag{8}$$

where

$$|S_2/S_1|^{\alpha/(1-\alpha)} = Q_\alpha \frac{w_1}{w_2}, \;\; Q_\alpha = Q_\alpha(u_1, u_2) = \left|\frac{q_\alpha(u_2)}{q_\alpha(u_1)}\right|^{\alpha/(1-\alpha)},$$

$$q_\alpha(u) = \frac{\sin(\alpha u)}{\cos^{1/\alpha} u}\left[\cos(u - \alpha u)\right]^{(1-\alpha)/\alpha}$$

based on (2) for generating α-stable random variables. The following lemmas provide useful bounds for $F_\alpha(t)$.

Lemma 1. *For all $t \geq 0$,*

$$F_\alpha(t) = E\left(\frac{1}{1 + Q_\alpha/t}\right) \tag{9}$$

$$\geq \max\left\{\frac{1/2}{1 + 1/t}, \; \frac{1 + (1/t - 3)\mathbf{Pr}(Q_\alpha \leq t)/2}{1 + 1/t}\right\}$$

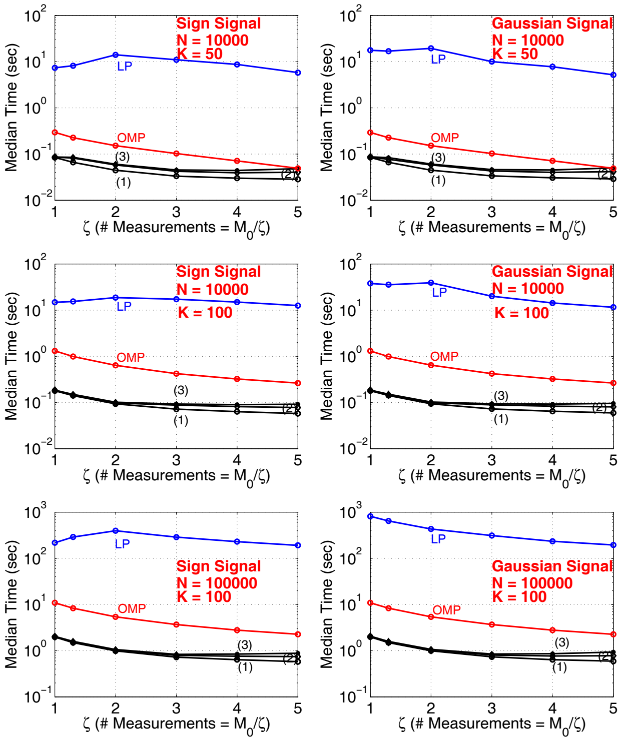

Figure 9: Median reconstruction times, for comparing our proposed algorithm with OMP and LP. When K is not small, our method can be significantly more efficient than OMP. LP (l_1-magic) is expensive.

In particular, when $t \le 1/3$, $F_\alpha(t) \ge \frac{1}{1+1/t}$. In addition, for any $t \ge 0$, $\lim_{\alpha\to 0} F_\alpha(t) = \frac{1}{1+1/t}$. □

Lemma 2. *If $0 < \alpha < 1/3$, then*

$$F_\alpha(t) \le C_\alpha t^{\frac{1-\alpha}{1+\alpha}} \max\{1, t^{\frac{2\alpha}{1+\alpha}}\} \tag{10}$$

$$C_\alpha = \mu_1\mu_2 + \frac{1}{\pi}\left(\mu_2(1-\alpha)\right)^{\frac{1-\alpha}{1+\alpha}}\left(\frac{1-\alpha}{\alpha}\right)^{\frac{2\alpha}{1+\alpha}}\left(\frac{1+\alpha}{1-\alpha}\right)$$

$$\mu_1 = \frac{1}{\pi}\frac{\Gamma\left(1/(2-2\alpha)\right)\Gamma\left((1-3\alpha)/(2-2\alpha)\right)}{\Gamma\left((2-3\alpha)/(2-2\alpha)\right)}$$

$$\mu_2 = 1/\cos\left(\pi\alpha/(2-2\alpha)\right)$$

The const. $C_\alpha \to 1+1/\pi$ as $\alpha \to 0$, $C_\alpha < 1.5$ if $\alpha \le 0.05$.□

Lemma 3. *For all $0 < s < t$,*

$$F_\alpha(t) - F_\alpha(s) \le (1-s/t)F_\alpha(t) \le (t/s-1)F_\alpha(s) \quad \square \tag{11}$$

5.2 Analysis of the Minimum Estimator

Recall the definition of the absolute min estimator:

$$\hat{x}_{i,min} = \frac{y_t}{s_{it}}, \qquad \text{where} \ \ t = \underset{1\le j\le M}{\operatorname{argmin}} \left|\frac{y_j}{s_{ij}}\right| \tag{12}$$

If $|\hat{x}_{i,min}| > \epsilon$, then the i-th coordinate is a (candidate of) nonzero entry. The task is to analyze the probability of false positive, $\mathbf{Pr}\left(|\hat{x}_{i,min}| > \epsilon, x_i = 0\right)$, and the probability of false negative, $\mathbf{Pr}\left(|\hat{x}_{i,min}| \le \epsilon, |x_i| > \epsilon\right)$. We should keep in mind that, in the proposed method, i.e., Alg. 1, the minimum estimator is just the first step for filtering out many true zero coordinates. False positives will have chance to be removed by the gap estimator and iterative process.

5.2.1 Analysis of False Positives

Theorem 1. *If $\psi = \left(\frac{\epsilon}{\theta}\right)^{\frac{\alpha}{1-\alpha}} \le \frac{1}{3}$, $\theta^\alpha = \sum_{i=1}^N |x_i|^\alpha$, then*

$$\mathbf{Pr}\left(|\hat{x}_{i,min}| > \epsilon, x_i = 0\right) \le \frac{1}{(1+\psi)^M}. \tag{13}$$

Proof: *$\frac{y_j}{s_{ij}} = \frac{\sum_{t=1}^N x_t s_{tj}}{s_{ij}} = \theta_i\frac{S_2}{S_1} + x_i$, where S_1 and S_2 are i.i.d. $S(\alpha,1)$ variables. When $x_i = 0$, $\frac{y_j}{s_{ij}} = \theta\frac{S_2}{S_1}$. Denote $\gamma = (1-\alpha)/\alpha$, By Lemma 1,*

$$\mathbf{Pr}\left(|\hat{x}_{i,min}| > \epsilon, x_i = 0\right) = \left[\mathbf{Pr}\left(\left|\frac{y_j}{s_{ij}}\right| > \epsilon, x_i = 0\right)\right]^M$$

$$= \left[\mathbf{Pr}\left(|S_2/S_1| > \epsilon/\theta\right)\right]^M = \left[1 - \mathbf{Pr}\left(|S_2/S_1|^{1/\gamma} \le (\epsilon/\theta)^{1/\gamma}\right)\right]^M$$

$$= \left(1 - F_\alpha(\psi)\right)^M \le \left(1 - \frac{1}{1+1/\psi}\right)^M = \frac{1}{(1+\psi)^M} \qquad \square$$

The assumption $\psi = \left(\frac{\epsilon}{\theta}\right)^{\frac{\alpha}{1-\alpha}} \le 1/3$ is very reasonable for small α because $\psi \approx \frac{\epsilon^\alpha}{K} \approx 1/K$, i.e., $1/K < 1/3$.

5.2.2 Required Number of Measurements

We derive the required M, number of measurements, based on the false positive probability in Theorem 1. This result is useful if we just use one iteration, which matches the known complexity bounds in the compressed sensing literature.

Theorem 2. *To ensure that the total number of false positives is bounded by δ, it suffices to let*

$$M \ge \frac{\log\left((N-K)/\delta\right)}{\log(1+\psi)} \qquad \square \tag{14}$$

Since $\psi = \left(\frac{\epsilon}{\theta}\right)^{\frac{\alpha}{1-\alpha}} \approx 1/K$ and $1/\log(1+\psi) \approx K$, we define

$$M_0 = K\log\left((N-K)/\delta\right) \tag{15}$$

as a convenient approximation. Note that the parameter ϵ affects the required M only through ϵ^α. This means our algorithm is not sensitive to the choice of ϵ. For example, when $\alpha = 0.03$, then $(10^{-3})^\alpha = 0.8128$, $(10^{-4})^\alpha = 0.7586$.

5.2.3 Analysis of False Negatives

Theorem 3. *If $\alpha \le 0.05$, $\left(\frac{|x_i|+\epsilon}{\theta_i}\right)^{\alpha/(1-\alpha)} < 1/3$, then*

$$\mathbf{Pr}\left(|\hat{x}_{i,min}| \le \epsilon, |x_i| > \epsilon\right) \tag{16}$$

$$\le \left\{1 - \left[1 - \frac{3}{4}\left|\frac{|x_i|+\epsilon}{\theta_i}\right|^{\frac{\alpha}{1+\alpha}}\left(1 - \left|\frac{|x_i|-\epsilon}{|x_i|+\epsilon}\right|^{\alpha/(1-\alpha)}\right)\right]^M\right\}$$

5.2.4 The Choice of Threshold ϵ

We can better understand the choice of ϵ from the false negative probability as shown in Theorem 3. Assume $x_i \ne 0$ and $|x_i|/\epsilon = H_i \gg 1$, the probability $\mathbf{Pr}\left(|\hat{x}_{i,min}| \le \epsilon, |x_i| > \epsilon\right)$ upper bound is roughly

$$1 - \left[1 - \frac{3/4}{K}\frac{2\alpha}{H_i}\right]^M \approx 1 - e^{-\frac{3/2\alpha M}{KH_i}} \approx \frac{3/2\alpha M}{KH_i}$$

As we usually choose $M \le M_0 = K\log((N-K)/\delta)$, we have $\frac{\alpha M}{KH_i} < \frac{\alpha\log((N-K)/\delta)}{H_i}$. To ensure that all the K nonzero coordinates can be safely detected by the minimum estimator (i.e., the total false negatives should be less than δ), we need

$$\sum_{x_i \neq 0} \frac{1}{H_i} < \frac{\delta}{1.5\alpha \log \frac{N-K}{\delta}}.$$

For sign signals, i.e., $|x_i| = 1$ if $x_i \neq 0$, we need $H_i > 1.5\alpha K \log \frac{N-K}{\delta}/\delta$, or equivalently $\epsilon < \frac{\delta}{1.5\alpha K \log \frac{N-K}{\delta}}$. If $K = 100$ (or 1000), it is sufficient to let $\epsilon = 10^{-4}$ (or 10^{-5}). Note that even with $N = 2^{32}$, $\log(N) = 22$ is still not large.

For general signals, when the smallest H_i dominates $\sum_{x_i \neq 0} \frac{1}{H_i}$, we just need the smallest $H_i > 1.5\alpha \log \frac{N-K}{\delta}/\delta$, without the K term. In our experiments, for simplicity, we let $\epsilon = 10^{-5}$, for both sign signals and Gaussian signals. In general, with the gap estimator and the iterative process, we find the performance is not sensitive to ϵ as long as it is small.

5.3 Analysis of the Gap Estimator

The minimum estimator only detects the locations of nonzero coordinate (in the first iteration). To estimate the magnitudes, we resort to the gap estimator, defined as

$$z_{i,j} = y_j/s_{ij}, \quad z_{i,(1)} \leq z_{i,(2)} \leq ... \leq z_{i,(M)} \tag{17}$$

$$j_i = \underset{1 \leq j \leq M-1}{\text{argmin}} \; \left\{ z_{i,(j+1)} - z_{i,(j)} \right\} \tag{18}$$

$$\hat{x}_{i,gap} = \frac{z_{i,(j_i)} + z_{i,(j_i+1)}}{2} \tag{19}$$

Theorem 4. *Let $\gamma = (1-\alpha)/\alpha$. Suppose the existence of $a_0 > 1$ and integer $k_0 > 1$ satisfying $\theta\left(a_0 k_0/(M + a_0 k_0)\right)^{\gamma} \leq \epsilon$, i.e., $a_0 k_0 \leq M/K^*_{\epsilon,\theta,\alpha}$, where $K^*_{\epsilon,\theta,\alpha} = 1/\left(\epsilon/\theta\right)^{\alpha/(1-\alpha)} - 1$. Then*

$$\mathbf{Pr}\left(|\hat{x}_{i,gap} - x_i| > \epsilon\right) \leq G_{M,K^*_{\epsilon,\theta,\alpha}} \tag{20}$$

$$= \min_{a_0,k_0} \left\{ B(M, a_0 k_0/M) + \sum_{k=k_0}^{M-2} \left(1 + \frac{1}{2k}\right) \eta_{k,\gamma} \right\}$$

where $B(M, a_0 k_0/M) = \mathbf{Pr}\left(Binomial(M, a_0 k_0/M) < k_0\right)$ is the binomial CDF, and $\eta_{k,\gamma}$ is defined as

$$\eta_{k,\gamma} = \min\left\{ u \in (0,1) : 2\left(1 - \left(\frac{u}{2k}\right)^{1/k}\right)^{\gamma} + \left(1 - \frac{u}{2k}\right)^{\gamma} \leq 1 \right\}$$

□

Since k_0 in Theorem 4 only takes finite values, we can basically numerically evaluate $G_{M,K^*_{\epsilon,\theta,\alpha}}$ to obtain the upper bound for $\mathbf{Pr}\left(|\hat{x}_{i,gap} - x_i| > \epsilon\right)$. It turns out that, once α and ϵ are fixed, G is only a function of $K^* = K^*_{\epsilon,\theta,\alpha}$ and the ratio $\frac{M}{K^*_{\epsilon,\theta,\alpha}}$. Also, note that $K^* \approx K/\epsilon^{\alpha}$.

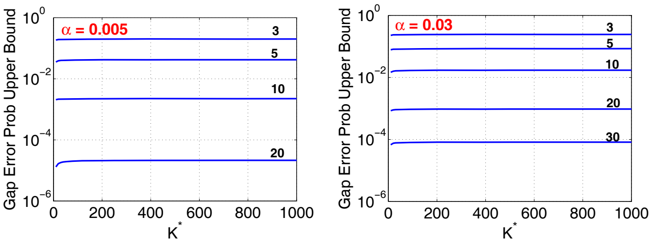

Figure 10: Values of the upper bound of the error probability $\mathbf{Pr}\left(|\hat{x}_{i,gap} - x_i| > \epsilon\right) \leq G_{M,K^*_{\epsilon,\theta,\alpha}}$ as computed in (20). The labels on the curves are the values of $M/K^*_{\epsilon,\theta,\alpha}$.

Fig. 10 plots the upper bound $G_{M,K^*_{\epsilon,\theta,\alpha}}$ for $\alpha = 0.005$ and 0.03, in terms of K^* and $\frac{M}{K^*}$. For example, when using $M = 5K^*$ and $\alpha = 0.005$ / 0.03, the error probabilities are 0.042 / 0.084. In other word, in order for the error probability to be ≤ 0.05, it suffices to use $M = 5K^*$ if $\alpha = 0.005$. When $\alpha = 0.03$, we will have to use respectively $M = 7K^*$ measurements in order to achieve error probability < 0.05.

This way, the required sample size can be numerically computed from Theorem 4. Of course, these numerical values are just the (possibly very conservative) upper bounds.

5.4 Connection to the "Idealized" Algorithm

The error probability bound $\mathbf{Pr}\left(|\hat{x}_{i,gap} - x_i| > \epsilon\right)$ in (20) has two parts: $\min_{a_0,k_0} \left\{ B(M, a_0 k_0/M) + \sum_{k=k_0}^{M-2} \left(1 + \frac{1}{2k}\right) \eta_{k,\gamma} \right\}$, where $B(M, a_0 k_0/M)$ corresponds to the error from the "idealized" algorithm (assuming $\alpha \to 0$) and $\sum_{k=k_0}^{M-2} \left(1 + \frac{1}{2k}\right) \eta_{k,\gamma}$ is the adjustment due to the use of $\alpha > 0$. It is clear from the definition of $\eta_{k,\gamma}$, when $\alpha = 0$, we have $\eta_{k,\gamma} = 0$ and $\min_{a_0,k_0} \left\{ B(M, a_0 k_0/M) = \mathbf{Pr}\left(Binomial(M, 1/K) < 2\right) \right\}$, which is exactly the probability that one or zero observation falls in the region $(x_i - e, x_i + e)$ with $e \to 0$.

The error bound (20) holds for any small α and ϵ. With a fixed small α, we can use enough measurements to bound $\mathbf{Pr}\left(|\hat{x}_{i,gap} - x_i| > \epsilon\right)$ even for a very small ϵ (for example, below the required precision). This means we can remove the reliably estimated x_i and improve the reconstruction by iterations. This is why our procedure requires significantly smaller number of measurements than $K \log N/\delta$.

6. MEASUREMENT NOISE

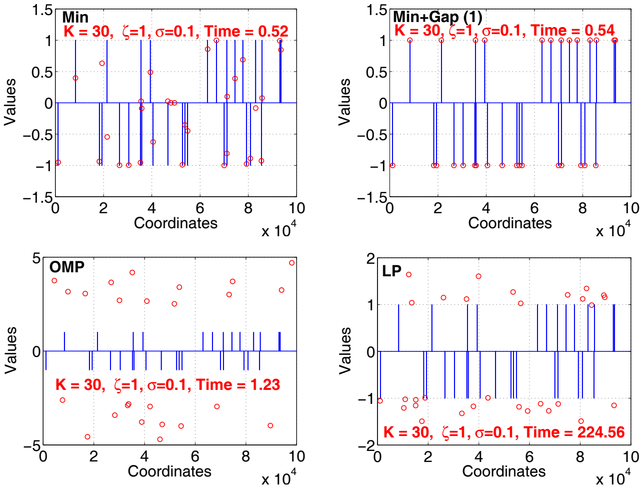

Figure 11: Reconstruction results from one simulation, with $N = 100000$, $K = 30$, $M = M_0$ (i.e., $\zeta = 1$), $\sigma = 0.1$, and sign signals. With the proposed method, the signal is perfectly reconstructed in one iteration. In comparisons, both OMP and LP perform very poorly.

Our method is robust against measurement noise. In the literature, it is common to assume additive measurement noise $\mathbf{y} = \mathbf{xS} + \mathbf{n}$, where each n_j is typically assumed to be $n_j \sim Normal\left(0, \sigma^2 N\right)$. We present a set of experiments with additive noise in Fig. 11. With $N = 100000$, $K = 30$, and $M = M_0$ (i.e., $\zeta = 1$), we have seen in the simulations in Sec. 4 that all methods perform well (when $\sigma = 0$). In the presence of additive measurement noises ($\sigma = 0.1$), Fig. 11

illustrates that our proposed method still achieves perfect recovery while LP and OMP fail.

To understand this interesting phenomenon, we examine

$$\frac{y_j + n_j}{s_{ij}} = x_i + \theta_i \frac{S_2}{S_1} + \frac{n_j}{S_1}$$

Without measurement noise, our algorithm uses observations with $S_2/S_1 \approx 0$ to recover x_i. For those "useful" observations, most likely $|S_1|$ is very large. When S_1 is small, $\frac{n_j}{S_1}$ might be large but very likely $\frac{S_2}{S_1}$ will be large as well (i.e., the observation is not useful anyway). This intuition explains why our method is indifferent to measurement noise.

7. CONCURRENT WORK

In parallel to this paper, we concurrently develop a new sparse recovery algorithm [19] using maximally-skewed stable random projections [17, 18], which has a number of significant advantages over the method in this paper: (i) It allows thorough theoretical analysis at least for $\alpha \in (0,\ 0.5]$, not just for α very close to 0. (ii) Both the theory and estimation procedure are much simpler. (iii) The accuracy is not as sensitive to α. The disadvantage is that [19] is restricted to nonnegative signals (which are common).

In addition, we also develop a sparse recovery algorithm based on "very sparse" matrices [20], using an idea similar to *very sparse stable random projections* [15] in KDD'07.

8. CONCLUSION

Compressed sensing has been an active area of research, as many important applications can be formulated as sparse recovery problems, for example, anomaly detections. In this paper, we present our study of using L0 projections for highly efficient exact sparse recovery. Our proposed procedure consists of the minimum estimator (for detection), the gap estimator (for estimation), and the iterative process. The procedure is able to produce accurate recovery results with smaller number of measurements, compared to LP and OMP using traditional Gaussian (or Gaussian-like) design matrix. Our method utilizes the α-stable distribution with $\alpha \approx 0$. The reported Matlab experiments use $\alpha = 0.03$. The algorithm is robust against measurement noises. Even without sufficient measurements, our method produces stable (partial) recovery results with no catastrophic failures.

9. ACKNOWLEDGEMENT

Ping Li is partially supported by ONR-N000141310261, NSF-1131848, NSF-1249316, and AFOSR-FA9550-13-1-0137. Cun-Hui Zhang is partially supported by NSF-1209014, NSF-1106753, and NSA H98230-11-1-0205.

10. REFERENCES

[1] E. Candès and J. Romberg. l_1-magic: Reocvery of sparse signals via convex programming. Technical report, Calinfornia Institute of Technology, 2005.

[2] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52(2):489–509, 2006.

[3] J. M. Chambers, C. L. Mallows, and B. W. Stuck. A method for simulating stable random variables. *Journal of the American Statistical Association*, 71(354):340–344, 1976.

[4] S. S. Chen, D. L. Donoho, Michael, and A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.

[5] G. Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithm*, 55(1):58–75, 2005.

[6] N. Cressie. A note on the behaviour of the stable distributions for small index. *Z. Wahrscheinlichkeitstheorie und Verw. Gebiete*, 31(1):61–64, 1975.

[7] D. L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52(4):1289–1306, 2006.

[8] D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *Information Theory, IEEE Transactions on*, 40(7):2845–2862, nov. 2001.

[9] D. L. Donoho, A. Maleki, and A. Montanari. Message-passing algorithms for compressed sensing. *PNAS*, 106(45):18914–18919, 2009.

[10] D. L. Donoho and P. B. Stark. Uncertainty principles and signal recovery. *SIAM Journal of Applied Mathematics*, 49(3):906–931, 1989.

[11] D. L. Donoho and J. Tanner. Counting faces of randomly projected polytopes when the projection radically lowers dimension. *Journal of the American Mathematical Society*, 22(1), jan. 2009.

[12] A. K. Fletcher and S. Rangan. Orthogonal matching pursuit from noisy measurements: A new analysis. In *NIPS*. 2009.

[13] A. Gilbert and P. Indyk. Sparse recovery using sparse matrices. *Proc. of the IEEE*, 98(6):937 –947, june 2010.

[14] P. Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of ACM*, 53(3):307–323, 2006.

[15] P. Li. Very sparse stable random projections for dimension reduction in l_α $(0 < \alpha \leq 2)$ norm. In *KDD*, San Jose, CA, 2007.

[16] P. Li. Estimators and tail bounds for dimension reduction in l_α $(0 < \alpha \leq 2)$ using stable random projections. In *SODA*, pages 10 – 19, San Francisco, CA, 2008.

[17] P. Li. Improving compressed counting. In *UAI*, Montreal, CA, 2009.

[18] P. Li and C.-H. Zhang. A new algorithm for compressed counting with applications in shannon entropy estimation in dynamic data. In *COLT*, 2011.

[19] P. Li, C.-H. Zhang, and T. Zhang. Compressed counting meets compressed sensing. Technical report, 2013.

[20] P. Li, C.-H. Zhang, and T. Zhang. Sparse recovery with very sparse compressed counting. Technical report, 2013.

[21] T.-H. Lin and H. T. Kung. Compressive sensing medium access control for wireless lans. In *Globecom*, 2012.

[22] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397 –3415, 1993.

[23] S. Muthukrishnan. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1:117–236, 2 2005.

[24] G. Samorodnitsky and M. S. Taqqu. *Stable Non-Gaussian Random Processes.* Chapman & Hall, New York, 1994.

[25] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society B*, 58(1):267–288, 1996.

[26] J. Tropp. Greed is good: algorithmic results for sparse approximation. *Information Theory, IEEE Transactions on*, 50(10):2231 – 2242, oct. 2004.

[27] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.*, 31(2):890–912, 2008.

[28] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk. Efficient and reliable low-power backscatter networks. In *SIGCOMM*, pages 61–72, Helsinki, Finland, 2012.

[29] M. Wang, W. Xu, E. Mallada, and A. Tang. Sparse recovery with graph constraints: Fundamental limits and measurement construction. In *Infomcom*, 2012.

[30] T. Zhang. Sparse recovery with orthogonal matching pursuit under rip. *Information Theory, IEEE Transactions on*, 57(9):6215 –6221, sept. 2011.