Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2008-001 January 10, 2008

Sparse recovery using sparse matrices

Radu Berinde and Piotr Indyk

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Sparse recovery using sparse random matrices

Radu Berinde
MIT
texel@mit.edu

Piotr Indyk
MIT
indyk@mit.edu

January 10, 2008

Abstract

We consider the *approximate sparse recovery problem*, where the goal is to (approximately) recover a high-dimensional vector x from its lower-dimensional *sketch* Ax. A popular way of performing this recovery is by finding $x^\#$ such that $Ax = Ax^\#$, and $||x^\#||_1$ is minimal. It is known that this approach "works" if A is a random *dense* matrix, chosen from a proper distribution.

In this paper, we investigate this procedure for the case where A is *binary* and very *sparse*. We show that, both in theory and in practice, sparse matrices are essentially as "good" as the dense ones. At the same time, sparse binary matrices provide additional benefits, such as reduced encoding and decoding time.

1 Introduction

Over the recent years, a new approach for obtaining a succinct approximate representation of n-dimensional vectors (or signals) has been discovered. For any signal x, the representation is equal to Ax, where A is a carefully chosen $m \times n$ matrix; the vector Ax is often referred to as the *measurement vector* or *sketch* of x. Although m is typically much smaller than n, the sketch Ax often contains plenty of useful information about the signal x.

The linearity of the sketching method is very convenient for many applications. In *data stream computing* [Mut03, Ind07], the vectors x are often very large, and are not represented explicitly (for example, x_i could denote the total number of packets with destination i passing through a network router). Instead, it is preferable to maintain the sketch Ax under updates to x (e.g., if a new packet arrives, the corresponding coordinate of x is incremented by 1). This is easy to do if the sketching procedure is linear. In *compressed sensing* [CRT04, Don06, DDTW+08], the data acquisition itself is done in (analog or digital) hardware, which is capable of computing a dot product of the measurement vector and the signal at a unit cost.

In this paper, we focus on using linear sketches Ax to compute *sparse approximations* of x. Formally, we say that a vector y is k-sparse if it contains at most k non-zero entries. The goal is to find a vector $x^\#$ such that the ℓ_p *approximation error* $||x - x^\#||_p$ is at most $C > 0$ times the smallest possible ℓ_q approximation error $||x - x^*||_q$, where x^* ranges over all k-sparse vectors. For the algorithms given in this paper we have $p = q$.

There have been many algorithms for this problem developed over the last few years. The

Paper	R/D	Sketch length	Encoding	Update	Decoding	Approximation error	Noise
[CCFC02]	R	$k \log^c n$	$n \log^c n$	$\log^c n$	$k \log^c n$	$\ell_2 \leq C\ell_2$	
[CM06]	R	$k \log n$	$n \log n$	$\log n$	$n \log n$	$\ell_2 \leq C\ell_2$	
[CM04]	R	$k \log^c n$	$n \log^c n$	$\log^c n$	$k \log^c n$	$\ell_1 \leq C\ell_1$	
	R	$k \log n$	$n \log n$	$\log n$	$n \log n$	$\ell_1 \leq C\ell_1$	
[CRT06]	D	$k \log(n/k)$	$nk \log(n/k)$	$k \log(n/k)$	LP	$\ell_2 \leq \frac{C}{k^{1/2}}\ell_1$	Y
	D	$k \log^c n$	$n \log n$	$k \log^c n$	LP	$\ell_2 \leq \frac{C}{k^{1/2}}\ell_1$	Y
[GSTV06]	D	$k \log^c n$	$n \log^c n$	$\log^c n$	$k \log^c n$	$\ell_1 \leq C \log n \ell_1$	
[GSTV07]	D	$k \log^c n$	$n \log^c n$	$\log^c n$	$k^2 \log^c n$	$\ell_2 \leq \frac{C}{k^{1/2}}\ell_1$	
[NV07]	D	$k \log^c n$	$n \log n$	$k \log^c n$	OMP	$\ell_2 \leq \frac{C(\log n)^{1/2}}{k^{1/2}}\ell_1$	Y
[GIKS07]	D	$k \log^c n$	$n \log^c n$	$\log^c n$	LP	$\ell_1 \leq C \log^c n \ell_1$	Y
	D	$kn^{c(p-1)}$	$n^{1+c(p-1)}$	$n^{c(p-1)}$	LP	$\ell_p \leq \frac{C}{(1-p)^2 k^{1-1/p}}\ell_1$	Y
[GLR08] (k "large")	D	$k(\log n)^{c \log\log\log n}$	kn^{1-a}	n^{1-a}	LP	$\ell_2 \leq \frac{C}{k^{1/2}}\ell_1$	
This paper	D	$k \log(n/k)$	$n \log(n/k)$	$\log(n/k)$	LP	$\ell_1 \leq C\ell_1$	Y

Figure 1: Summary of the sketching methods (only the best known results are shown). For simplicity, we ignore some aspects of the algorithms, such as explicitness or universality of the measurement matrices. Also, we present only the algorithms that work for arbitrary vectors x, while many other results are known for the case where the vector x itself is required to be k-sparse, e.g., see [TG05, DWB05, Don06, XH07]. The columns describe: citation; sketch type: (D)eterministic or (R)andomized, where the latter schemes work for each signal x with probability $1 - 1/n$; sketch length; time to compute Ax given x; time to update Ax after incrementing one of the coordinates of x; time to recover an approximation of x given Ax (below); approximation guarantee (below); does the algorithm tolerate noisy measurements. In the decoding time column, "LP" denotes the time needed to solve a linear program with $O(n)$ variables and constraints, "OMP" denotes the time needed to perform the Orthogonal Matching Pursuit procedure. In the approximation error column, $\ell_p \leq A\ell_q$ means that the algorithm returns a vector $x^\#$ such that $\|x - x^\#\|_p \leq A \min_{x^*} \|x - x^*\|_q$, where x^* ranges over all k-sparse vectors. The parameters $C > 1$, $c \geq 2$ and $a > 0$ denote some absolute constants, possibly different in each row. We assume that $k < n/2$.

best currently known bounds are presented in Figure 1. [1] Generally, most of the methods [2] can be characterized as either *combinatorial* or *geometric*. The combinatorial methods (see, e.g., [GGI+02a, CCFC02, CM04, DWB05, CM06, GSTV06, GSTV07, XH07]) use sparse measurement matrices, and iterative methods to finding the sparse approximation. The geometric methods (see [CRT04, CRT06, Don06] and many other papers [Gro06]) use dense matrices, and linear or convex programming for finding the approximations. Given that each of those approaches has pros and cons, there have been recent papers that attempt to unify them [NV07, XH07, GIKS07, GLR08]. In particular, the paper [GIKS07] is very closely related to our work, as we elaborate below.

In this paper, we focus on the recovery method that computes a solution $x^\#$ to the linear

[1] Some of the papers, notably [CM04], are focused on somewhat different formulation of the problem. However, it is known that the guarantees presented in the table hold for those algorithms as well. See Lecture 4 in [Ind07] for a more detailed discussion.

[2] Other methods include the *algebraic* approach of [Man92, GGI+02b].

program

$$(P_1) \qquad min\ ||x^{\#}||_1 \text{ subject to } Ax^{\#} = Ax$$

It is known that this approach "works" for any signal x, if the matrix A satisfies the (Bk, b)-*RIP* property, for some "sufficient" constants $B > 1$ and $b > 1$ [CRT04, CRT06]. [3] A matrix A has that property if there exists a scaling factor S such that for any Bk-sparse vector x we have $||x||_2 \leq S||Ax||_2 \leq b||x||_2$. It is known that random Gaussian or Fourier matrices of dimensions as in the Figure 1 satisfy such property with high probability. Moreover, if a matrix satisfies this property, then the LP decoding procedure can be made resilient to measurement noise [CRT06].

In this paper, we consider a different class of matrices. These matrices are *binary* and *sparse*, i.e., they have only a fixed small number d of ones in each column, and all the other values are equal to zero. It was shown [GIKS07] that such matrices can satisfy a weaker form of the RIP property (called *RIP-p* property), where the ℓ_2 norm is replaced by the ℓ_p norm for $p \approx 1$. They also show that even this weaker property suffices for LP decoding (and is tolerant to measurement noise), although the approximation factor becomes super-constant (see Figure 1). The advantage of such matrices is its efficient update time, equal to the sparsity parameter d. In addition, such matrices (with somewhat weaker measurement bounds) can be constructed explicitly using *expander graphs*.

The contribution of this paper is two-fold. First, we prove (Theorem 1 in Section 2.3) that the approximation factor can be reduced to a small constant (which can be made arbitrarily close to 2) and simultaneously reduce the number of measurements to the best known bound of $O(k \log(n/k))$. As a benefit, we also reduce the update time to $O(\log(n/k))$, and the encoding time to $O(n \log(n/k))$ (or less, if the encoded vector is already sparse). This also speeds-up the decoding, given that the linear program is typically solved using the interior-point method, which repeatedly performs the matrix-vector multiplication. The proof of the aforementioned result uses the RIP-1 property of sparse matrices shown in [GIKS07], as well as a variant of the sparse recovery argument of [KT07].

We also report experimental results which indicate that, in practice, binary sparse matrices are as "good" as random Gaussian or Fourier matrices when used in LP decoding (both in terms of the number of necessary measurements, and in terms of the recovery error). At the same time, the LP decoding is noticeably faster for sparse matrices.

1.1 Notation

For any set S that is a subset of some "universe" set U (specified in the context), we use S^c to denote the complement of S in U, i.e., the set $U - S$. Also, for any vector $x \in \mathbb{R}^n$, and any set $S \subset \{1 \ldots n\}$, we define the vector $x_S \in \mathbb{R}^n$ (not $\mathbb{R}^{|S|}$) such that $(x_S)_i = x_i$ for $i \in S$, and $(x_S)_i = 0$ otherwise.

2 Theory

2.1 Definitions

Definition 1. A (l, ϵ)-*unbalanced expander* is a bipartite simple graph $G = (A, B, E)$, $|A| = n, |B| = m$, with left degree d such that for any $X \subset A$ with $|X| \leq l$, the set of neighbors $N(X)$ of X has size $|N(X)| \geq (1 - \epsilon)d|X|$.

[3] For convenience, the definition of the RIP property in this paper differs slightly from the original one.

We also define $E(X:Y) = E \cap (X \times Y)$ to be the set of edges between the sets X and Y.

The following well-known proposition can be shown using Chernoff bounds.

Claim 1. For any $n/2 \geq l \geq 1$, $\epsilon > 0$, there exists a (l, ϵ)-unbalanced expander with left degree $d = O(\log(n/l)/\epsilon)$ and right set size $O(ld/\epsilon) = O(l \log(n/l)/\epsilon^2)$.

Let A be an $m \times n$ adjacency matrix of an unbalanced $(2k, \epsilon)$-expander G with left degree d. Let $\alpha(\epsilon) = (2\epsilon)/(1-2\epsilon)$.

2.2 L1 Uncertainty Principle

Lemma 1. *Consider any $y \in \mathbb{R}^n$ such that $Ay = 0$, and let S be any set of k coordinates of y. Then we have*

$$\|y_S\|_1 \leq \alpha(\epsilon)\|y\|_1$$

Proof. Without loss of generality, we can assume that S consists of the largest (in magnitude) coefficients of y. We partition coordinates into sets $S_0, S_1, S_2, \ldots S_t$, such that (i) the coordinates in the set S_l are not-larger (in magnitude) than the coordinates in the set S_{l-1}, $l \geq 1$, and (ii) all sets but S_t have size k. Therefore, $S_0 = S$. Let A' be a submatrix of A containing rows from $N(S)$.

By Theorem 10 of [GIKS07] we know that $\|A'y_S\|_1 = \|Ay_S\|_1 \geq d(1-2\epsilon)\|y_S\|_1$. At the same time, we know that $\|A'y\|_1 = 0$. Therefore

$$\begin{aligned}
0 = \|A'y\|_1 &\geq \|A'y_S\|_1 - \sum_{l \geq 1} \sum_{(i,j) \in E, i \in S_l, j \in N(S)} |y_i| \\
&\geq d(1-2\epsilon)\|y_S\|_1 - \sum_{l \geq 1} |E(S_l : N(S))| \min_{i \in S_{l-1}} |y_i| \\
&\geq d(1-2\epsilon)\|y_S\|_1 - \sum_{l \geq 1} |E(S_l : N(S))| \cdot \|y_{S_{l-1}}\|_1/k
\end{aligned}$$

From the expansion properties of G it follows that, for $l \geq 1$, we have $|N(S \cup S_l)| \geq d(1-\epsilon)|S \cup S_l|$. It follows that at most $d\epsilon 2k$ edges can cross from S_l to $N(S)$, and therefore

$$\begin{aligned}
0 &\geq d(1-2\epsilon)\|y_S\|_1 - \sum_{l \geq 1} |E(S_l : N(S))| \cdot \|y_{S_{l-1}}\|_1/k \\
&\geq d(1-2\epsilon)\|y_S\|_1 - d\epsilon 2k \sum_{l \geq 1} \|y_{S_{l-1}}\|_1/k \\
&\geq d(1-2\epsilon)\|y_S\|_1 - 2d\epsilon\|y\|_1
\end{aligned}$$

It follows that $d(1-2\epsilon)\|y_S\|_1 \leq 2d\epsilon\|y\|_1$, and thus $\|y_S\|_1 \leq (2\epsilon)/(1-2\epsilon)\|y\|_1$. □

2.3 LP recovery

The following theorem provides recovery guarantees for the program P_1, by setting $u = x$ and $v = x^\#$.

Theorem 1. *Consider any two vectors u, v, such that for $y = v - u$ we have $Ay = 0$, and $\|v\|_1 \leq \|u\|_1$. Let S be the set of k largest (in magnitude) coefficients of u. Then*

$$\|v - u\|_1 \leq 2/(1-2\alpha(\epsilon)) \cdot \|u_{S^c}\|_1$$

Proof. We have

$$\begin{aligned} \|u\|_1 \geq \|v\|_1 &= \|(u+y)_S\|_1 + \|(u+y)_{S^c}\|_1 \\ &\geq \|u_S\|_1 - \|y_S\|_1 + \|y_{S^c}\|_1 - \|u_{S^c}\|_1 \\ &= \|u\|_1 - 2\|u_{S^c}\|_1 + \|y\|_1 - 2\|y_S\|_1 \\ &\geq \|u\|_1 - 2\|u_{S^c}\|_1 + (1-2\alpha(\epsilon))\|y\|_1 \end{aligned}$$

where we used Lemma 1 in the last line. It follows that

$$2\|u_{S^c}\|_1 \geq (1-2\alpha(\epsilon))\|y\|_1$$

□

Theorem 2. *Consider any two vectors u, v, such that for $y = v - u$ we have $\|Ay\|_1 = \beta \geq 0$, and $\|v\|_1 \leq \|u\|_1$. Let S be the set of k largest (in magnitude) coefficients of u. Then*

$$\|v - u_S\|_1 \leq 2/(1-2\alpha(\epsilon)) \cdot \|u_{S^c}\|_1 + \frac{2\beta}{d(1-2\epsilon)(1-2\alpha)}$$

Proof. Analogous to the proof of Theorem 1. □

3 Experiments

A binary sparse matrix of m rows and n columns is generated in the following way: for each column, d random values between 1 and m are generated, and $1's$ are placed in that column, in rows corresponding to the d numbers. If the d numbers are not distinct, the generation for the column is repeated until they are (this is not really an issue when $d \ll m$). Note that such a matrix is the adjacency matrix of an expander graph of degree d with high probability (for a proper value of d).

It is not desirable for a matrix to have identical columns: if (for example) the first two columns of a matrix A are identical, then sparse vectors like $[-1, +1, 0, 0, \ldots]$ are in the null-space of A and thus are not LP recoverable. In practice, we found that the value of d doesn't matter much; for most experiments in this paper $d = 8$ was used, but other values like $4, 10, 16$ usually yield similar results.

3.1 Comparison with Gaussian matrices

We performed experiments that compare the performance of sparse matrices with that of Gaussian matrices. In addition, we also performed each experiment using a *real-valued scrambled Fourier ensemble*, obtained from the Fourier transform matrix by randomly permuting the columns, selecting a random subset of rows, and separating the real and complex values of each element into two real values (see [CRT06]). In practice, with high probability for a given signal, the scrambled sine and cosine functions in the ensemble rows are effectively similar to Gaussian noise, and we can use this ensemble instead of Gaussian matrices with the advantage of $O(n)$ space usage and $O(n \log n)$ multiplication time, making this matrix feasible for experiments with large signals, like images.

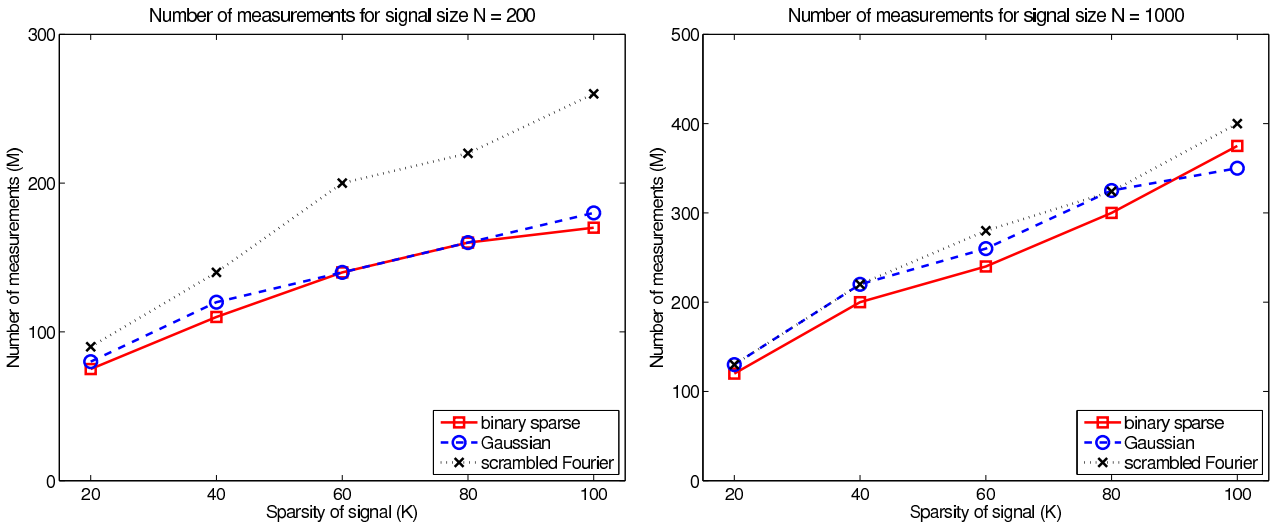


Figure 2: Number of measurements necessary for fixed signal size n and variable sparsity k

3.2 Synthetic data

This section describes experiments that use synthetic data. Two type of experiments are presented: exact recovery, in which the signal is truly k-sparse, and approximate recovery, in which the signal allows a good k-sparse approximation. The k-sparse signals used were generated by randomly choosing k positions for the peaks, and randomly setting the peaks to either $+1$ or -1. For experiments which required signals which allow good k-sparse approximations, random Gaussian noise of a certain magnitude was added to the generated k-sparse signal.

All the experiments involve solving the linear program P_1. This was accomplished using the ℓ_1-MAGIC package for Matlab (see [L1MAGIC]).

3.2.1 Exact recovery of sparse signals

Experiments in this section deal with exact recovery of a sparse signal; the number of measurements necessary for reconstruction is recorded.

Each experiment starts with generating a signal x_0 of size n which contains only k non-zero values ± 1. The locations and signs of the peaks are chosen randomly. For a chosen number of measurements m, an $m \times n$ matrix is generated, the measurement vector $y = Ax_0$ is computed, and a vector $x^\#$ is recovered from y by solving the linear program P_1. The random matrix generation and recovery is repeated 10 times (with the same signal), and an experiment is successful if every time the recovery is successful, i.e. $x^\# = x_0$.

Of interest is the smallest m value which allows exact reconstruction of the signal. Note that for all binary sparse matrix experiments, d is a small value (between 4 and 10). The results of these experiments show that the number of measurements necessary in the case of binary sparse matrices is similar to that in the case of Gaussian matrices. An illustration of typical results of such an experiment can be seen in figures 2 and 3. In all these graphs we can see that the necessary number of measurements is similar for the two matrix types. Moreover, figure 3 shows that m grows roughly proportional to k, and figure 2 shows that m grows sublinearly with n (as expected).

3.2.2 Approximate recovery

In another set of experiments we tested recovery of a measured vector x_0 which isn't k-sparse, but which allows for a good k-sparse approximation. The vectors are generated as before, by randomly picking k peaks of values ± 1, but then random Gaussian noise of a certain magnitude is added. The theory says that for all types of measurement matrices, the error in the recovery should be proportional to the magnitude of this noise, as only the noise contributes to the tail of x_0.

An experiment consists of choosing reasonable values of n, k, m (based on results from exact recovery experiments), and setting the magnitude of the noise in the input. The ℓ_2 error in the recovery $\|x^{\#} - x_0\|_2$ is measured. Each experiment is repeated 10 times, and the maximum recovery error is noted. We show typical results in figures 4 and 5 as plots of recovery error versus input noise magnitude, for fixed n, k, m. In the left part of both figures, the scrambled Fourier ensemble is not shown because respective number of measurements is too small for this type of matrix. Note that for the binary sparse matrices, we used $d = 8$ for all experiments (higher values of d yield similar results). The ℓ_1 error was also measured, but it was always roughly proportional to the ℓ_2 error and thus it didn't provide any additional information (the shapes of the plots are essentially identical).

3.3 Image data

In order to experiment with real data, we performed tests using encodings of images. We show experiments using the boat image from [CRT06], shown in figure 7; similar results occurred with different images.

In order for sparse recovery to be useful for an image, we have to change the basis. We used the two-dimensional wavelet decomposition with the Daubechies-4 wavelet, at decomposition level 3. The boat image is of size 256×256. The corresponding wavelet coefficients vector has size 71542 (the size in larger than 65536 because of the symmetric half-point padding used to prevent border distortion).

We then aim to recover an approximation to the wavelet coefficient vector (and thus an ap-

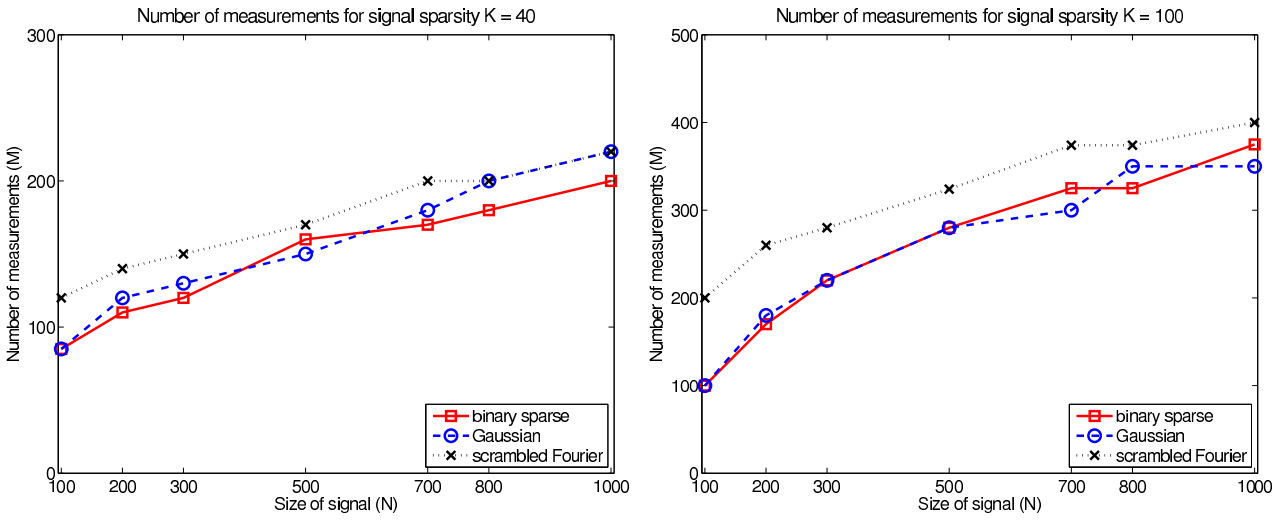


Figure 3: Number of measurements necessary for fixed sparsity k and variable signal size n

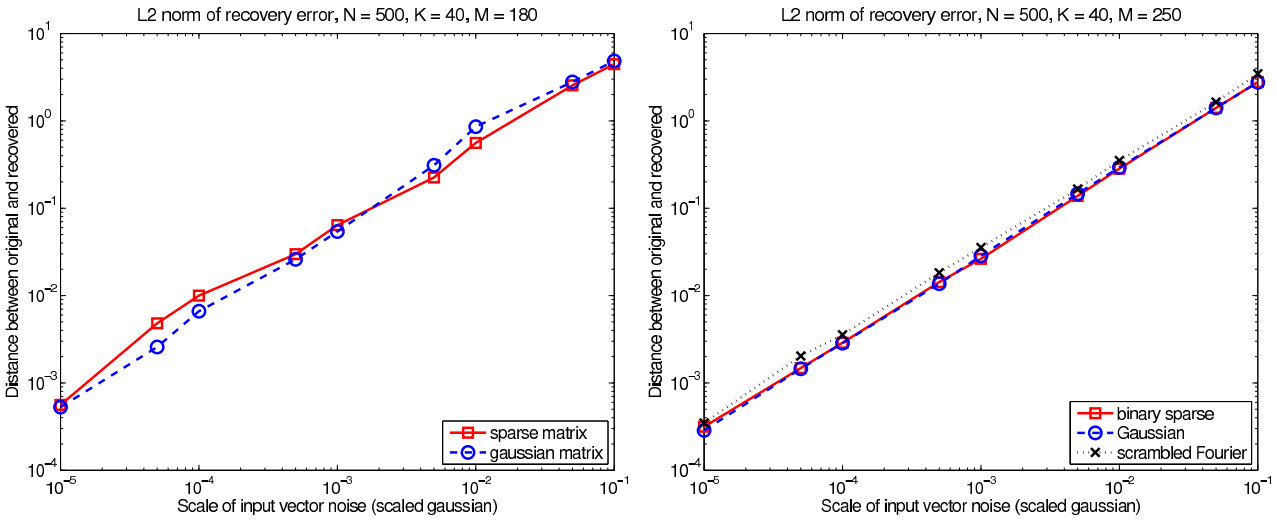


Figure 4: Recovery error versus input noise for $n = 500$, $k = 40$ and $m \in \{180, 250\}$.

proximation of the image) from linear measurements of this vector (as in [CRT06]); or, equivalently, from image measurements of the form AW^*x where A is the measurement matrix and W^* is the matrix that implements the wavelet transform. The experiments use sparse and scrambled Fourier matrices for matrix A; using Gaussian matrices is not feasible because of the huge space and multiplication time requirements. An experiment consists of choosing the number of measurements m and, for sparse matrix experiments, the matrix parameter d. A random matrix is generated and linear measurements of the vector are computed; then the ℓ_1 minimization program is used to recover the vector.

Figure 8 shows images recovered from different number of measurements using binary sparse matrices; results are shown for sparsity $d = 8$; approximations using $d = 16$ were very similar in quality. Figure 9 shows images recovered using scrambled Fourier ensembles.

3.3.1 Min-TV

There is a slightly different model which yields better results for images, avoiding the high-frequency artifacts. Instead of using a wavelet, we recover a sparse approximation of the image *gradient*. Using the following approximation for the gradient

$$|(\nabla x)_{i,j}| \approx \sqrt{(x_{i+1,j} - x_{i,j})^2 + (x_{i,j+1} - x_{i,j})^2}$$

define the *total variance* (TV) norm of x:

$$\|x\|_{TV} = \sum_{i,j} \sqrt{(x_{i+1,j} - x_{i,j})^2 + (x_{i,j+1} - x_{i,j})^2}$$

We can then recover an approximate image x from measurements $y = Ax_0$ by solving

$$\text{(min-TV)} \qquad \min \|x\|_{TV} \text{ subject to } Ax = y$$

In particular, we can recover exactly signals which have sparse gradients ([CRT06a]). See [CRT06] for a similar application of min-TV.

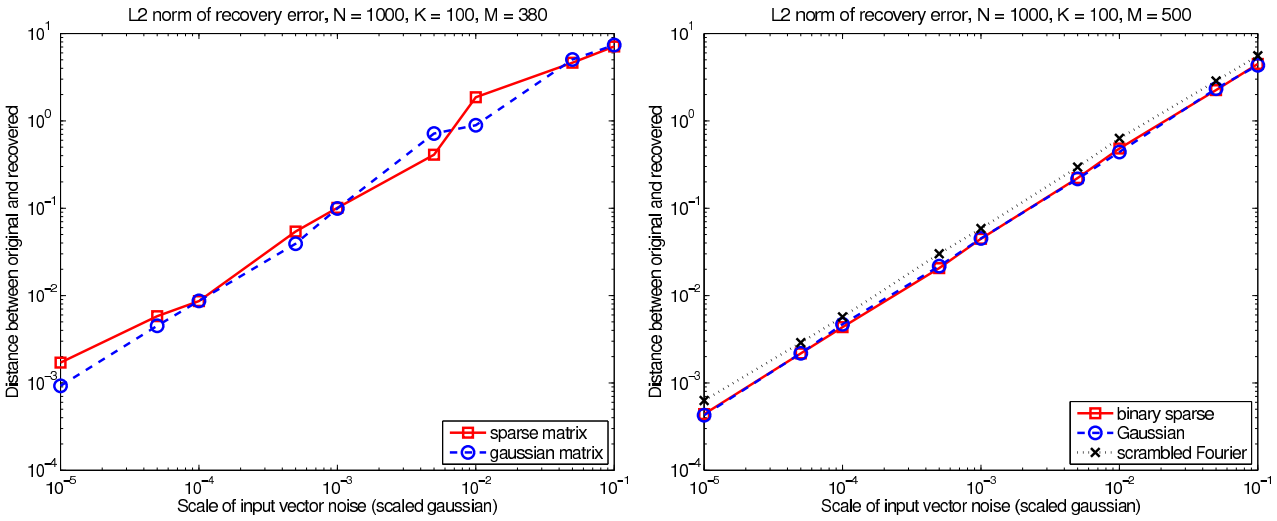


Figure 5: Recovery error versus input noise for $n = 1000$, $k = 100$ and $m \in \{380, 500\}$.

We show the results of min-TV recovery with binary sparse matrices in figure 11. Similar results are obtained with real-valued Fourier ensembles, shown in 12. For all images shown, $d = 8$ was used; other values like 2, 4, or 16 yield very similar results.

3.3.2 Timing experiments

The basic operation inside the recovery programs is the multiplication of the measurement matrix or its transpose with a vector; about half of the operations involve the transpose. Applying an $m \times n$ binary sparse matrix to a vector x takes only $\|x\|_0 \cdot d$ additions, where $\|x\|_0 \leq n$ is the number of non-zero elements of the vector. Applying the $n \times m$ transpose to a vector y takes $\|y\|_0 \cdot nd/m$ additions on average, where $\|y\|_0 \leq m$. The binary sparse matrix multiplication was implemented in a C program, using Matlab's external interface (MEX). Note that for the boat image, the relevant vectors used by the recovery programs were dense, so multiplications effectively took nd additions.

For the scrambled Fourier measurement matrix, the matrix-vector multiplication involves performing the FFT, which runs in $O(n \log n)$. The scrambled Fourier matrix multiplication code is a part of ℓ_1-MAGIC and uses Matlab's `fft` and `ifft` functions.

To investigate the actual running times for different measurement matrices, we performed a few timing experiments. The benchmark was run with Matlab 7.1 on an AMD Athlon XP 2600+ (1.8Ghz). Figure 6 shows the CPU times (averaged over 2 runs) for LP and min-TV recovery of the boat image with $m \in \{10000, 25000\}$ using scrambled Fourier and sparse matrices with $d \in \{4, 8, 16\}$. We mostly used the default parameters for the ℓ_1-MAGIC solver. The only exception was increasing the number of maximum line backtracking iterations inside the primal-dual algorithm to 128. This prevents the program from finishing too early, before the optimum is achieved.[4]

[4]It is plausible that smaller value of the parameter would yield reduced running time for sparse matrices without affecting the accuracy of the output, but we did not exploit this issue. The increased bound does not affect the running time for Fourier matrices, which only incur a limited amount of backtracking.

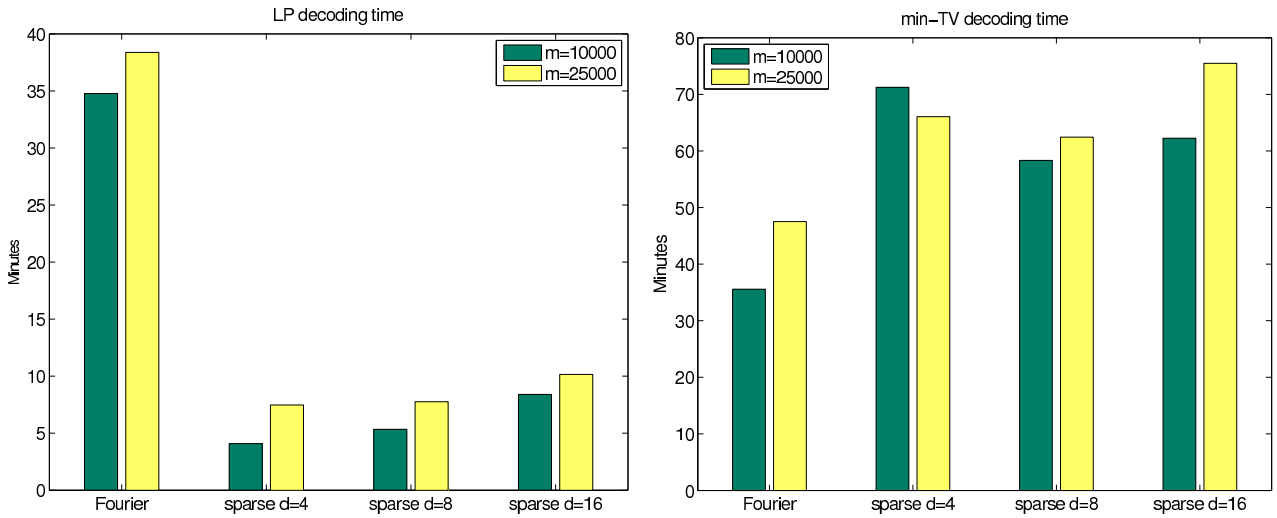


Figure 6: Recovery times in minutes for LP and min-TV recoveries for $m \in \{10000, 25000\}$.

4 Acknowledgments

The authors thank Justin Romberg for his help and clarifications regarding the ℓ_1-MAGIC package. Also, they thank Tasos Sidiropoulos for many helpful comments. Finally, they thank Venkatesan Guruswami, for bringing the reference [KT07] to their attention.

References

[CCFC02] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. *ICALP*, 2002.

[CM04] G. Cormode and S. Muthukrishnan. Improved data stream summaries: The count-min sketch and its applications. *FSTTCS*, 2004.

[CM06] G. Cormode and S. Muthukrishnan. Combinatorial algorithms for Compressed Sensing. *Proc. 40th Ann. Conf. Information Sciences and Systems*, Princeton, Mar. 2006.

[L1MAGIC] E. Candes, J.Romberg. ℓ_1-MAGIC: Recovery of Sparse Signals via Convex Programming, Caltech, *http://www.acm.caltech.edu/l1magic*

[CRT04] E. Candès, J. Romberg, and T. Tao. Exact signal reconstruction from highly incomplete frequency information. Submitted for publication, June 2004.

[CRT06] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8):1208–1223, 2006.

[CRT06a] E. J. Candès, J. Romberg, and T. Tao. Robust Uncertainty Principles: Exact Signal Reconstruction *IEEE Trans. on Info. Theory*, 52(2) pp. 489-509, Feb. 2006.

[Don06] D. L. Donoho. Compressed Sensing. *IEEE Trans. Info. Theory*, 52(4):1289–1306, Apr. 2006.

[DWB05] M. F. Duarte, M. B. Wakin, and R. G. Baraniuk. Fast reconstruction of piecewise smooth signals from random projections. *Proc. SPARS05*, Rennes, France, Nov. 2005.

[GGI+02b] A. Gilbert, S. Guha, P. Indyk, M. Muthukrishnan, and M. Strauss. Near-optimal sparse Fourier representations via sampling. *STOC*, 2002.

[GGI+02a] A. C. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. Fast, small-space algorithms for approximate histogram maintenance. *STOC*, 2002.

[GIKS07] A. Gilbert, P. Indyk, H. Karloff, and M. Strauss. Combining geometry and combinatorics: a unified approach to sparse signal recovery. Manuscript, 2007.

[GLR08] V. Guruswami and J. Lee and A. Razborov. Almost Euclidean subspaces of L1 via expander codes. *SODA*, 2008.

[Gro06] Rice DSP Group. Compressed sensing resources. *Available at* `http://www.dsp.ece.rice.edu/cs/`, 2006.

[GSTV06] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin. Algorithmic linear dimension reduction in the ℓ_1 norm for sparse vectors. Submitted for publication, 2006.

[GSTV07] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin. One sketch for all: fast algorithms for compressed sensing. *STOC 2007*, pages 237–246, 2007.

[Ind07] P. Indyk. Sketching, streaming and sublinear-space algorithms. *Graduate course notes, available at* `http://stellar.mit.edu/S/course/6/fa07/6.895/`, 2007.

[KT07] B. S. Kashin and V. N. Temlyakov. A remark on compressed sensing. Manuscript, 2007.

[Man92] Y. Mansour. Randomized Interpolation and Approximation of Sparse Polynomials. *ICALP*, 1992.

[Mut03] S. Muthukrishnan. Data streams: Algorithms and applications (invited talk at soda'03). Available at `http://athos.rutgers.edu/$\sim$muthu/stream-1-1.ps`, 2003.

[NV07] D. Needell and R. Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. Manuscript, 2007.

[TG05] J. A. Tropp and A. C. Gilbert. Signal recovery from partial information via Orthogonal Matching Pursuit. Submitted to *IEEE Trans. Inform. Theory*, April 2005.

[TLW+06] Dharmpal Takhar, Jason Laska, Michael B. Wakin, Marco F. Duarte, Dror Baron, Shriram Sarvotham, Kevin Kelly, and Richard G. Baraniuk. A new compressive imaging camera architecture using optical-domain compression. *Proc. IS&T/SPIE Symposium on Electronic Imaging*, 2006.

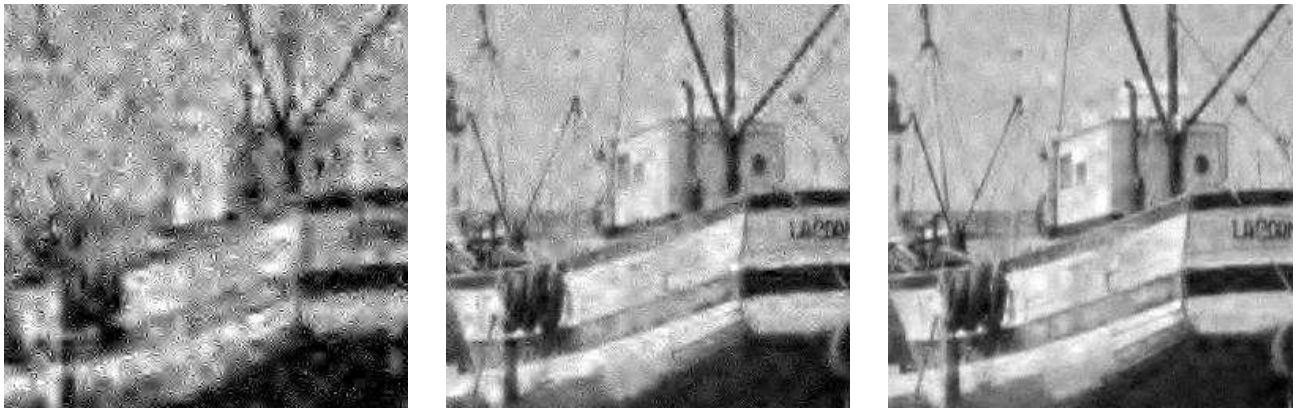
[DDTW+08] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly and R. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 2008, to appear.

[XH07] W. Xu and B. Hassibi. Efficient compressive sensing with deterministic guarantees using expander graphs. *IEEE Information Theory Workshop*, 2007.

Figure 7: The original boat image.

(a) $m = 10000$ (b) $m = 20000$ (c) $m = 30000$

Figure 8: Image decodings of the LP-recovered wavelet vector using binary sparse matrices with (a) 10000 measurements, (b) 20000 measurements, and (c) 30000 measurements.



(a) $m = 10000$ (b) $m = 20000$ (c) $m = 30000$

Figure 9: Image decodings of the LP-recovered wavelet vector using scrambled Fourier matrices with (a) 10000 measurements, (b) 20000 measurements, and (c) 30000 measurements.

Figure 10: The original boat image.

(a) $m = 7500$ (b) $m = 15000$ (c) $m = 25000$

Figure 11: Recovered images using min-TV with binary sparse matrices with (a) 7500 measurements, (b) 15000 measurements, and (c) 25000 measurements.

(a) $m = 7500$ (b) $m = 15000$ (c) $m = 25000$

Figure 12: Recovered images using min-TV with scrambled Fourier matrices with (a) 7500 measurements, (b) 15000 measurements, and (c) 25000 measurements.