Explicit Constructions for Compressed Sensing of Sparse Signals

Piotr Indyk
MIT

July 12, 2007*

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 succinct representation of x is equal to Ax, where A is a carefully chosen $R \times n$ real matrix, $R << n$. Often, A is chosen at random from some distribution over $R \times n$ matrices. The vector Ax is often refered to as the *measurement vector* or a *sketch* of x. Although the dimension of Ax is much shorter than of x, it contains plenty of useful information about x (see next page).

Why is *linear* compression/sketching useful ? There are several reasons.

Sketch maintenance. A linear sketch Ax can be easily maintained under linear updates of the signal x. For example, updating the sketch after incrementing the i-th coordinate x_i can be easily done, since $A(x+e_i) = Ax + Ae_i$. Similarly, one easily obtains a sketch of a sum of two signals given the sketches for individual signals x and y, since $A(x+y) = Ax + Ay$. Both properties are very useful in several areas, notably network measurement and databases. For example, we can define x_i to be the number of packets sent through a network with destination address $i \in \{0 \dots 2^{32}-1\}$. The knowledge of such "state vector" is useful for the purpose of detecting unusual behavior in networks. Although the whole vector x is too large to be maintained, its sketch Ax is much shorter and easier to store. From the linearity of the mapping A, it follows that we can easily update the sketch of a state vector given a new packet with destination i. Finally, if the measurements performed by two different routers of the network yield state vectors x and y, one can easily compute a sketch of $x + y$. See e.g., [GKMS01b, EV03] for more on applications of sketching to network monitoring, and [AMS96, GGR02] for more on applications to databases.

Compressed sensing. Another scenario where linear compression is of key importance is *compressed sensing* [Don06, CT06], a rapidly developing area in digital signal processing. Here, x is a signal one wishes to sense, e.g., an image obtained from a digital camera. A camera "senses" the vector by computing a dot product with a number of pre-specified measurement vectors. A typical camera senses the whole vector x (consisting of, say, several megapixels), and then stores the compressed version (say, in a JPEG format). However, it may be much less costly to sense the compressed version directly, since then one needs much fewer sensors. This is a premise of "compressed sensing". See [WLD$^+$06] for a prototype camera built using this framework.

*Most of this work has been done in October 2005. However, the current version of this document has been greatly influenced (and simplified) thanks to the recent developments in the area, notably the notion of row tensor product [CM06, GSTV06] and the explicit construction of [CM06].

How much information about signal x can be retrieved from its sketch Ax ? As it turns out, quite a lot. In [AMS96] (cf. [JL84]) it was shown that one can retrieve the approximation to $\|x\|_2$ from a very short sketch Ax. Further developments [GGI$^+$02a] resulted in algorithms which, given a sketch of length $R = (k + \log n)^{O(1)}$, could recover a piece-wise constant approximation h of x, consisting of k pieces, such that $\|x-h\|_2 \leq (1+\epsilon)\|x-h^*\|_2$, where h^* is the best such approximation to x; the recovery time was polynomial in R. Similarly, one can find approximations of the signal using other representations, such as a linear combination of at most k vectors from some basis, e.g.. Fourier or wavelet[1]; see [GKMS01a, TGIK02, CCFC02, EV03, CM03, CM04] for some of the work along these lines [2]. These early results were *randomized* (or *non-uniform*), in the following sense: the matrix A was chosen from some distribution, and the reconstruction algorithm was guaranteed to be correct for a given x with "high" probability.

Recently, matrices with significantly stronger *uniform* recovery properties (and resulting in much shorter sketches!) have been discovered. Specifically, it was shown [Don06, CT06, CRT06] that one can select *one* matrix A which is "good" (in the above sense) for *all* signals x. Moreover, the vector x can be recovered from Ax via linear-programming, using a well-investigated method of *Basis Pursuit*. Faster (in fact, sublinear-time) algorithms for uniform recovery were discovered in [CM06, GSTV06, GSTV07]. Many other papers are devoted to this topic [Gro06].

Several of the aforementioned algorithms address a particularly simple class of *pure signals*, where the signal x is r-*sparse*, that is, it contains only r non-zero entries. In this case the aforementioned algorithms recover the signal x exactly, from a sketch as small as $O(r\log(n/r))$ [Don06, CT06, CRT06, CRTV05].

In this paper, we focus on this simple type of signals, and address the following drawbacks of the aforementioned algorithms: (i) the matrices A are generated *at random*, and (ii) there is no efficient method for *verifying* if a given matrix A does indeed have the strong reconstruction properties. For this and other reasons, it is interesting to have an *explicit* construction of a good matrix A. Unfortunately, the only results of this type suffer from the following problems: (i) either they require sketches of size at least quadratic in r [CM06, Mut06, DeV07], or (ii) representing matrix entries requires $\Omega(n)$ bits per entry (when van der Monde matrices are used, see [DeV06], p. 20). See [Tao07] for a discussion on explicit construction for compressed sensing.

Result. In this paper we provide an explicit construction of a *binary* $R\times n$ matrix A, such that one can recover an r-sparse signal x from Ax for R *near-linear* in r. Specifically, for some constant $E > 1$, $R = r2^{O(\log\log n)^E} = rn^{o(1)}$ measurements suffice. [3] The reconstruction is performed in time $O(R\log^{O(1)} n)$. Note that if $r << n$, the reconstruction time is sub-linear in n.

Techniques. The basic block in our construction is as follows. We use a small family of hash functions mapping $\{1\dots n\}$ into roughly $\{1\dots O(r)\}$ "buckets". A bucket is defined as *overflowing* if the number of non-zero elements of x mapped to it is greater than some threshold t. We show that if the hash functions are constructed using *extractor graphs*, then the hash functions

[1] Note that due to the linearity of sketching, it suffices to have an algorithm that recovers the best k-term approximation using the standard basis $e_1 \dots e_n$; the sketching matrix A can be adapted to any other basis by multiplying it by the change-of-basis matrix.

[2] The origin of these results can be traced even further in the past, to [Man92] (cf. [GGI$^+$02b, GMS05]). In [Man92, GGI$^+$02a], it was shown how to estimate the largest k Fourier coefficients of a given signal vector by accessing only $(k+\log n)^{O(1)}$ entries of the vector. By change-of-basis, this implies that one can estimate the k largest coefficients of a signal x by using only $(k+\log n)^{O(1)}$ dot products of (inverse) Fourier matrix rows with the signal. Amazingly enough, this implication has not been explicitly stated until the recent work of [CT06].

[3] The constant E depends on the best-known construction of an *extractor*.

distributes the non-zero elements of the vector x in such a way that "most" elements are mapped to non-overflowing buckets (this part of the construction resembles the use of extractors in a recent paper [Ind07] by the author). The elements in non-overflowing buckets are then recovered using group testing techniques.

Unfortunately, the recovery process is not complete, due to the elements in the overflowing buckets. The group testing procedure, applied to those buckets, can result in misidentification of the elements of x (both false positives and negatives are possible). Therefore, the whole procedure must be applied recursively, to the difference between the original vector x and the recovered vector. Since the sparsity of the difference vector is much lower than the sparsity of x, the whole process converges in $O(\log n)$ steps. Summing up all the difference vectors constructed during this process recovers the original vector x.

2 Preliminaries

Consider $x \in \mathbb{R}^n$. Define $\|x\|_0$ to be the number of non-zero entries of x.

We use notation $[n] = \{1 \dots n\}$. For any graph G with vertex set V, and $v \in V$, we use $\Gamma_G(v)$ to denote the set of neighbors of v in G. The subscript G is skipped if it is clear from the context. We also define $\Gamma(v, X)$, $v \in V, X \subset V$, to be the set of all edges from v to X.

3 Proof

We start by selecting a bipartite graph $G = (A, B, E)$, where $A = [n]$ and $B = [k]$. The nodes in A have degree d_A.

The graph is chosen so that it is an ϵ-*extractor* (see [Sha04] for an overview) for some $\epsilon > 0$ specified later. That is, G has the following property. For any $X \subset A$, $|X| \geq k$, consider the distribution $\mathcal{D}$ over B obtained by choosing $a \in X$ uniformly at random, and then choosing $b \in \Gamma(a)$ uniformly at random. Then $\mathcal{D}$ should satisfy

$$\sum_{b \in B} |\Pr_{\mathcal{D}}(b) - \frac{1}{|B|}| \leq \epsilon \tag{1}$$

It is known (e.g., see [Sha04]) that, for any absolute constant $\epsilon > 0$, ϵ-extractors can be explicitly constructed, with $d_A = 2^{O(\log \log n)^E}$, for a constant $E > 1$. It appears that the best known extractor construction guarantees $E = 2$ (see [Vad07], Lecture 12, Table 1).

For an integer $t > 0$, define

$$Overflow_t(X) = \{b \in B : |\Gamma(b, X)| > t\}$$

In the following we set $t = 2d_A$. Observe that, for any $b \in Overflow_t(X)$, we have $\Pr_{\mathcal{D}}[b] \geq 2/|B|$. It follows that

$$|Overflow_t(X)| \quad \leq \quad \epsilon |B| \tag{2}$$

In the following, we present the construction of the matrix A in parallel with the procedure for recovering x from Ax.

Let H be the adjacency matrix of the graph G. Let $X = \{i \in [n] : x_i \neq 0\}$, $|X| \leq r$. From the properties of G, for any $i \notin Overflow_t(X)$, the value $(Hx)_i$ is a sum of at most t different non-zero entries of x. Denote the indices of those entries by $\{j^i_1 \dots j^i_t\}$. In the following, we show that by using slightly more measurements, we can recover exactly those indices as well as the corresponding values; specifically, we recover the set $J_i = \{j^i_1, \dots, j^i_t\}$ and the sequence $U_i = x_{j^i_1}, \dots, x_{j^i_t}$. We then use the information in sets J_i to partially recover the vector x. The recovery will be incomplete, since in general we do not know if a given i belongs to $Overflow_t(X)$ at this stage.

We now show how to augment measurements H to a larger set of measurements H', so that given $H'x$ we can recover J_i for any $i \notin Overflow_t(X)$. The augmentation uses the notion of *row tensor product* [CM06, GSTV06], defined as follows.

Definition 1. *Given a $v \times n$ matrix V, and a $w \times n$ matrix W, the row tensor product $V \otimes W$ is a $vw \times n$ matrix such that $(V \otimes W)_{iv+l,j} = V_{i,j} * W_{l,j}$.*

In words, the matrix $(V \otimes W)$ is obtained by copying each row of V w times, and performing a coordinate-wise multiplication of each copy with a respective row of W.

In [CM06], the authors provided an explicit construction of an $n \times R'$ matrix B, $R' = (t + \log n)^{O(1)} = 2^{O(\log^2 \log n)}$, such that for any t-sparse vector $x \in \mathbb{R}^n$, one can recover x given Bx in time $R'^{O(1)}$. We now define $H' = H \otimes B$. Observe that, given $H'x$, we can recover, for each $i \notin Overflow_t(X)$, the sets J_i and U_i, by applying the reconstruction algorithm of [CM06] to each such i. Of course, for each $i \in Overflow_t(X)$, the reconstruction algorithm will report some sets J_i and U_i as well. To "reduce the damage", we check for each i if the cardinality of a reported set J_i exceeds t. If this is the case, we set $J_i = \emptyset$; otherwise we leave J_i intact.

We now show how to use the information in J_i to recover x. The recovery process will consist of two stages. First, we show how to use J_i's to construct a vector y which is very close to x, in the sense that $\|x - y\|_0 = O(\epsilon k)$. We call this procedue REDUCE. Second, we show how apply reduction recursively to completely recover x.

The reduction procedure is as follows. It receives as input the sparsity parameter k, the measurement matrix H', and a vector $b = Fx$ for some vector x such that $\|x\|_0 \leq k$.

The procedure maintains a vector $votes[\cdot]$ of multisets. Initially, all entries are set to $\emptyset$.

```
REDUCE(H', b, k)
  FOR l = 1...k
    COMPUTE J_l AND U_l (AS DESCRIBED EARLIER)
    FOR EACH j ∈ J_l
      votes[j] = votes[j] ∪ {u_j}
  y = 0
  FOR j = 1...n
    IF votes[j] CONTAINS > d_A/2 COPIES OF val THEN y_j = val
RETURN y
```

Lemma 1. *The procedure* REDUCE *outputs a vector y such that $\|y - x\|_0 = O(\epsilon k)$.*

Proof. By equation (2) we know that

$$\sum_{i \in Overflow_t(X)} |J_i| \leq \epsilon |B| t$$

Thus, at most $\epsilon|B|t$ "correct votes" have been replaced by at most $\epsilon|B|t$ "incorrect votes". Since $d_A/2$ vote changes are needed to make any entry y_j to have an incorrect value, it follows that at most

$$\frac{2\epsilon|B|t}{d_A/2} = 8\epsilon|B| = 8\epsilon k$$

entries of y are incorrect.

□

In the following we set $\epsilon < 1/16$; this implies that $\|x - y\|_0 < k/2$. Now, it suffices to recover the difference vector $\Delta = x - y$. Since $\|\Delta\|_0 < k/2$, we can perform this task recursively.

The formal description of the final algorithm is as follows. We prepare the matrices H' for $k = r, r/2, \ldots, 1$; denote them by $H'(r), \ldots, H'(1)$. The compressed sensing matrix A is equal to a vertical concatenation of matrices $H'(k), \ldots, H'(1)$. Then we invoke a function Recover(b, r), described as follows.

```
RECOVER(b, k)
  y = REDUCE(H'(k), k, b)
  { WE NOW NEED TO RECOVER x − y}
  z = 0
  IF k > 1 THEN z =RECOVER(b − H'(k)y, k/2)
  RETURN y + z
```

Acknowledgments

The author would like to thank Mark Davenport, Anna Gilbert, Howard Karloff and Martin Strauss for very helpful comments.

References

[AMS96] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Proceedings of the Symposium on Theory of Computing*, pages 20–29, 1996.

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

[CM03] G. Cormode and S. Muthukrishnan. What is hot and what is not: tracking most frequent items dynamically. *Proceedings of the ACM Symposium on Principles of Database Systems*, 2003.

[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. *SIROCCO*, 2006.

[CRT06] E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52:489 – 509, 2006.

[CRTV05] E. Candes, M. Rudelson, T. Tao, and R. Vershynin. Error correction via linear programming. *Proceedings of the Symposium on Foundations of Computer Science*, 2005.

[CT06] E. Candes and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies. *IEEE Transactions on Information Theory*, 2006.

[DeV06] R. DeVore. Optimal computation. *Proceedings of the International Congress of Mathematicians*, 2006.

[DeV07] R. DeVore. Deterministic constructions of compressed sensing matrices. *preprint*, 2007.

[Don06] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289 – 1306, 2006.

[EV03] C. Estan and G. Varghese. New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. *ACM Transactions on Computer Systems*, 2003.

[GGI+02a] A. Gilbert, S. Guha, P. Indyk, Y. Kotidis, M. Muthukrishnan, and M. Strauss. Fast, small-space algorithms for approximate histogram maintenance. *Proceedings of the Symposium on Theory of Computing*, 2002.

[GGI+02b] A. Gilbert, S. Guha, P. Indyk, M. Muthukrishnan, and M. Strauss. Near-optimal sparse fourier representations via sampling. *Proceedings of the Symposium on Theory of Computing*, 2002.

[GGR02] M. Garofalakis, J. Gehrke, and R. Rastogi. Querying and mining data streams: You only get one look. *http://www.cs.berkeley.edu/ minos/tutorials.html*, 2002.

[GKMS01a] A. Gilbert, Y. Kotidis, M. Muthukrishnan, and M. Strauss. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. *Proceedings of the International Conference on Very Large Databases (VLDB)*, 2001.

[GKMS01b] A. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. Strauss. Quicksand: Quick summary and analysis of network data. *DIMACS Technical Report 2001-43*, 2001.

[GMS05] A. Gilbert, M. Muthukrishnan, and M. Strauss. Improved time bounds for near-optimal space fourier representations. *SPIE Conference, Wavelets*, 2005.

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

[GSTV06] A. Gilbert, M. Strauss, J. Tropp, and R. Vershynin. Algorithmic linear dimension reduction in the l1 norm for sparse vectors. *44th Annual Allerton Conference on Communication, Control, and Computing*, 2006.

[GSTV07] A. Gilbert, M. Strauss, J. Tropp, and R. Vershynin. One sketch for all: fast algorithms for compressed sensing. *Proceedings of the Symposium on Theory of Computing*, 2007.

[Ind07] P. Indyk. Uncertainty principles, extractors and explicit embedding of l2 into l1. *Proceedings of the Symposium on Theory of Computing*, 2007.

[JL84] W.B. Johnson and J. Lindenstrauss. Extensions of lipshitz mapping into hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[Man92] Y. Mansour. Randomized interpolation and approximation of sparse polynomials. *ICALP*, 1992.

[Mut06] S. Muthukrishnan. Compressed sensing algorithms for functions. *Allerton Conference on Communication, Control and Computing*, 2006.

[Sha04] R. Shaltiel. Recent developments in extractors. In *Current trends in theoretical computer science. The Challenge of the New Century. Vol 1: Algorithms and Complexity*, 2004.

[Tao07] T. Tao. Open question: deterministic uup matrices. *Web log at* `http://terrytao.wordpress.com/2007/07/02/open-question-deterministic-uup-matrices/`, 2007.

[TGIK02] N. Thaper, S. Guha, P. Indyk, and N. Koudas. Dynamic multidimensional histograms. *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2002.

[Vad07] S. Vadhan. Pseudorandomness (cs225). *available at* `http://www.eecs.harvard.edu/ salil/`, Spring 2007.

[WLD+06] Michael Wakin, Jason Laska, Marco Duarte, Dror Baron, Shriram Sarvotham, Dharmpal Takhar, Kevin Kelly, and Richard Baraniuk. An architecture for compressive imaging. *Proc. Int. Conf. on Image Processing (ICIP)*, 2006.