Near-Optimal Sparse Recovery in the L_1 norm

Piotr Indyk
MIT
indyk@mit.edu

Milan Ružić
ITU Copenhagen
milan@itu.dk

Abstract

We consider the *approximate sparse recovery problem*, where the goal is to (approximately) recover a high-dimensional vector $x \in \mathbb{R}^n$ from its lower-dimensional *sketch* $Ax \in \mathbb{R}^m$. 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.

In this paper we provide a sparse recovery scheme which achieves close to optimal performance on virtually all attributes (see Figure 1). In particular, this is the first scheme that guarantees $O(k \log(n/k))$ sketch length, and near-linear $O(n \log(n/k))$ recovery time *simultaneously*. It also features low encoding and update times, and is noise-resilient.

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 $m \times n$ 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 very convenient 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 has been 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 was 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. In the same figure we also show the best known bound for each of the aforementioned attributes of the algorithms (see the caption for further explanation).

Our result. The main result of this paper is a very efficient sparse recovery scheme, with parameters as depicted in the last row of Figure 1. Up to the leading constants, the scheme achieves the best known bounds for: the error probability (our scheme is deterministic), sketch length, encoding and update times. Its decoding time is in general incomparable to the best prior bounds; however, it provides an improvement for k large enough. Finally, our scheme is resilient to noise (see Theorem 10 for the exact guarantee). The only drawback of our 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 earlier schemes (although given that our scheme can be instantiated with $C = 1 + \epsilon$ for any $\epsilon > 0$, our guarantees are technically incomparable to those of [DM08, NT08]).

The efficiency with respect to many attributes makes our scheme an attractive option for sparse recovery problems. In particular, this is the first scheme that guarantees the $O(k \log(n/k))$ sketch length, and the near-linear decoding time *simultaneously*. Both attributes are of key importance: the sketch length determines the compression ratio (so any extra $\log n$ factor can reduce that ratio tenfold), while running times of $\Omega(nk)$ can quickly become prohibitive for n large enough (say, $n = 1000,000$).

Our techniques. We use an adjacency matrix of an *unbalanced expander* as the encoding matrix A. Since such matrices can be very sparse (with only $O(\log(n/k))$ ones per column), the resulting scheme has very efficient encoding and update times. To make the scheme fully efficient, we also design a very fast recovery procedure that we call *Expander Matching Pursuit (EMP)* (Figure 2). The procedure roughly resembles the "greedy iterative approach" (a.k.a. *Orthogonal Matching Pursuit* [Tro04, TG05, NV07, DM08, NT08]), where the idea is to iteratively identify and eliminate "large" coefficients. However, in our procedure, the "large" coefficients are identified only at the beginning (in Step 1). In the remainder of the procedure, the choice of coordinates to iterate on is based on the structure of the expander matrix A, rather than the estimated magnitudes of coefficients.

We remark that the use of adjacency matrices of sparse random or expander graphs as encoding matrices for sparse approximation problems is not new; see, e.g., [CCFC02, CM06, CM04, GSTV06, GSTV07, XH07, Ind08, GLR08, BGI+08] for related work. However, all previous schemes were sub-optimal is some respects. In particular, the schemes of [CCFC02, CM06, CM04] provided only randomized guarantees and slightly worse measurement bounds; the sublinear-time algorithms of [GSTV06, GSTV07, Ind08] incurred polylogarithmic overhead in the number of measurements; the result of [XH07] was shown only for vectors x that are themselves k-sparse (or are slight

[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.

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]	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
Best bounds per each column	D	$k \log(n/k)$	$n \log(n/k)$	$\log(n/k)$	$\min[k \log^d n,$ $n \log n]$	$\ell_2 \leq \frac{\epsilon}{k^{1/2}}\ell_1$	Y
This paper	D	$k \log(n/k)$	$n \log(n/k)$	$\log(n/k)$	$n \log(n/k)$	$\ell_1 \leq (1+\epsilon)\ell_1$	Y

Figure 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, such as explicitness or universality of the measurement matrices. 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, 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 $Ax + \nu$. 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 $\|x\|_2$ if the coordinates of x 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 $\Omega(n)$.

generalizations of thereof); the matrices employed in [GLR08] had only sublinear, not logarithmic sparsity; and the decoding algorithm of [BGI+08] required solving a linear program, resulting in $\Omega(n^{3/2})$ running time.

Practicality of the algorithm and further developments. We have implemented a version of the EMP algorithm. As expected, the algorithm runs very fast. However, the number of measurements required by the algorithm to achieve correct recovery is somewhat suboptimal. In particular, we performed recovery experiments on random signed k-sparse signals of length n. For $k = 50$ and $n = 20000$, one typically needs about 5000 measurements to recover the signal. In comparison, the linear-programming-based recovery algorithm [BGI+08] requires only about 450 measurements to perform the same task[2].

Very recently, based on the ideas from this paper as well as from [BGI+08, NT08], we have developed an improved algorithm for the sparse recovery problem [IR08]. The running time of the new algorithm, called *Sparse Matching Pursuit*, or *SMP*, is slightly higher (by a logarithmic factor) than of EMP, and has the same asymptotic bound on the number of required measurements. However, empirically, the algorithm performs successful recovery from a smaller number of measurements. In particular, for the instances described earlier, SMP typically needs about 1800 measurements. See [IR08] for further empirical evaluation.

2 Preliminaries about expander graphs

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 $\{1, 2, \ldots, n\}$ 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 $(1 - \varepsilon)d|S|$ 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 $d = O(\log(|U|/s)/\varepsilon)$ and $|V| = O(s \log(|U|/s)/\varepsilon^2)$.

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 $d = O((\log |U|)(\log s)/\varepsilon)^{1+1/\alpha}$ and right set size $O(d^2 s^{1+\alpha})$, for any fixed $\alpha > 0$. In the remainder of this paper, we will assume expanders with the optimal parameters.

The set of neighbors of a set $S \subset U$ is denoted by

$$\Gamma_G(S) = \{v \in V \mid (\exists u \in S)\ (u, v) \in E\}\ .$$

The subscript G will be omitted when it is understood, and we write $\Gamma(u)$ as a shorthand for $\Gamma(\{u\})$. Recall that the unique existential quantifier is denoted by $\exists!$ – it can be read as "there

[2]For both algorithms we used randomly generated graphs with left degree equal to 20.

exists a unique". Let

$$\Phi_G(S) = \{v \in V \mid (\exists! u \in S)\ (u,v) \in E\}\ .$$

We call the elements of $\Phi_G(S)$ *unique neighbor* nodes.

We will make use of the well-known fact that high expansion implies that any small set has many unique neighbors.

Lemma 1. *For any $S \subset U$ such that $|S| \leq s$, we have $|\Phi_G(S)| \geq (1-2\varepsilon)d|S|$.*

Lemma 2. *Let $S' = \{u \in S\ :\ |\Gamma(u) \cap \Phi(S)| \geq (1-\lambda)d\}$, for a given $\lambda \geq 2\varepsilon$. Then $|S'| > (1-\frac{2\varepsilon}{\lambda})|S|$.*

Proof. Let $m = |S|$, and suppose that $|S'| = m - k$. Let k^* be the largest integer that satisfies $k^*(1-\lambda)d + (m-k^*)d > |\Phi(S)|$. It is easy to see that k^* is never smaller than $k = m - |S'|$. Using Lemma 1 to lower bound $|\Phi(S)|$ gives $k^* < \frac{2\varepsilon}{\lambda}m$. □

In the special case when $\lambda = 2\varepsilon$ the set S' contains at least one element.

3 Sparse recovery

We consider linear sketches of form Ax, where A is an $m \times n$ adjacency matrix of a (s, d, ε)-expander G, where $s = 4k$, $d = O(\frac{1}{\varepsilon}\log\frac{n}{k})$, $m = O(\frac{k}{\varepsilon^2}\log\frac{n}{k})$ and $\varepsilon < 1/16$. We will consider the general case of noisy measurements. Let $c = Ax + \nu$ be a sketch vector contaminated with noise ν.

We use K to denote the set of k indexes of coordinates of x having k largest magnitudes (ties are broken arbitrarily). In other words, x_K is a best k-term approximation to x. For a coordinate subset I we will write x_I to denote the vector obtained from x by zeroing-out all entries outside of I.

3.1 Algorithm

The outline of the recovery algorithm is given in Figure 2. Note that the nonzero components of the approximation $\hat{x}$ are confined to the set $\bar{I}$. The set $\bar{I}$ can be computed as as follows. First, observe that the set $\bar{I}$ is uniquely defined. The following claim establishes that $|\bar{I}| < 4k$.

Claim 3. *Let γ be a constant value larger than ε. Suppose that $I \subset \{1, 2, \ldots, n\}$ is a given set of coordinate positions, with $|I| \leq (1-\frac{\varepsilon}{\gamma})s$. Let J be the smallest-size superset of I with the property that*

$$(\forall j \in \{1, 2, \ldots, n\} \setminus J)\ |\Gamma(j) \cap \Gamma(I)| \leq \gamma d\ . \tag{3}$$

The size of the set J is smaller than $(1-\frac{\varepsilon}{\gamma})^{-1}|I|$.

Proof. Because J is the smallest-size superset of I satisfying (3), it follows that there exists a sequence of indexes $(j_1, \ldots, j_l)$, with $l = |J| - |I|$, such that $\{j_1, \ldots, j_l\} = J \setminus I$ and

$$|\Gamma(j_{k+1}) \cap \Gamma(I \cup \{j_1, \ldots, j_k\})| > \gamma d\ .$$

Therefore, we have that $d|J| - |\Gamma(J)| > \gamma dl$, and then $d(|J| - \gamma l) > |\Gamma(J)| \geq (1-\varepsilon)d|J|$. We see that $\gamma l < \varepsilon|J|$, and thus $(\gamma - \varepsilon)|J| < \gamma|I|$. □

Expander Matching Pursuit

1. Compute $x^* = x^*(c)$ such that for any $i = 1 \dots n$:

$$x_i^*(c) = \text{median}(c_{j_1}, c_{j_2}, \dots, c_{j_d}) \ , \tag{1}$$

   where $\{j_1, \dots, j_d\} = \Gamma(i)$;

2. Find the set I consisting of $2k$ indexes of the coordinates of x^* of highest magnitudes, breaking ties arbitrarily;

3. Find the set $\bar{I}$ being the smallest-size superset of I such that

$$(\forall i \in \{1, 2, \dots, n\} \setminus \bar{I}) \ |\Gamma(i) \cap \Gamma(\bar{I})| \leq 2\varepsilon d \ ; \tag{2}$$

   (we provide a more detailed description of this step later in this section)

4. $\hat{x} \leftarrow \mathbf{0}$;

5. $j \leftarrow 1$; $\quad I_j \leftarrow \bar{I}$;

6. Repeat the following sequence of steps until $I_j = \emptyset$:

   (a) Find $I'_j = \{i \in I_j \ : \ |\Gamma(i) \cap \Phi(I_j)| \geq (1 - 2\varepsilon)d\}$;

   (b) $x^* \leftarrow x^*(c)$;

   (c) $\hat{x}_{I'_j} \leftarrow x^*_{I'_j}$;

   (d) $c \leftarrow c - Ax^*_{I'_j}$;

   (e) $I_{j+1} \leftarrow I_j \setminus I'_j$; $\quad j \leftarrow j + 1$;

Figure 2: Recovery algorithm

We now discuss the procedure for finding $\bar{I}$, together with the running time analysis. Initially, we let $\bar{I}$ be equal to I. The algorithm maintains a priority queue over the set $\{1, 2, \ldots, n\} \setminus \bar{I}$, with the priority of element i being $|\Gamma(i) \cap \Gamma(\bar{I})|$. Each vertex from $V \setminus \Gamma(\bar{I})$ will have an associated list of references to the elements of the priority queue that have it as a neighbor. When a vertex from V enters the set $\Gamma(\bar{I})$ the priorities of the corresponding elements are updated. Elements whose priorities become higher than $2\varepsilon d$ are added to the set $\bar{I}$. The priority queue can be implemented as an array of $2\varepsilon d$ linked lists, so that every required operation runs in constant time. The entire process takes a time of $O(nd)$, since we can attribute a unit cost of work to every edge of G. The algorithm uses $O(n)$ words of storage (in addition to the space used to store the matrix A). A more space-efficient (but slightly slower) algorithm is described in the appendix.

Total running time of step 6 of the algorithm is $O(kd)$. The procedure that performs step 6.a uses a similar method with a priority queue, only that here elements get extracted from the set, and priorities are decreasing. This part uses $O(kd)$ words of storage.

In the remainder of this section we will focus on proving the approximation bounds. We start from technical lemmas providing guarantees for the initial approximation vector x^*. Then we give the proof of the approximation guarantee.

3.2 Technical lemmas

The statement of the following lemma may look somewhat unintuitive, due to the fact that its formulation needs to allow a proof by induction. More intuitive error bounds for estimates x_i^* will be presented in Theorem 5.

Lemma 4. *Let $I, J, L, M \subset \{1, 2, \ldots, n\}$ be given sets of coordinate positions such that*

- $|I| \leq s/2$;
- *I, J, L, M are mutually disjoint, and $I \cup J \cup L \cup M = \{1, 2, \ldots, n\}$;*
- $(\forall l \in L)(|\Gamma(l) \cap \Gamma(I)| \leq 2\varepsilon d)$;
- $(\forall i \in I)(|\Gamma(i) \cap \Gamma(M)| \leq \alpha d)$, *where $\alpha < 1/2 - 2\varepsilon$.*

There exist a chain of subsets of I and a family of disjoint subsets of J, which we respectively write as $I = I_0 \supset I_1 \supset \ldots \supset I_q$ and $\{J_0, J_1, \ldots, J_q\}$ [3]*, satisfying the following two properties:*

$$\begin{aligned} \|x_I - x_I^*\|_1 \quad \leq \quad & (1/2 - 2\varepsilon - \alpha)^{-1} \Big(2\varepsilon \|x_I\|_1 + \frac{1}{d} \|(A\, x_L)_{\Gamma(I)}\|_1 + \\ & + \frac{1}{d} \sum_{k=0}^{q} \|(A\, x_{J_k})_{\Gamma(I_k)}\|_1 + \frac{1}{d} \|\nu_{\Gamma(I)}\|_1 \Big) \ , \end{aligned}$$

and $(\forall j \in J_k)(|\Gamma(j) \cap \Gamma(I_k)| \leq 2\varepsilon d)$, for $0 \leq k \leq q$.

Proof. We will prove the claim by induction on the size of the set I. Suppose first that $|I| = 1$, that is $I = \{i\}$ for some i. In this case we have that $q = 0$ and $J_0 = J$. Consider the multiset $\phi = \{c_{v_1}, c_{v_2}, \ldots, c_{v_l}\}$ such that v_k are the indexes from $\Gamma(i) \setminus M$ (therefore $l \geq (1 - \alpha)d$). The estimate x_i^* will have a rank between $d/2 - (d - l)$ and $d/2$ with respect to the multiset ϕ (the value

[3] Some of the subsets may be empty; thus we cannot call this family a partition of J.

x_i^* need not be equal to one of the elements of ϕ). In other words, at least $(1/2-\alpha)d$ elements of ϕ are not larger than x_i^*, and at at least $(1/2-\alpha)d$ elements of ϕ are not smaller than x_i^*. Therefore in any case ($x_i < x_i^*$ or $x_i > x_i^*$) we have that

$$\begin{aligned}(1/2-\alpha)d\,|x_i - x_i^*| &\leq \|(A\,x_{\{i\}} - c)_{\Gamma(i)\setminus M}\|_1 \leq \|(A\,x_{L\cup J_0} + \nu)_{\Gamma(I)}\|_1 \\ &\leq \|(A\,x_L)_{\Gamma(I)}\|_1 + \|(A\,x_{J_0})_{\Gamma(I)}\|_1 + \|\nu_{\Gamma(I)}\|_1 \ ,\end{aligned}$$

which proves the claim for the case $|I| = 1$.

Now suppose that $|I| > 1$ and that the claim is true for index sets of smaller sizes. Let $J^* = \{j \in J \ : \ |\Gamma(j) \cap \Gamma(I)| > 2\varepsilon d\}$. Because of the expansion property of the graph G and the condition that $|I| \leq s/2$, the size of J^* can be at most $|I|$. Let

$$I' = \{i \in I \ : \ |\Gamma(i) \cap \Phi(I \cup J^*)| \geq (1-2\varepsilon)d\} \ .$$

We will have that for any $i \in I'$ the influence of other entries from $x_{I\cup J^*}$ to the estimate x_i^* is relatively minor. Since all elements from J^* have less than $(1-2\varepsilon)d$ unique neighbor nodes with respect to the set $I \cup J^*$, from Lemma 2 we conclude that $|I'| \geq 1$. Let $I_1 = I \setminus I'$, and apply the induction hypothesis to the set I_1, with I' and $J \setminus J^*$ merged with the set L. Let the returned quasi-partition of J^* be $\{J_0', J_1', \ldots, J_{q-1}'\}$. We make assignments $J_{k+1} = J_k'$, $0 \leq k < q$, and $J_0 = J \setminus J^*$.

By the triangle inequality and the induction hypothesis we have that

$$\begin{aligned}\|x_I - x_I^*\|_1 \ \leq \ & \|x_{I'} - x_{I'}^*\|_1 + (1/2-2\varepsilon-\alpha)^{-1}\Big(2\varepsilon\|x_{I_1}\|_1 + \\ & + \frac{1}{d}\|(A\,x_{I'\cup L\cup J_0})_{\Gamma(I_1)}\|_1 + \frac{1}{d}\sum_{k=1}^{q}\|(A\,x_{J_k})_{\Gamma(I_k)}\|_1 + \frac{1}{d}\|\nu_{\Gamma(I_1)}\|_1\Big) \ .\end{aligned} \tag{4}$$

To upper-bound $\|(A\,x_{I'})_{\Gamma(I_1)}\|_1$ we use the fact that with respect to the graph G each $i \in I'$ has at most $2\varepsilon d$ adjacent nodes shared with the elements of I_1.

$$\|(A\,x_{I'})_{\Gamma(I_1)}\|_1 \leq \sum_{i\in I'}\|(A\,x_{\{i\}})_{\Gamma(I_1)}\|_1 \leq \sum_{i\in I'} 2\varepsilon d|x_i| = 2\varepsilon d\|x_{I'}\|_1 \ .$$

Now we will analyse estimation error for coordinates in I'. For any $i \in I'$ consider the multiset $\phi^{(i)} = \{c_{v_1}, c_{v_2}, \ldots, c_{v_l}\}$ such that v_k are the indexes from $\Gamma(i) \cap \Phi(I \cup J^* \cup M)$ (therefore $l \geq (1-2\varepsilon-\alpha)d$). The estimate x_i^* will have a rank between $d/2 - (d-l)$ and $d/2$ with respect to the multiset $\phi^{(i)}$ (the value x_i^* need not be equal to one of the elements of $\phi^{(i)}$). In other words, at least $(1/2-2\varepsilon-\alpha)d$ elements of $\phi^{(i)}$ are not larger than x_i^*, and at at least $(1/2-2\varepsilon-\alpha)d$ elements of $\phi^{(i)}$ are not smaller than x_i^*. Therefore,

$$\|(A\,x_{\{i\}} - c)_{\Gamma(i)\cap\Phi(I\cup J^*\cup M)}\|_1 \geq (1/2-2\varepsilon-\alpha)d\,|x_i - x_i^*| \ .$$

By aggregating over all $i \in I'$ we get

$$\begin{aligned}(1/2-2\varepsilon-\alpha)d\,\|x_{I'} - x_{I'}^*\|_1 &\leq \|(A\,x_{I'} - c)_{\Gamma(I')\cap\Phi(I\cup J^*\cup M)}\|_1 \leq \\ &\leq \|(A\,x_{L\cup J_0} + \nu)_{\Gamma(I')\cap\Phi(I\cup J^*\cup M)}\|_1 \leq \\ &\leq \|(A\,x_L)_{\Gamma(I')\cap\Phi(I\cup J^*\cup M)}\|_1 + \\ &\quad + \|(A\,x_{J_0})_{\Gamma(I')\cap\Phi(I\cup J^*\cup M)}\|_1 + \\ &\quad + \|\nu_{\Gamma(I')\cap\Phi(I\cup J^*\cup M)}\|_1 \ .\end{aligned}$$

These terms are substituted into (4). Observe that the following four inequalities hold

$$\begin{aligned}
\|(A\,x_L)_{\Gamma(I')\cap\Phi(I\cup J^*\cup M)}\|_1 + \|(A\,x_L)_{\Gamma(I_1)}\|_1 &\leq \|(A\,x_L)_{\Gamma(I)}\|_1 \ , \\
\|(A\,x_{J_0})_{\Gamma(I')\cap\Phi(I\cup J^*\cup M)}\|_1 + \|(A\,x_{J_0})_{\Gamma(I_1)}\|_1 &\leq \|(A\,x_{J_0})_{\Gamma(I)}\|_1 \ , \\
\|\nu_{\Gamma(I')\cap\Phi(I\cup J^*\cup M)}\|_1 + \|\nu_{\Gamma(I_1)}\|_1 &\leq \|\nu_{\Gamma(I)}\|_1 \ , \\
2\varepsilon\|x_{I_1}\|_1 + 2\varepsilon\|x_{I'}\|_1 &\leq 2\varepsilon\|x_I\|_1 \ .
\end{aligned}$$

This proves the claimed bound on $\|x_I - x_I^*\|_1$. □

The following theorem is a simple consequence of Lemma 4.

Theorem 5. *Let $I \subset \{1,2,\ldots,n\}$ be a given set of coordinate positions, with $|I| \leq s/2$. Suppose that $\{J,M\}$ is a partition of the set of remaining coordinates $\{1,2,\ldots,n\} \setminus I$ where the set M satisfies*

$$(\forall i \in I)(|\Gamma(i) \cap \Gamma(M)| \leq \alpha d) \ ,$$

with $\alpha < 1/2 - 2\varepsilon$. It holds that

$$\|x_I - x_I^*\|_1 \leq (1/2 - 2\varepsilon - \alpha)^{-1}(2\varepsilon\|x_{I\cup J}\|_1 + \tfrac{1}{d}\|\nu_{\Gamma(I)}\|_1) \ .$$

We will now state a lemma for a special case when it is known that all elements of I have "many" unique neighbor nodes, and the remaining elements have a well determined structure of intersections with the neighbors of I.

Lemma 6. *Let $\{I,J,L,M\}$ be a given partition of $\{1,2,\ldots,n\}$ with the following properties:*

- *$|I| \leq s/2$ and $(\forall i \in I)(|\Gamma(i) \cap \Phi(I \cup M)| \geq (1-\alpha)d)$, where $\alpha < 1/2$;*
- *$(\forall l \in L)(|\Gamma(l) \cap \Gamma(I)| \leq \beta d)$;*
- *Let $\Delta = \frac{1}{d}\|(A\,x_J + \nu)_{\Gamma(I)\cap\Phi(I\cup M)}\|_1$.*

It holds that $\|x_I - x_I^*\|_1 \leq (1/2-\alpha)^{-1}(\beta\|x_L\|_1 + \Delta)$.

Proof. The proof is a simpler version of the previous proofs. For any $i \in I$ consider the multiset $\phi^{(i)} = \{c_{v_1}, c_{v_2}, \ldots, c_{v_l}\}$ such that v_k are the indexes from $\Gamma(i) \cap \Phi(I \cup M)$ (therefore $l \geq (1-\alpha)d$). The estimate x_i^* will have a rank between $d/2-(d-l)$ and $d/2$ with respect to the multiset $\phi^{(i)}$. In other words, at least $(1/2-\alpha)d$ elements of $\phi^{(i)}$ are not larger than x_i^*, and at at least $(1/2-\alpha)d$ elements of $\phi^{(i)}$ are not smaller than x_i^*. Therefore in any case ($x_i < x_i^*$ or $x_i > x_i^*$) we have that

$$\|(A\,x_{\{i\}} - c)_{\Gamma(i)\cap\Phi(I\cup M)}\|_1 \geq (1/2-\alpha)d\,|x_i - x_i^*| \ .$$

By aggregating over all $i \in I'$ we get that

$$\begin{aligned}
(1/2-\alpha)d\,\|x_I - x_I^*\|_1 &\leq \|(A\,x_I - c)_{\Gamma(I)\cap\Phi(I\cup M)}\|_1 \leq \\
&\leq \|(A\,x_{L\cup J} + \nu)_{\Gamma(I)\cap\Phi(I\cup M)}\|_1 \leq \\
&\leq \|(A\,x_L)_{\Gamma(I)}\|_1 + \|(A\,x_J + \nu)_{\Gamma(I)\cap\Phi(I\cup M)}\|_1 \ .
\end{aligned}$$

Similarly as earlier we observe that

$$\|(A\,x_L)_{\Gamma(I)}\|_1 \leq \sum_{l\in L}\|(A\,x_{\{l\}})_{\Gamma(I)}\|_1 = \sum_{l\in L}|\Gamma(l)\cap\Gamma(I)|\cdot|x_l| \leq \sum_{l\in L}\beta d|x_l| = \beta d\|x_L\|_1 \ .$$

The claimed bound on $\|x_I - x_I^*\|_1$ now easily follows. □

The following corollary is an obvious consequence of Theorem 5.

Corollary 7. *Suppose that noise ν is zero. For any $\lambda \geq 1$,*

$$\left|\left\{i \in \{1,2,\ldots,n\} \ : \ |x_i - x_i^*| > \frac{8\varepsilon}{1-4\varepsilon}\frac{\lambda}{s}\|x\|_1\right\}\right| < \frac{s}{2\lambda} \ .$$

3.3 Approximation guarantees

In this section we finish the analysis of the approximation error. We start by showing that the set $\bar{I}$ contains all "important" coefficients of the vector x.

Lemma 8. $\|x - x_{\bar{I}}\| \leq \frac{1-8\varepsilon}{1-16\varepsilon}\|x - x_K\|_1 + \frac{4}{(1-16\varepsilon)d}\|\nu\|_1$.

Proof. Let $K_1 = K \setminus \bar{I}$. If $K_1 = \emptyset$ then the claim is clearly true. In general we need to show that $\|x_{K_1}\|_1$ is not much larger than $\|x_{\bar{I}\setminus K}\|_1$. Let $I' = \{i \in \bar{I} \ : \ |\Gamma(i) \cap \Phi(\bar{I})| \geq (1-4\varepsilon)d\} \setminus K$. According to Lemma 2, it is $|I'| \geq \frac{1}{2}|\bar{I}| - |K \cap \bar{I}| \geq k - |K \cap \bar{I}| = |K_1|$. Since every coordinate of $x^*_{I'}$ is not smaller than any coordinate of $x^*_{K_1}$ we see that $\|x^*_{I'}\|_1 \geq \|x^*_{K_1}\|_1$. Hence, $\|x_{I'}\|_1 + \|x_{I'} - x^*_{I'}\|_1 \geq \|x_{K_1}\|_1 - \|x_{K_1} - x^*_{K_1}\|_1$, and so $\|x_{K_1}\|_1 - \|x_{I'}\|_1 \leq \|x_{I'} - x^*_{I'}\|_1 + \|x_{K_1} - x^*_{K_1}\|_1$.

An upper bound on $\|x_{I'} - x^*_{I'}\|_1$ follows from Lemma 6, (in the context of Lemma 6 we have $\alpha = 4\varepsilon$, $\beta = 2\varepsilon$, and $\Delta = \frac{1}{d}\|\nu_{\Gamma(I')\cap\Phi(I)}\|_1$). Therefore, $\|x_{I'} - x^*_{I'}\|_1 \leq (1/2 - 4\varepsilon)^{-1}(2\varepsilon\|x - x_{\bar{I}}\|_1 + \frac{1}{d}\|\nu\|_1)$. To bound $\|x_{K_1} - x^*_{K_1}\|_1$ we apply Theorem 5, which gives $\|x_{K_1} - x^*_{K_1}\|_1 \leq (1/2 - 4\varepsilon)^{-1}(2\varepsilon\|x - x_{\bar{I}}\|_1 + \frac{1}{d}\|\nu\|_1)$. Combining the obtained inequalities we get that

$$\begin{aligned}
\|x - x_{\bar{I}}\| &\leq \|x - x_K\|_1 + \|x_{K_1}\|_1 - \|x_{I'}\|_1 \\
&\leq \|x - x_K\|_1 + \|x_{I'} - x^*_{I'}\|_1 + \|x_{K_1} - x^*_{K_1}\|_1 \\
&\leq \|x - x_K\|_1 + \frac{8\varepsilon}{1-8\varepsilon}\|x - x_{\bar{I}}\|_1 + \frac{4}{(1-8\varepsilon)d}\|\nu\|_1 \ ,
\end{aligned}$$

which implies the claimed bound. □

Lemma 9. *Suppose that $(\Delta_{ij})_{1\leq i<j\leq p}$ is a sequence of real values that satisfy for each i*

$$\sum_{j=i+1}^{p} \Delta_{ij} \leq \rho(\Delta_i + \Delta_{1i} + \ldots + \Delta_{i-1\,i}) \ ,$$

where ρ and Δ_i are some constants, with $0 < \rho < 1$. Let Δ denote $\sum_i \Delta_i$. The following inequality holds:

$$\sum_{i\geq 1}\sum_{j>i} \Delta_{ij} \leq \frac{\rho}{1-\rho}\Delta \ .$$

Proof. We have that

$$\sum_{i\geq 1}\sum_{j>i} \Delta_{ij} \leq \sum_{i\geq 1} \rho\Big(\Delta_i + \sum_{k=1}^{i-1}\Delta_{ki}\Big) = \rho\Delta + \rho\sum_{l\geq 1}\sum_{m>l}\Delta_{lm} \ .$$

As a result, $(1-\rho)\sum_{i\geq 1}\sum_{j>i}\Delta_{ij} \leq \rho\Delta$. □

Theorem 10. *Given a vector $c = Ax + \nu$, the algorithm returns approximation vector $\hat{x}$ satisfying*

$$\|x - \hat{x}\|_1 \leq \frac{1-4\varepsilon}{1-16\varepsilon}\,\|x - x_K\|_1 + \frac{6}{(1-16\varepsilon)d}\,\|\nu\|_1 \ .$$

where K is the set of the k largest (in magnitude) coordinates of x.

Proof. Let $R = \{1, 2, \ldots, n\} \setminus \bar{I}$ and $\Delta_j = \frac{1}{d}\|(A\,x_R + \nu)_{\Gamma(I'_j)\cap\Phi(I_j)}\|_1$, $j \geq 1$. Denoting $\Delta = \sum_j \Delta_j$, we have that

$$\Delta \leq \frac{1}{d}\|(A\,x_R + \nu)_{\bar{I}}\|_1 \leq 2\varepsilon\|x_R\|_1 + \frac{1}{d}\|\nu\|_1 \ .$$

When we write $x^*_{I'_j}$ it formally means $x^*_{I'_j}(c^j)$, where c^j is the value of vector c at the beginning of the jth iteration of step 6 of the algorithm. By Lemma 6 it is $\|x_{I'_1} - x^*_{I'_1}\|_1 \leq (1/2 - 2\varepsilon)^{-1}\Delta_1$. Let $\nu' = A(x_{I'_1} - x^*_{I'_1})$ and $\Delta_{1j} = \frac{1}{d}\|\nu'_{\Gamma(I'_j)\cap\Phi(I_j)}\|_1$, for $j \geq 2$. We have that

$$\sum_{j\geq 2}\Delta_{1j} \leq \frac{1}{d}\|(A(x_{I'_1} - x^*_{I'_1}))_{\Gamma(I_2)}\|_1 \leq 2\varepsilon\|x_{I'_1} - x^*_{I'_1}\|_1 \leq 2\varepsilon(1/2 - 2\varepsilon)^{-1}\Delta_1 \ .$$

To bound $\|x_{I'_2} - x^*_{I'_2}\|_1$ in the second step of the algorithm we will again use Lemma 6. Let $x' = x - x_{I'_1}$. For the second step we can write that $c = A\,x' + \nu + \nu'$, so $\nu + \nu'$ is viewed as noise. Since $x_{I'_2} = (x')_{I'_2}$, through Lemma 6 we get that $\|x_{I'_2} - x^*_{I'_2}\|_1 \leq (1/2 - 2\varepsilon)^{-1}(\Delta_2 + \Delta_{12})$. In general, let $\Delta_{ij} = \frac{1}{d}\|(A(x_{I'_i} - x^*_{I'_i}))_{\Gamma(I'_j)\cap\Phi(I_j)}\|_1$, for $j > i \geq 1$. Similarly as before we find that

$$\|x_{I'_j} - x^*_{I'_j}\|_1 \leq (1/2 - 2\varepsilon)^{-1}(\Delta_j + \Delta_{1j} + \ldots + \Delta_{j-1\,j}) \ .$$

Further,

$$\sum_{l>j}\Delta_{jl} \leq \frac{1}{d}\|(A(x_{I'_j} - x^*_{I'_j}))_{\Gamma(I_{j+1})}\|_1 \leq 2\varepsilon\|x_{I'_j} - x^*_{I'_j}\|_1 \leq 2\varepsilon(1/2 - 2\varepsilon)^{-1}(\Delta_j + \Delta_{1j} + \ldots + \Delta_{j-1\,j}) \ .$$

Denote the value $2\varepsilon(1/2 - 2\varepsilon)^{-1}$ by ρ. Summing the bounds on all the terms $\|x_{I'_j} - x^*_{I'_j}\|_1$ produces

$$\begin{aligned}
\sum_{j\geq 1}\|x_{I'_j} - x^*_{I'_j}\|_1 &\leq (1/2 - 2\varepsilon)^{-1}\Big(\Delta + \sum_{j\geq 1}\sum_{i<j}\Delta_{ij}\Big) \\
&= (1/2 - 2\varepsilon)^{-1}\Big(\Delta + \sum_{i\geq 1}\sum_{j>i}\Delta_{ij}\Big) \\
&\leq (1/2 - 2\varepsilon)^{-1}\Big(\Delta + \frac{\rho}{1-\rho}\Delta\Big) \quad \text{(By Lemma 9)} \\
&= (1/2 - 2\varepsilon)^{-1}\Delta(1-\rho)^{-1} \ .
\end{aligned}$$

Now we can write:

$$\begin{aligned}
\|x - \hat{x}\|_1 &= \|x_R\|_1 + \sum_{j\geq 1}\|x_{I'_j} - x^*_{I'_j}\|_1 \leq \\
&\leq \|x_R\|_1 + \Delta(1/2 - 2\varepsilon)^{-1}(1-\rho)^{-1} = \\
&= \|x_R\|_1 + \Delta(1/2 - 4\varepsilon)^{-1} \leq \|x_R\|_1\Big(1 + \frac{4\varepsilon}{1-8\varepsilon}\Big) + \frac{2}{(1-8\varepsilon)d}\|\nu\|_1 \ .
\end{aligned}$$

We finish the proof by plugging in the bound on $\|x_R\|_1$ given by Lemma 8. □

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. *Preprint*, 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.

[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.

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

[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. Practical near-optimal sparse recovery in the l_1 norm. *Allerton*, 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.

[NV07] D. Needell and R. Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. 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. In *Proc. IS&T/SPIE Symposium on Electronic Imaging*, 2006.

[Tro04] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Inform. Theory*, 50(10):2231–2242, Oct. 2004.

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

A Space efficient method for computing $\bar{I}$

We will describe how it is possible reduce the storage requirement to $O(kd \log n)$ bits, at the expense of increasing the running time to $O(nd \log n)$. Some constant factors in the parameters have to be increased as well. Suppose (in this paragraph only) that $\bar{I}$ is redefined so that the constant 2ε in (2) is changed to 6ε (any value higher than 3ε would in principle work). Observe that any element of $\bar{I} \setminus I$ that has at least $(1 - 4\varepsilon)d$ unique neighbors within the set $\bar{I} \setminus I$ must have at least $2\varepsilon d$ neighbors shared with the neighbors of I. Therefore at least half of the elements of $\bar{I} \setminus I$ belong to the set

$$T_1 = \{i \in \{1, 2, \ldots, n\} \setminus I \ : \ |\Gamma(i) \cap \Gamma(I)| \geq 2\varepsilon\} \ .$$

The algorithm first finds T_1, which takes $O(nd)$ time. The set $\bar{I}$ is initialized to I, and it will be expanded incrementally. To efficiently determine $T_1 \cap \bar{I}$ the algorithm constructs a priority queue over the set T_1 with the priority of element i being $|\Gamma(i) \cap \Gamma(\bar{I})|$; in this part the process is the same

as earlier. Time $O(kd)$ is spent on finding the intersection of T_1 and $\bar{I}$, since T_1 can have at most $|I| = 2k$ elements. It is clear how the algorithm can proceed to run in total time $O(nd \log n)$.