Practical Near-optimal Sparse Recovery in the L1 Norm

R. Berinde, P. Indyk and M. Ružić

***Abstract*— We consider the *approximate sparse recovery problem*, where the goal is to (approximately) recover a high-dimensional vector $x \in R^n$ from its lower-dimensional *sketch* $Ax \in R^m$. Specifically, we focus on the sparse recovery problem in the ℓ_1 norm: for a parameter k, given the sketch Ax, compute an approximation $\hat{x}$ of x such that the ℓ_1 approximation error $\|x - \hat{x}\|_1$ is close to $\min_{x'} \|x - x'\|_1$, where x' ranges over all vectors with at most k terms. The sparse recovery problem has been subject to extensive research over the last few years. Many solutions to this problem have been discovered, achieving different trade-offs between various attributes, such as the sketch length, encoding and recovery times.**

A recent paper [IR08] provided a sparse recovery scheme which achieved close to optimal performance on virtually all attributes (see Figure 1). In particular, this was the first recovery scheme that guaranteed $O(k \log(n/k))$ sketch length, and near-linear $O(n \log(n/k))$ recovery time *simultaneously*. This was achieved by using sketching matrices A which were themselves very sparse. The matrix sparsity enabled decreasing the amount of computation spent on encoding and recovery.

In this paper we present a new practical variant of that algorithm, that we call *Sparse Matching Pursuit*, or *SMP*. The running time of the new algorithm is slightly higher (by a logarithmic factor) than of its predecessor, and its sketch length bound remains unchanged. However, its *empirical* sketch length is substantially lower. This makes our scheme an attractive option for sparse recovery problems, both in theory and in practice.

I. 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 $m \times n$ 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 contains plenty of useful information about the signal x.

The linearity of the sketching method is crucial for a wide variety of applications. In the area of *data stream computing* [Mut03], [Ind07], the vectors x are often very large, and cannot be represented explicitly; for example, x_i could denote the total number of packets with destination i passing through a network router. It is thus preferable to maintain instead the sketch Ax, under incremental updates to x. Specifically, if a new packet arrives, the corresponding coordinate of x is incremented by 1. This can be easily done if the sketching procedure is linear. In the area of *compressed sensing* [CRT06a], [Don06], [TLW+06], [DDT+08], the data acquisition itself is done using (analog or digital) hardware, which is capable of computing a dot product of the measurement vector and the signal at a unit cost. Other applications include breaking privacy of databases via aggregate queries [DMT07].

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 $\hat{x}$ such that the ℓ_p approximation error $\|x - \hat{x}\|_p$ is at most $c > 0$ times the smallest possible ℓ_q approximation error $\|x - x'\|_q$, where x' ranges over all k-sparse vectors (we denote this type of guarantee by "$\ell_p \leq c\,\ell_q$"). Note that for any value of q, the error $\|x - \hat{x}\|_q$ is minimized when $\hat{x}$ consists of the k largest (in magnitude) coefficients of x.

The problem was subject to an extensive research over the last few years, in several different research communities, including applied mathematics, digital signal processing and theoretical computer science. The goal of that research is to obtain encoding and recovery schemes with low probability of error (ideally, deterministic[1] schemes), short sketch lengths, low encoding, update and recovery times, good approximation error bounds and resilient to measurement noise. The current state of the art is presented in Figure 1[2].

Our work builds on a recent paper [IR08]. In that paper the authors propose a recovery scheme, called *EMP*, that is based on sparse random (or pseudo-random) matrices. Up to the leading constants, EMP achieves the best known bounds for the sketch length, encoding and update times, and the recovery time (if k is large enough). Also, the scheme is resilient to noise. The only theoretical drawback of that scheme is the $\ell_1 \leq C\ell_1$ error guarantee, which is known [CDD06] to be weaker than the $\ell_2 \leq \frac{C}{k^{1/2}}\ell_1$ guarantee achievable by some of the other algorithms.

Berinde is with the Department of Electrical Engineering and Computer Science, MIT. Email: texel@mit.edu

Indyk is with the Computer Science and Artificial Intelligence Laboratory, MIT. Email: indyk@theory.csail.mit.edu

Ružić is with ITU Copenhagen. Email: milan@itu.dk

RB and PI are supported in part by David and Lucille Packard Fellowship. PI is partially supported by MADALGO (Center for Massive Data Algorithmics, funded by the Danish National Research Association) and NSF grant CCF-0728645.

[1] We use the term "deterministic" for a scheme in which one matrix A works for all signals x, and "randomized" for a scheme that generates a"random" matrix A which, for each signal x, works with probability $1 - 1/n$. However, "deterministic" does not mean "explicit" - we allow the matrix A to be constructed using the probabilistic method.

[2] The figure depicts only the algorithms that work for *arbitrary* signals x. Many other results are known for the case where the vector x itself is required to be k-sparse, e.g., see [TG05], [SBB06b], [SBB06a], [Don06], [XH07].

Paper	R/D	Sketch length	Encoding time	Sparsity/ Update time	Decoding time	Approximation error	Noise
[CCFC02], [CM06]	R	$k \log^d n$	$n \log^d n$	$\log^d n$	$k \log^d n$	$\ell_2 \leq C\ell_2$	
	R	$k \log n$	$n \log n$	$\log n$	$n \log n$	$\ell_2 \leq C\ell_2$	
[CM04]	R	$k \log^d n$	$n \log^d n$	$\log^d n$	$k \log^d n$	$\ell_1 \leq C\ell_1$	
	R	$k \log n$	$n \log n$	$\log n$	$n \log n$	$\ell_1 \leq C\ell_1$	
[CRT06b], [RV06]	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	$k \log^d n$	$n \log n$	$k \log^d n$	LP	$\ell_2 \leq \frac{C}{k^{1/2}}\ell_1$	Y
[GSTV06]	D	$k \log^d n$	$n \log^d n$	$\log^d n$	$k \log^d n$	$\ell_1 \leq C \log n \ell_1$	Y
[GSTV07]	D	$k \log^d n$	$n \log^d n$	$\log^d n$	$k^2 \log^d n$	$\ell_2 \leq \frac{\epsilon}{k^{1/2}}\ell_1$	
[GLR08] (k "large")	D	$k(\log n)^{d \log\log\log n}$	kn^{1-a}	n^{1-a}	LP	$\ell_2 \leq \frac{C}{k^{1/2}}\ell_1$	
[BGI+08]	D	$k \log(n/k)$	$n \log(n/k)$	$\log(n/k)$	LP	$\ell_1 \leq C\ell_1$	Y
[DM08]	D	$k \log(n/k)$	$nk \log(n/k)$	$k \log(n/k)$	$nk \log(n/k) \log R$	$\ell_2 \leq \frac{C}{k^{1/2}}\ell_1$	Y
[NT08]	D	$k \log(n/k)$	$nk \log(n/k)$	$k \log(n/k)$	$nk \log(n/k) \log R$	$\ell_2 \leq \frac{C}{k^{1/2}}\ell_1$	Y
	D	$k \log^d n$	$n \log n$	$k \log^d n$	$n \log n \log R$	$\ell_2 \leq \frac{C}{k^{1/2}}\ell_1$	Y
[IR08]	D	$k \log(n/k)$	$n \log(n/k)$	$\log(n/k)$	$n \log(n/k)$	$\ell_1 \leq (1+\epsilon)\ell_1$	Y
Best bound per column	D	$k \log(n/k)$	$n \log(n/k)$	$\log(n/k)$	$\min[k \log^d n, n \log(n/k)]$	$\ell_2 \leq \frac{\epsilon}{k^{1/2}}\ell_1$	Y
This paper	D	$k \log(n/k)$	$n \log(n/k)$	$\log(n/k)$	$n \log(n/k) \log R$	$\ell_1 \leq C\ell_1$	Y

Fig. 1. Summary of the sparse recovery results. Virtually all references can be found at [Gro06]. All bounds ignore the $O(\cdot)$ constants. We also ignore other aspects of algorithms, such as explicitness or universality of the measurement matrices. The columns describe: citation; sketch type (deterministic or randomized); 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 measurement vectors of the form $Ax+\mu$. The parameters $C > 1$, $d \geq 2$ and $a > 0$ denote some absolute constants, possibly different in each row. The parameter ϵ denotes any positive constant. We assume that $k < n/2$. In the decoding time column LP=LP(n, m, T) denotes the time needed to solve a linear program defined by an $m \times n$ matrix A which supports matrix-vector multiplication (i.e., the encoding) in time T. Heuristic arguments indicate that LP$(n, m, T) \approx \sqrt{n}T$ if the interior-point method is employed. Some of the running times of the algorithms depend on the "precision parameter" R, which is always bounded from the above by the norm of the vector x if its coordinates are integers. It is known [CDD06] that "$\ell_2 \leq \frac{c}{k^{1/2}}\ell_1$" implies "$\ell_1 \leq (1 + O(c))\ell_1$", and that it is impossible to achieve "$\ell_2 \leq C\ell_2$" deterministically unless the number of measurements is $\Omega(n)$.

Although EMP offers excellent asymptotic guarantees, its empirical performance is not so great; specifically, the number of measurements required by the algorithm to achieve correct recovery is suboptimal. For example, our recovery experiments on random signed k-sparse signals of length n, for $k = 50$ and $n = 20000$, show that one typically needs at least 5000 measurements to recover the signal correctly using the EMP algorithm. In comparison, the linear-programming-based recovery algorithm for sparse matrices [BGI+08] requires only about 450 measurements to perform the same task[3].

In this paper we present a new variant of EMP, called *Sparse Matching Pursuit*, or *SMP*. The algorithm is based on the ideas from [IR08] as well as from [BGI+08] and [NT08]. The running time of the new algorithm is slightly higher (by a logarithmic factor) than of EMP. However, empirically, the algorithm performs successful recovery from a significantly smaller number of measurements. In particular, for the instances described above, SMP typically needs about 2000 measurements (see section V for more detailed empirical evaluation of the algorithm). The asymptotic bound on the number of required measurements is still $O(k \log(n/k))$.

The combination of good asymptotic and empirical performance makes our scheme an attractive option for sparse recovery problems, both in theory and in practice.

A. Techniques and Related Work

As in [IR08], our scheme uses a matrix A that is *binary* and *sparse*. Such a matrix can be interpreted as an adjacency matrix of a bipartite graph G, with left vertex set $\{1 \dots n\}$

[3]For both algorithms we used randomly generated 0-1 matrices with column sparsity equal to 20.

and right vertex set $\{1 \dots m\}$. The sparsity immediately implies low encoding and sketch update times. The recovery algorithm is iterative, in the spirit of *Orthogonal Matching Pursuit* [TG05]. In each iteration, the algorithm estimates the difference between the current approximation x^j and the signal x from the (possibly noisy) sketch $A(x^j - x) = Ax^j - Ax$. The estimation u^* of the difference $u = x^j - x$ is obtained by using voting-like mechanism, where each coordinate u^*_i is equal to the median of the coordinates of the sketch vector Au that correspond to the neighbors of i in the graph G. The approximation x^j is updated by u, and the process is repeated.

Our algorithms builds on a large body of work in several areas, such as *data stream computing* [Mut03], [Ind07], *compressed sensing* [CRT06b], [Don06] and *coding theory*. In the following, we describe prior works that inspired and/or are closely related to our algorithm. The reader is referred to a companion paper [BGI$^+$08] for a broader overview of ideas developed in the aforementioned areas.

The most immediate ancestor of SMP is the algorithm presented in [IR08] (which in turn has been influenced by several other data stream algorithms, including those from [GGI$^+$02], [CCFC02], [CM04], [CM06], [GSTV06], [GSTV07]). The algorithm in [IR08] used sparse sketching matrices, and iterative, voting-like mechanism to recover the signal approximation. In order to achieve desired approximation guarantees, however, the algorithm imposed strict conditions on the recovery process. In particular, each coordinate of the approximation was estimated only once, and was never revised again. This process resulted in an increased empirical sketch length.

The analysis and the general algorithmic framework of SMP have been significantly influenced by the ideas from [DM08], [NT08], [BM08]. In those papers the authors presented iterative algorithms for matrices A satisfying the following *Restricted Isometry Property* or *RIP* [CRT06b]: for any k-sparse x, we have $\|Ax\|_2 \approx \|x\|_2$. In turn, the analysis of SMP uses the fact that sparse random matrices satisfy a related property called *RIP-1* [BGI$^+$08], where we require that for any k-sparse x we have $\|Ax\|_1 \approx \|x\|_1$ (i.e., the near-isometry holds in the ℓ_1 norm). As a result, some ideas from the aforementioned papers apply here as well. In particular, we use the decomposition of the input signal x into the "head" $x^{(k)}$ (containing the k most significant components of x) and the "tail" $x - x^{(k)}$, and interpret the "sketch of the tail" term $A(x-x^{(k)})$ as "measurement noise".

The iterative voting procedure used by SMP is reminiscent of the iterative decoding algorithms used for *low-density parity-check* codes. In fact, iterative algorithms of this type have been used, e.g., in [XH07], [Ind08], to design sparse recovery algorithms. However, those algorithms were designed to work only for the case where the signal x is k-sparse or "almost" k-sparse. In contrast, SMP is designed and guaranteed to work for arbitrary input signals x.

Finally, there are interesting connections between SMP and the message-passing-based recovery methods for sparse matrices [SBB06a], [LMP$^+$08]. In particular, the Counter Braids algorithm of [LMP$^+$08] provides provable guarantees for the quality of recovered approximation. However, that scheme works only for non-negative signals x.

II. Preliminaries and formal statement of the result

An essential tool for our constructions are *unbalanced expander graphs*. Consider a bipartite graph $G = (U, V, E)$. We refer to U as the "left" part, and refer to V as the "right" part; a vertex belonging to the left (right) part is called a left (right) vertex. In our constructions the left part will correspond to the set $\{1, 2, \dots, n\}$ of coordinate indexes of vector x, and the right part will correspond to the set of row indexes of the measurement matrix. A bipartite graph is called *left-d-regular* if every vertex in the left part has exactly d neighbors in the right part.

Definition 1: A bipartite, left-d-regular graph $G = (U, V, E)$ is an (s, d, ϵ)-*expander* if any set $S \subset U$ of at most s left vertices has at least $(1 - \epsilon)d|S|$ neighbors.

Since expander graphs are meaningful only when $|V| < d|U|$, some vertices must share neighbors, and hence the parameter ϵ cannot be smaller than $1/d$. Using the probabilistic method one can show that there exist (s, d, ϵ)-expanders with $d = O(\log(|U|/s)/\epsilon)$ and $|V| = O(s \log(|U|/s)/\epsilon^2)$.

For many applications one usually needs an *explicit* expander, i.e., an expander for which we can efficiently compute the neighbor set of a given left vertex. No explicit constructions with the aforementioned (optimal) parameters are known. However, it is known [GUV07] how to explicitly construct expanders with left degree $d = O((\log |U|)(\log s)/\epsilon)^{1+1/\alpha}$ and right set size $(d^2 s^{1+\alpha})$, for any fixed $\alpha > 0$. In the remainder of this paper, we will assume expanders with the optimal parameters.

For a set S of nodes of G, the set of its neighbors in G is denoted by $\Gamma_G(S)$. The subscript G will be omitted when it is clear from the context, and we write $\Gamma(u)$ as a shorthand for $\Gamma(\{u\})$.

For any n-dimensional vector x, and $S \subset \{1 \dots n\}$, we use x_S to denote an $|S|$-dimensional projection of x on coordinates in S.

A. Formal statement of the result

Consider any n-dimensional "signal" vector x that is k-sparse. The sketching matrix A is a $m \times n$ matrix induced by a (s, d, ϵ)-expander $G = (\{1 \dots n\}, \{1 \dots m\}, E)$, for $s = O(k)$. Let μ be the m-dimensional "noise" vector, and let $b = Ax + \mu$ be the "noisy measurement" vector. Also, denote $\eta = \|\mu\|_1/d$.

For technical reasons we assume that n is divisible by four. Moreover, we set ϵ to be a "sufficiently small" constant.

1) Let $j = 0$
2) Let $x^j = 0$
3) Repeat T times
   a) Let $j = j + 1$
   b) Let $c = b - Ax^{j-1}$
      Note: $c = A(x - x^{j-1}) + \mu$
   c) Let $u^* = u^*(c)$ be such that for any $i = 1 \dots n$
      $$u^*(c)_i = \text{median}(c_{\Gamma(i)})$$
   d) Let $u^j = H_{2k}[u^*]$
      Note: from Lemma 3 we have
      $$\|u^j - (x - x^{j-1})\|_1 \leq \|x - x^{j-1}\|/4 + C\eta$$
   e) Let $x^j = x^{j-1} + u^j$
      Note: $\|x - x^j\|_1 \leq \|x - x^{j-1}\|/4 + C\eta$
   f) Let $x^j = H_k[x^j]$
      Note: from Lemma 6 we have
      $$\|x - x^j\|_1 \leq \|x - x^{j-1}\|/2 + 2C\eta$$

Fig. 2. Sparse Matching Pursuit algorithm

Theorem 1: There exists an algorithm that, for any k-sparse signal x and noise vector μ,given $b = Ax+\mu$, recovers x^* such that $\|x - x^*\|_1 = O(\|\mu\|_1/d)$. The algorithm runs in time $O(nd\log(d\|x\|_1/\|\mu\|_1))$.

Let $x^{(k)}$ be the best k-sparse approximation of x, i.e., $x^{(k)} = \text{argmin}_{k-sparse\ x'}\|x-x'\|_1$. Since $Ax+\mu = Ax^{(k)} + [\mu + A(x - x^{(k)})]$, and $\|A(x - x^{(k)})\|_1 \leq d\|x - x^{(k)}\|_1$ for any x, Theorem 1 immediately implies the following more general statement.

Corollary 2: For any parameter k, any vector x and noise vector μ, given $b = Ax + \mu$, the algorithm recovers x^* such that $\|x - x^*\|_1 = O(\|\mu\|_1/d + \|x - x^{(k)}\|_1)$.

III. The Algorithm

In this section we describe the *Sparse Matching Pursuit* algorithm. We also present an overview of the algorithm analysis.

The algorithm is as follows. Recall that we assume that the signal x is k-sparse, and $b = Ax + \mu$. The algorithm will compute approximations $x^0, x^1, \dots, x^T$. Let $H_l[y]$ be a "thresholding operator", which zeros out all but l largest in magnitude coefficients of the argument y.

From the comments in the algorithm description (assuming Lemmas 3 and 6) we conclude that, for ϵ small enough and for any $j = 1, 2, \dots T$, we have

$$\|x^j - x\|_1 \leq \|x\|_1/2^j + 4C\eta$$

Thus, setting the number of iterations to $T = \log(\|x\|_1/\eta)$ guarantees that

$$\|x^T - x\|_1 = O(\eta)$$

A. Main Lemmas

Lemma 3: For any $2k$-sparse u, let $c = Au + \mu$ and $\eta = \|\mu\|_1/d$. Then

$$\|H_{2k}[u^*(c)] - u\|_1 \leq O(\epsilon)\|u\|_1 + O(\eta)$$

The above lemma is in fact a simplification of Theorem 5 in [IR08]. However, in this paper we will present a self-contained proof, which is very different from the combinatorial argument used in [IR08].

Proof: For the purpose of analysis, we assume that $|u_1| \geq \dots \geq |u_n|$. Note that since u is $2k$-sparse, we have $u_{2k+1} = \dots = u_n = 0$. Let $S = \{1 \dots 2k\}$. The proof of Lemma 3 relies on the following two lemmas, whose proof is deferred to the next section.

Lemma 4:

$$\|(u^* - u)_S\|_1 = O(\epsilon)\|u\|_1 + O(\eta)$$

Lemma 5: Let $B \subset \overline{S}$ be a set of coordinates of size at most $2k$. Then

$$\|u^*_B - u_B\|_1 = \|u^*_B\|_1 = O(\epsilon)\|u\|_1 + O(\eta)$$

Let T be the coordinates of the $2k$ largest in magnitude coefficients of u^*. Then:

$$\begin{aligned}
& \|H_{2k}[u^*] - u\|_1 \\
= \ & \|u^*_T - u\|_1 \\
= \ & \|(u^* - u)_{S\cap T}\|_1 + \|u^*_{T-S}\|_1 + \|u_{S-T}\|_1 \\
\leq \ & \|(u^* - u)_{S\cap T}\|_1 + \|u^*_{T-S}\|_1 + \\
& \|u^*_{S-T}\|_1 + \|(u^* - u)_{S-T}\|_1 \\
= \ & \|(u^* - u)_S\|_1 + \|u^*_{T-S}\|_1 + \|u^*_{S-T}\|_1
\end{aligned}$$

To bound $\|u^*_{S-T}\|_1$, observe that for any $i \in S - T$ and $i' \in T - S$, we have $|u^*_i| \leq |u^*_{i'}|$. Since $|T - S| = |S - T|$, it follows that $\|u^*_{S-T}\|_1 \leq \|u^*_{T-S}\|_1$. Hence, for $B = T - S$, it follows from Lemmas 4 and 5 that

$$\|H_{2k}[u^*]-u\|_1 \leq \|(u^*-u)_S\|_1+2\|u^*_B\|_1 = O(\epsilon)\|u\|_1+O(\eta)$$

■

Lemma 6: For any k-sparse vector x, and any vector x' we have

$$\|H_k[x'] - x\|_1 \leq 2\|x' - x\|_1$$

Proof: Let S be a support of x, and let T be the largest k coefficients of x'. Observe that from the definition of T and the fact that $|S - T| = |T - S|$, it follows that $\|x'_{S-T}\|_1 \leq \|x'_{T-S}\|_1$. We have

$$\begin{aligned}
& \|x'_T - x\|_1 \\
= \ & \|x_{S-T}\|_1 + \|x'_{T-S}\|_1 + \|x_{S\cap T} - x'_{S\cap T}\|_1 \\
\leq \ & (\|x'_{S-T}\|_1 + \|x'_{S-T} - x_{S-T}\|_1) + \\
& \|x'_{T-S}\|_1 + \|x_{S\cap T} - x'_{S\cap T}\|_1 \\
\leq \ & \|x'_{T-S}\|_1 + \|x'_{S-T} - x_{S-T}\|_1 + \\
& \|x'_{T-S}\|_1 + \|x_{S\cap T} - x'_{S\cap T}\|_1 \\
\leq \ & 2\|x - x'\|_1
\end{aligned}$$

■

IV. Proofs of Lemmas 4 and 5

We start from some basic observations.

Decomposition. For the analysis, it is convenient to decompose the vector Au into a sum $v + v$". The vector v is such that $v_j = u_{i(j)}$ where $i(j)$ is the index i' from $\Gamma(j)$ with the largest value of $|u_{i'}|$.

Fact 1: Let $v" = Au - v$. Then $\|v"\|_1 \leq 2\epsilon d\|u\|_1$.

The proof of this fact is virtually identical to the proof of the RIP1 property of expander matrices described in [BGI+08]. The reader is referred to that paper for the details.

It follows that the vector $c = Au + \mu$ can be represented as $c = v + v'$, where $\|v'\|_1 \leq O(\epsilon)d\|u\|_1 + \|\mu\|_1$.

Composing quantiles. In the proof we will compute bounds on the medians of vectors $|c_B|$, where B are subsets of coordinates. In light of the above decomposition, it would make sense to compute separate bounds for medians of $|v_B|$ and $|v'_B|$, and then combine them. Unfortunately, it is in general not true that median$(u+v) \leq$ median$(u)+$median(v), even for positive vectors u, v. Fortunately, we can overcome this problem by using lower quantiles. Specifically, for any non-negative n-dimensional vector u, and any $\alpha \in (0,1)$ such that αn is an integer, we define quant$_\alpha(u)$ to be the αn-th largest element of u. Then we have the following fact.

Fact 2: For any vectors $u, v \geq 0$ of dimension n divisible by 4, we have

$$\text{quant}_{1/2}(u+v) \leq \text{quant}_{1/4}(u) + \text{quant}_{1/4}(v)$$

Proof: Let $U = \text{quant}_{1/4}(u)$ and $V = \text{quant}_{1/4}(v)$. There are at most $n/4-1$ elements of u (v, resp.) that are greater than U(V, resp.). Therefore, there are at least $n - 2(n/4 - 1) = n/2 + 2$ elements of $u + v$ that are not greater than $U + V$, which concludes the proof. ∎

Telescoping trick. Consider two sequences: $a = a_1, \ldots, a_s$ and $b = b_1, \ldots b_t$ of positive numbers, such that $\{a_1, \ldots, a_s\} \subset \{b_1, \ldots b_t\}$. In addition, we assume $b_1 \geq b_2 \geq \ldots \geq b_t = 0$. The two sequences are related in the following way. For each b_i, define $c(i)$ to be the cardinality of the set $C(i) = \{j : a_j \geq b_i\}$, i.e., the number of times an element from $\{b_1 \ldots b_i\}$ appears in the sequence a; we assume $c(0) = 0$. The following claim states that if this number is bounded, then the sum of the elements of a is only a fraction of the sum of elements in b.

Claim 7: Assume that $c(i) \leq \alpha i$ for some $\alpha > 0$. Then $\|a\|_1 \leq \alpha\|b\|_1$.

Proof: For simplicity of exposition, we assume that all terms in b are distinct. First, observe that for each i, the number of times the value b_i occurs in a is equal to $c(i) - c(i-1)$. Thus

$$\begin{aligned}
\sum_j a_j &= \sum_{i=1}^{t}[c(i) - c(i-1)]b_i \\
&\leq \sum_{i=1}^{t} c(i)b_i - \sum_{i=2}^{t} c(i-1)b_{i-1} + \\
&\quad \sum_{i=2}^{t} c(i-1)(b_{i-1} - b_i) \\
&\leq \alpha t b_t + \sum_{i=2}^{t} \alpha(i-1)(b_{i-1} - b_i) \\
&\leq \sum_{i=2}^{t} \alpha(b_{i-1} - b_t) \\
&\leq \alpha\|b\|_1
\end{aligned}$$

∎

A. Proof of Lemma 4

Recall that $u^*_i = \text{median}(v_{\Gamma(i)} + v'_{\Gamma(i)})$. Therefore, we need to bound

$$\begin{aligned}
\|(u^* - u)_S\|_1 &= \sum_{i \in S} |\text{median}(v_{\Gamma(i)} + v'_{\Gamma(i)}) - u_i| \\
&= \sum_{i \in S} |\text{median}(v_{\Gamma(i)} - u^d_i + v'_{\Gamma(i)})| \\
&\leq \sum_{i \in S} \text{median}(|v_{\Gamma(i)} - u^d_i| + |v'_{\Gamma(i)}|)
\end{aligned}$$

where u^d_i is a vector of dimension d containing as coordinates d copies of u_i. For any $i \in S$, let $w^i = v_{\Gamma(i)} - u^d_i$. Then, it suffices to bound

$$\begin{aligned}
\sum_{i \in S} \text{median}(|w^i| + |v'_{\Gamma(i)}|) &\leq \sum_{i \in S} \text{quant}_{1/4}(|w^i|) + \\
&\quad \sum_{i \in S} \text{quant}_{1/4}(|v'_{\Gamma(i)}|)
\end{aligned}$$

We bound the second term using the following claim.

Claim 8: Let P be any set of at most s coordinates. Then

$$\sum_{i \in P} \text{quant}_{1/4}(|v'_{\Gamma(i)}|) = O(\|v'\|_1/d)$$

Proof: For each v'_j, let $c(j)$ be the number of $i \in P$ having at least $d/4$ neighbors in the set $\{1 \ldots j\}$. From the expansion properties of the graph G it follows that $c(j)(d/4 - \epsilon d) \leq j$. Thus $c(j) \leq 8j/d$. Applying Claim 7 finishes the proof. ∎

Therefore, the contribution of the second term is at most $O(\|v'\|_1/d) = O(\epsilon\|u\|_1) + O(\eta)$.

To bound the first term, we proceed as follows. Recall that we assumed that the entries $|u_1|, |u_2|, \ldots$ appear in the non-increasing order. Partition S into the union of S^+ and $S^- = S - S^+$, where $S^+ = \{i \in S : |\Gamma(i) \cap \Gamma(\{1 \ldots i -$

$1\})| < d/4\}$. We will bound the first term separately for elements in S^+ and S^-.

Observe that for each $i \in S^+$ the number of non-zero elements in w^i is smaller than $d/4$. Therefore, $\sum_{i \in S^+} \text{quant}_{1/4}(|w^i|) = 0$.

To take care of the elements in S^- we need to bound

$$\sum_{i \in S^-} \text{quant}_{1/4}(|v_{\Gamma(i)} - u_i^d|) \leq \sum_{i \in S^-} \text{quant}_{1/4}(|v_{\Gamma(i)}|) \quad (1)$$

For any $r \in S$, consider the set S_r containing indices $i \in S^-$ such that $|\Gamma(i) \cap \Gamma(\{1 \dots r\})| \geq d/4$. Our goal is to show that $|S_r|$ is relatively small, and therefore few elements of the sum in equation 1 can be large.

We partition S_r into $S_r^< = S_r \cap \{1 \dots r\}$ and $S_r^> = S_r - S_r^<$. From the expansion properties of G it follows that

$$d(1-\epsilon)(r + |S_r^>|) \leq dr + 3/4 \cdot d|S_r^>|$$

Therefore we have $|S_r^>| \leq \frac{\epsilon}{1/4-\epsilon} r \leq 8\epsilon r$.

To bound $S_r^<$, we observe that from the definition of S^- and the expansion of G it follows that

$$d(1-\epsilon)r \leq d(r - |S_r^<|) + 3/4 \cdot d|S_r^<|$$

Therefore we have $|S_r^>| \leq 4\epsilon r$ and $|S_r| = |S_r^>| + |S_r^<| \leq 12\epsilon r$. That is, for any r, at most $12\epsilon r$ terms in the sum in equation 1 can be greater than $|u_r|$. From Claim 7 it follows that the total value of the sum is at most $O(\epsilon)\|u\|_1$.

Putting all terms together concludes the proof of Lemma 4.

B. Proof of Lemma 5

We need to bound

$$\begin{aligned} \|(u^*)_B\|_1 &= \sum_{i \in B} |\text{median}(v_{\Gamma(i)} + v'_{\Gamma(i)})| \\ &\leq \sum_{i \in B} |\text{quant}_{1/4}(v_{\Gamma(i)})| + \sum_{i \in B} |\text{quant}_{1/4}(v'_{\Gamma(i)})| \end{aligned}$$

Using Claim 8 we can bound the second term by $O(\epsilon)\|u\|_1 + O(\eta)$. To bound the first term we proceed as follows. For each $r \in S$ we define $B_r = \{i \in B : |\Gamma(i) \cap \Gamma(\{1 \dots r\})| \geq d/4\}$. From expansion of the graph G and the fact that $B \cap S = \emptyset$ it follows that

$$(1-\epsilon)d(r + |B_r|) \leq dr + 3/4 \cdot d|B_r|$$

It follows that $|B_r| \leq 8\epsilon r$. From Claim 7 we conclude that the first term is bounded by $O(\epsilon)\|u\|_1$.

V. Experiments

To estimate the empirical sketch sizes that our SMP algorithm requires, we perform experiments in which we test exact recovery of sparse vectors (similarly to [BI08]). An experiment is performed as follows: a vector is generated consisting of k non-zero "peaks" which are randomly either $+1$ or -1; the sketch is computed by applying a given measurement matrix; a recovery of the vector from the sketch is attempted and the experiment is deemed successful if the recovered vector is identical to the original vector. For given parameters of signal length n, sketch length m, and signal sparsity k, we generate a single measurement matrix with a given sparsity and perform a number of separate experiments with the same matrix. The fraction of the experiments that are successful is indicative of the probability that a vector is correctly recovered using a "typical" matrix of the given parameters. To visualize the sketch sizes, we fix the signal length n and vary m,k along a two-dimensional grid.

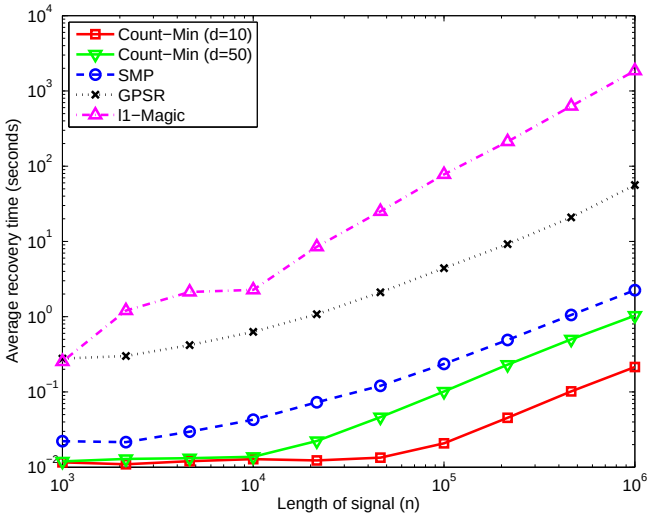

Fig. 3. Comparisons of recovery times. We used the following parameters: the signal sparsity $k = 0.002n$, the number of measurements $m = 0.1n$, and the matrix sparsity parameter $d = 10$ (except for Count-Min for which we also use the higher value of $d = 50$). The SMP algorithm is run for 10 iterations.

Our results are shown in figure 4. We compare our algorithm with three other recovery methods capable of handling sparse matrices. The first one is the Count-Min algorithm [CM04][4]. The other two algorithms rely on linear-programming-based recovery. Specifically, we use two linear optimization algorithms: ℓ_1-magic with the `l1eq_pd` program [CR05] and GPSR [FNW07][5]. See [BI08], [BGI+08] for more details on linear-programming-based recovery using sparse sketching matrices.

We also measured the running time of our recovery algorithms; the results are shown in figure 3. The algorithms were implemented in Matlab, with critical subroutines (sparse matrix multiplication, median recovery) written in C and compiled as a Matlab Executable. The code is avail-

[4]Since we are dealing with signals that can have negative coordinates, we are using the "median" version of the algorithm, see the original paper for details.

[5]The GPSR algorithm requires a precision parameter τ, which impacts the quality of recovery as well as the running time of the algorithm. In our experiments we used $\tau = 10^{-3}\|\mathbf{A}^T\mathbf{y}\|_\infty$. Higher values of τ resulted in slightly lower running times, but at the expense of reducing the quality of recovery, which increased the number of measurements needed to achieve a given pobability of success.

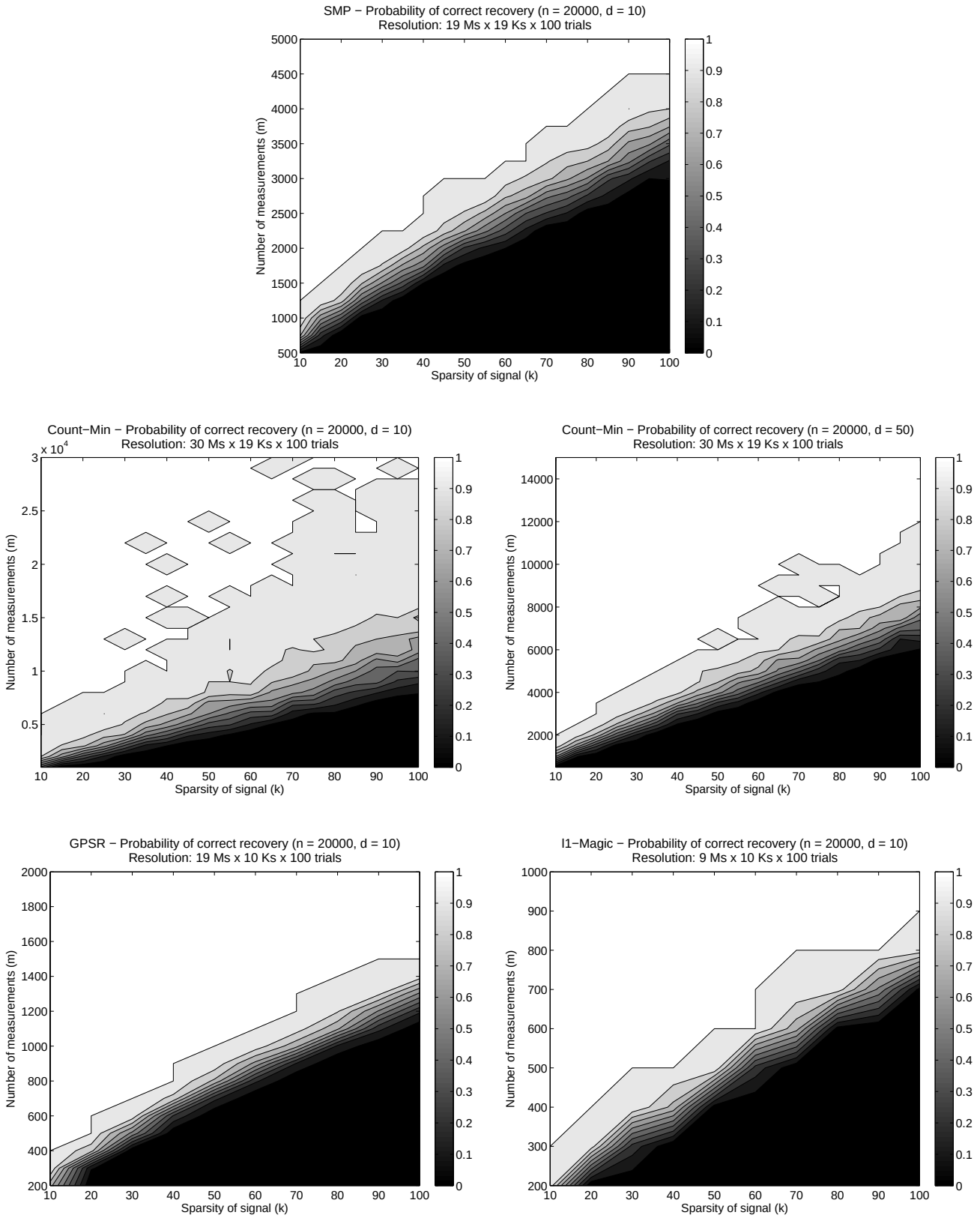

Fig. 4. Results of exact recovery experiments for SMP (top), Count-Min (middle row), GPSR (bottom-left) and ℓ_1-MAGIC (bottom-right). All plots are for the same signal length $n = 20000$ and the same sparsity range $k \in [10, 100]$. The resolution of each plot is shown: the number of vertical axis divisions ("Ms"), number of horizontal axis divisions ("Ks") and number of trials per point. The matrix sparsity was set to $d = 10$ for all algorithms except Count-Min for which $d = 50$ was also used. SMP uses 10 iterations and attempts to recover a $2k$-sparse vector.

able at http://groups.csail.mit.edu/toc/sparse/wiki/index.php?title=Sparse_Recovery_Experiments .

A. Discussion

All the aforementioned methods use the same type of measurement matrix (sparse and binary). Any of the methods can be used to recover a signal x from a given sketch Ax. Thus, the choice of the recovery method depends on the amount of available computational resources. As per figures 3 and 4, using SMP instead of linear-programming methods can reduce the running time by up to a few orders of magnitude. At the same time, the sketch length for a given sparsity k is increased only by small factor (between 3 and 5, for $n = 20,000$).

Both SMP and Count-Min have low (near-linear) running times. For fixed matrix sparsity parameter $d = 10$, SMP runs roughly 10 times longer than Count-Min (which is not surprising, since each of the 10 iterations of SMP utilizes a Count-Min-like voting process). However, Count-Min requires higher values of d to achieve good recovery performance, which brings its running time closer to that of SMP (and also increases the encoding and update time of the algorithm). In that case, Count-Min requires sketches twice the length of those required by SMP.

ACKNOWLEDGEMENTS

The authors would like to thank Wei Dai, Anna Gilbert, Olgica Milenkovic, Martin Strauss and Joel Tropp for many useful discussions.

REFERENCES

[BGI+08] R. Berinde, A. Gilbert, P. Indyk, H. Karloff, and M. Strauss. Combining geometry and combinatorics: a unified approach to sparse signal recovery. *Allerton*, 2008.

[BI08] R. Berinde and P. Indyk. Sparse recovery using sparse random matrices. *MIT-CSAIL Technical Report*, 2008.

[BM08] T. Blumensath and M.E.Davies. Iterative hard thresholding for compressed sensing. *Preprint, available at arXiv:0805.0510v1*, 2008.

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

[CDD06] A. Cohen, W. Dahmen, and R. DeVore. Compressed sensing and best k-term approximation. *Preprint*, 2006.

[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. In *Proc. 40th Ann. Conf. Information Sciences and Systems*, Princeton, Mar. 2006.

[CR05] E. J. Candès and J. Romberg. *ℓ_1-MAGIC: Recovery of Sparse Signals via Convex Programming*, 2005. Available at http://www.acm.caltech.edu/l1magic .

[CRT06a] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Inf. Theory*, 52(2):489–509, 2006.

[CRT06b] 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.

[DDT+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.

[DM08] W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing: Closing the gap between performance and complexity. *Arxiv:0803.0811*, 2008.

[DMT07] C. Dwork, F. McSherry, and K. Talwar. The price of privacy and the limits of LP decoding. *STOC*, 2007.

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

[FNW07] M.A.T. Figueiredo, R.D Nowak, and S.J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEE Journal of Selected Topics in Signal Processing*, 1, 2007.

[GGI+02] A. C. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. Fast, small-space algorithms for approximate histogram maintenance. In *ACM Symposium on Theoretical Computer Science*, 2002.

[GLR08] V. Guruswami, J. Lee, and A. Razborov. Almost euclidean subspaces of l_1 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. In *ACM STOC 2007*, pages 237–246, 2007.

[GUV07] V. Guruswami, C. Umans, and S. P. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. In *IEEE Conference on Computational Complexity (CCC 2007)*, pages 96–108, 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.

[Ind08] P. Indyk. Explicit constructions for compressed sensing of sparse signals. *SODA*, 2008.

[IR08] P. Indyk and M. Ruzic. Near-optimal sparse recovery in the l_1 norm. *FOCS*, 2008.

[LMP+08] Y. Lu, A. Montanari, B. Prabhakar, S. Dharmapurikar, and A. Kabbani. Counter braids: A novel counter architecture for per-flow measurement. *SIGMETRICS*, 2008.

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

[NT08] D. Needell and J. A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Appl. Comp. Harmonic Anal.*, 2008. To appear.

[RV06] M. Rudelson and R. Veshynin. Sparse reconstruction by convex relaxation: Fourier and Gaussian measurements. In *Proc. 40th Ann. Conf. Information Sciences and Systems*, Princeton, Mar. 2006.

[SBB06a] S. Sarvotham, D. Baron, and R. G. Baraniuk. Compressed sensing reconstruction via belief propagation. *Technical Report ECE-0601, Electrical and Computer Engineering Department, Rice University*, 2006.

[SBB06b] S. Sarvotham, D. Baron, and R. G. Baraniuk. Sudocodes - fast measurement and reconstruction of sparse signals. *IEEE International Symposium on Information Theory*, 2006.

[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. In *Proc. IS&T/SPIE Symposium on Electronic Imaging*, 2006.

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