



Compressive
Sampling

© DIGITAL VISION

Imaging via Compressive Sampling

[Introduction to compressive sampling
and recovery via convex programming]

[Justin Romberg]

The ease with which we store and transmit images in modern-day applications would be unthinkable without compression. Image compression algorithms can reduce data sets by orders of magnitude, making systems that acquire extremely high-resolution images (billions or even trillions of pixels) feasible.

There is an extensive body of literature on image compression, but the central concept is straightforward: we transform the image into an appropriate basis and then code only the important expansion coefficients. The crux is finding a good transform, a problem that has been studied extensively from both a theoretical [14] and practical [25] standpoint. The most notable product of this research is the wavelet transform [9], [16]; switching from sinusoid-based representations to wavelets marked a watershed in image compression and is the

Digital Object Identifier 10.1109/MSP.2007.914729

essential difference between the classical JPEG [18] and modern JPEG-2000 [22] standards.

Image compression algorithms convert high-resolution images into a relatively small bit streams (while keeping the essential features intact), in effect turning a large digital data set into a substantially smaller one. But is there a way to avoid the large digital data set to begin with? Is there a way we can build the data compression directly into the acquisition? The answer is yes, and is what compressive sampling (CS) is all about.

To begin, we need to generalize our notion of “sampling” an image. Instead of collecting point evaluations of the image X at distinct locations, or averages over small areas (pixels), each measurement y_k in our acquisition system is an inner product against a different *test function* ϕ_k :

$$y_1 = \langle X, \phi_1 \rangle, \quad y_2 = \langle X, \phi_2 \rangle, \dots, y_m = \langle X, \phi_m \rangle. \quad (1)$$

We note here that our entire discussion in this article (and the majority of the work to date in the field of CS) will revolve around *finite dimensional* signals and images. To make the transition to acquisition of continuous-time (and -space) signals, we would choose a *discretization space* on which to apply the discrete theory. For example, we might assume that the image is (or can very closely be approximated by) a gridded array of n pixels. The test functions ϕ_k , which would also be pixelated, then give us measurements of the projection of the continuous image onto this discretization space.

The choice of the ϕ_k allows us to choose in which domain we gather information about the image. For example, if the ϕ_k are sinusoids at different frequencies, we are essentially collecting Fourier coefficients (as in magnetic resonance imaging), if they are delta ridges, we are observing line integrals (as in tomography), and if they are indicator functions on squares, we are back to collecting pixels (as in a standard digital camera). Imagers that take these generalized kinds of samples are often referred to as *coded imaging systems*, as the measurements y_1, \dots, y_m are in some sense a coded version of the image X rather than direct observations. For now we will assume that we have complete control over which ϕ_k to use—description of real coded imaging systems that implement various types of measurements can be found in [15], [19], and [23].

How then should we choose the ϕ_k to minimize the number of measurements m we need to reconstruct X faithfully? One idea is to match these test functions to the image structure. That is, we try to make the measurements in the same domain in which we would compress the image. For example, if the ϕ_k are cosines at a series of “low-pass” frequencies, the imaging system is computing the important discrete cosine transform (DCT) coefficients (the first step of JPEG compression) up front. With the ϕ_k matched to the signal structure, we reconstruct our

image using least squares, finding the closest image that matches the observed projection onto the span of $\{\phi_1, \dots, \phi_m\}$:

$$\hat{X} = \Phi^*(\Phi\Phi^*)^{-1}y, \quad (2)$$

where Φ is the linear operator that maps an image to a set of m measurements, Φ^* is its adjoint, and y is the m -vector of observed values (we can think of the ϕ_k as the rows of Φ). The effectiveness of this coded imaging strategy, along with the linear reconstruction procedure (2), is determined by how well images of interest can be approximated in the *fixed* linear subspace spanned by the ϕ_k . Speaking qualitatively, an acquisition device with the ϕ_k as low-frequency sinusoids will perform on par with the JPEG compression standard.

Although linear-coded imaging systems of this type have the potential to outperform traditional systems, they have a severe shortcoming. They cannot *adapt* to changes in structure from one image to the next; they are stuck recording the same m transform coefficients for every image. This problem

becomes particularly pronounced when we consider using wavelets for the ϕ_k above. The entire advantage wavelets hold over Fourier-based representations comes from the fact that they automatically adapt to singularities in the image; important wavelet coefficients tend to cluster around edge contours, while

large smooth regions can be built up with relatively few terms, facts that modern compression algorithms take full advantage of. While the wavelet transform is, of course, linear and the approximate image the compression algorithm produces can be written as a linear combination of wavelet basis functions, *which* wavelets are included in this approximation changes from image to image. (In mathematics, approximating a function by projecting it onto a fixed linear subspace is called linear approximation while projecting onto a linear subspace adapted to the function is called nonlinear approximation. See [10] for a beautiful introduction.)

Since they have the entire image (or at least a high-resolution version of it) and its transform to examine at their leisure, image compression algorithms can incorporate these adaptations with ease [21]. The same luxury is not afforded to an acquisition system, as there is no way to judge which transform coefficients are important until after the image is formed.

This brings us to our next question: Is there any way to match the *adaptive* approximation performance with a *pre-determined* set of linear measurements? Surprisingly, there is. Even more surprising is the way that the measurements should be taken. Instead of carefully matching the test functions ϕ_k to the structure in the image, the ideal thing to do is the exact opposite. The ϕ_k should be completely unstructured and look more like random noise than any feature we would expect to see in our image.

**THE CENTRAL CONCEPT
IS STRAIGHTFORWARD: WE
TRANSFORM THE IMAGE INTO AN
APPROPRIATE BASIS AND THEN
CODE ONLY THE IMPORTANT
EXPANSION COEFFICIENTS.**

The central result of CS [3]–[5], [11] is that from m of these noise-like, *incoherent* measurements, we can reconstruct the image as well as if we had observed the $m/\log n$ most important wavelet coefficients, where n is the number of pixels in our recovery (we can think of n as being the target resolution). (For the sake of the exposition, we are forgoing all constants in this part of the discussion.) In practice, the story is even better: from m measurements, we are usually able to reconstruct an image comparable to the best $m/5$ term wavelet approximation. To put this in context, consider that for a typical n -pixel image, an almost lossless approximation can be constructed from just 5% ($.05n$) of the wavelet coefficients. Then from $0.25n$ measurements we can reconstruct an image that is almost as good as if we had measured each of the n pixels individually, a savings of 75%. Thus the number of measurements we need depends more on the inherent complexity of the image (in the wavelet domain) than on the number of pixels we wish to reconstruct. Furthermore, there is nothing exceptional about the wavelet transform, this strategy applies to nonlinear approximation in any transform domain.

While the measurements remain linear (they are the same as in (1), with noise-like pseudorandom ϕ_k), the CS recovery procedure is decidedly nonlinear. It has to be, as I imagine there is no way I could convince the reader that the linear recovery in (2) will be more effective with random noise than with low-frequency sinusoids. Given the m -vector of measurements $y = \Phi X$, the recovery algorithm consists of solving the convex program

$$\min_{X'} \|\Psi^* X'\|_{\ell_1} \quad \text{subject to} \quad \Phi X' = y. \quad (3)$$

Let us describe what (3) is doing in words before turning to the mathematics: It is searching for the n -pixel image with the *sparsest* wavelet transform that explains the measure-

ments we have observed. (Again, we can replace the wavelet transform here with any suitable transform in which images we are interested in acquiring are compressible.) The constraints $\Phi X' = y$ ask that we only consider images that would produce the same measurements y which we have observed; the operator Ψ^* takes an n -point wavelet transform of a candidate image I' , and the ℓ_1 norm of the resulting n -vector $w = \Psi^* X'$, is simply the sum-of-magnitudes

$$\|w\|_{\ell_1} = \sum_{i=1}^n |w(i)|.$$

As we will discuss in detail below, we use the ℓ_1 norm in (3) because

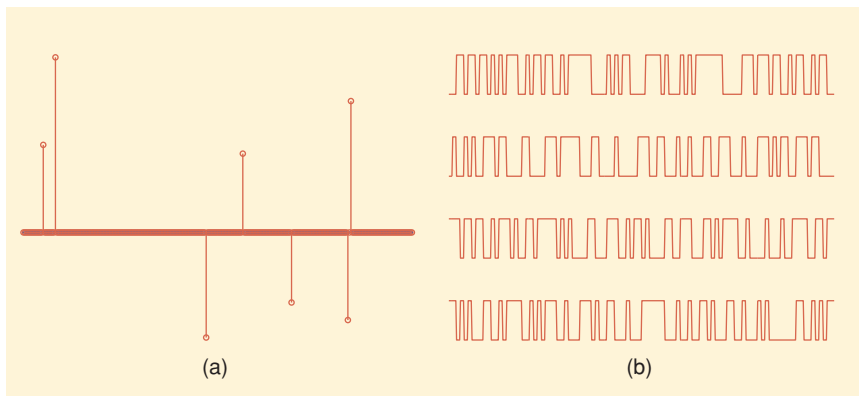
a) sparse signals have small ℓ_1 norms relative to their energy, and b) it is convex, which makes the optimization problem computationally tractable. Reconstruction using (3) is effective for the same reason wavelet-based image compression algorithms are effective: it is making the most of the information it has by exploiting the fact that images tend to have sparse wavelet transforms.

Solving (3) accomplishes two things: it *locates* the important transform coefficients and *reconstructs* their values. It is in this first task that the incoherence of the measurements comes into play. While we know that the image we are trying to acquire is sparse in the wavelet domain, it is critical that our measurement functions (and linear combinations thereof) not be. Figure 1 illustrates this point: if we don't know where the important values are, sampling the transform coefficients in (a) directly (which is the same as using wavelets for the ϕ_k in our discussion above) will be for the most part a fruitless effort, as most of the time we will see values that are very close to zero. If instead we take *global* combinations of the transform coefficients, an effect achieved by using incoherent ϕ_k as illustrated in (b), we “pick up” a little bit of information about the sparse coefficient sequence with each measurement. The program (3) then “triangulates” the locations of important transform coefficients and their values.

The concepts in the preceding paragraph are made precise by establishing certain *uncertainty principles* between the domain in which the image is sparse and the domain in which it is being measured. While we can expect that the signal is concentrated on a somewhat arbitrary and relatively small set in the Ψ domain, the measurements will be spread out.

We will explain exactly what we mean by an uncertainty principle and explore its ramifications later. Before going further, however, it will probably help to see a numerical experiment that demonstrates the potential of CS.

THE MATHEMATICAL THEORY UNDERLYING CS IS DEEP AND BEAUTIFUL AND DRAWS FROM DIVERSE FIELDS.



[FIG1] Sampling a sparse vector. (a) An example of a very sparse vector. If we sample this vector directly with no knowledge of which components are active, we will see nothing most of the time. (b) Examples of pseudorandom, incoherent test vectors ϕ_k . With each inner product of a test vector from (b), we pick up a little bit of information about (a).

CS IN ACTION

In Figure 2, we see the famous *Camera Man* image. Of course, this image has already been acquired and is sitting in “high resolution” (here a rather modest 256×256) pixelated form on my hard drive. But we can use this high-resolution version to simulate how well we could have acquired it from measurement functions formed by linear combinations of these pixels (i.e., the ϕ_k in (1) are piecewise constant over the pixel regions). Taking m of these measurements is equivalent to applying an $m \times 65,536$ matrix Φ to the 65,536 dimensional vector X representing the (rasterized) high-resolution image.

We will compare two different coded imaging strategies. (The MATLAB code that reproduces the results in this section can be downloaded from <http://users.ece.gatech.edu/~justin/spmag>.) The first scheme, which we will call simply “linear imaging,” measures low-pass DCT coefficients and reconstructs using the pseudo-inverse as in (2). The order in which the DCT coefficients are acquired is the same zig-zagging pattern used by the JPEG compression standard [18]. (We are not breaking the image into sub-blocks here as JPEG does, but the results do not change significantly if we do.) The results of this strategy are shown as the blue curve in Figure 2(a). The coarse-scale features of the image are acquired very quickly: we capture about 97% of the energy of X in the first 1,000 (1.5% of 65,536) DCT coefficients. The curve levels off after this, suggesting that the linear imaging scheme is very good at getting a rough sketch of the image but less efficient at filling in the details.

The second scheme, which we call “compressive imaging,” again sketches the image by measuring the first 1,000 DCT

coefficients but then switches to pseudorandom ϕ_k to acquire the details. The ϕ_k used here are binary-valued functions called *noiselets* [8]. These particular functions have a number of desirable properties (see [2] for a discussion), but for our purposes here we can just view the measurements $\langle X, \phi_k \rangle$ as randomly changing the signs of each of the pixels of X and then summing the results. We stress,

though, that the exact nature of the measurements is not too important, and we achieve very similar results with many different choices of incoherent test functions.

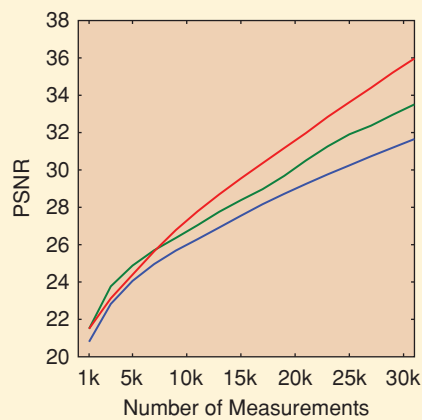
From this combination of low-pass DCT and noiselet coefficients, we reconstruct the image using a variant of (3). Instead of minimizing the ℓ_1 norm of the wavelet transform, we minimize the *total variation*, which can be interpreted as the ℓ_1 norm of the (appropriately discretized) gradient (we are taking the magnitude of a two-vector at each point in the image, rather than a scalar)

$$\min_{X'} \sum_{i,j} |(\nabla X')_{i,j}| \quad \text{subject to} \quad \Phi X' = y. \quad (4)$$

The program (4) is in much the same spirit as (3) but tends to work a little better in practice.

The results for the compressive imaging strategy are plotted as the red curve in Figure 2(a). The difference between compressive and linear imaging is dramatic; not only are the reconstructions uniformly better, but they are improving at a faster *rate* as measurements are added (just as the theory predicts). An example of an image acquired from 21,000 measurements (about 32% of the number of pixels) for both strategies is also shown in Figure 2. Note in particular that the CI reconstruction is cleaner around the edges than the linear reconstruction.

IS THERE A WAY WE CAN BUILD THE DATA COMPRESSION DIRECTLY INTO THE ACQUISITION?



[FIG2] Coded imaging simulation. (a) Recovery error versus number of measurements for linear DCT acquisition (blue), compressive imaging (red), and DCT imaging augmented with total-variation minimization (green). The error is measured using the standard definition of peak signal-to-noise ratio: $\text{PSNR} = 20 \log_{10}(255 \cdot 256 / \|X - \hat{X}\|_2)$. (b) Image recovered using linear DCT acquisition with 21,000 measurements. (c) Image recovered using compressive imaging from 1,000 DCT and 20,000 noiselet measurements.

There are really two innovations in the compressive imaging scheme: we are using pseudorandom measurements in place of low frequency sinusoids, and the nonlinear reconstruction given by (4) which is designed to favor our notion of image structure. It is fair to ask if the gains from compressive imaging would disappear if we were to use (4) instead of the pseudoinverse (2) to reconstruct the image from the DCT measurements. (Making the switch the other way, using the pseudoinverse to reconstruct from the incoherent measurements, would be a complete disaster.) They do not, as shown by the green curve in Figure 2(a). Using the nonlinear reconstruction program boosts the performance of the DCT imaging system, but the details are still coming in more slowly than with the incoherent measurements.

**THE MORAL IS VERY GENERAL:
A GOOD SIGNAL REPRESENTATION
CAN FUNDAMENTALLY AID THE
ACQUISITION PROCESS.**

UNCERTAINTY PRINCIPLES AND SPARSE RECOVERY

We turn briefly away from imaging to consider the following idealized problem. Say that $\alpha_0 \in \mathbb{R}^n$ is a sparse vector in that only a small number, which we will call S , of its n components are nonzero (later we will consider such α_0 as the transform coefficients of an image). We wish to acquire α_0 using as few linear measurements (inner products with test functions as in (1)) as possible. We know that α_0 is sparse, but have no information whatsoever about which components happen to be nonzero.

One strategy for acquiring α_0 is to sample it directly, recording $\alpha_0(i)$ for different location indices i . If we knew the locations of the S active components, we could simply take S samples at these locations and be done. If, however, we do not know anything about which components are active, sampling α_0 directly is a bad idea. Since $S \ll n$, most of our samples will be zero, telling us very little about which components are active and nothing at all about what values those active components have. In general, we will have to take all n samples to make sure that nothing is missing.

A better idea is to measure a series of random combinations of the entries of α_0 . Instead of taking samples, we observe inner products of α_0 against a series of random codes. Our test vectors $\phi_1, \phi_2, \dots, \phi_m$ will be binary valued and have unit norm, taking values of $+1/\sqrt{n}$ and $-1/\sqrt{n}$ with equal probability; examples are shown in Figure 1. The idea here is that since the measurements are global, we learn something new about the sparse vector with every measurement.

This random sensing strategy works because each sparse signal will have a unique set of measurements. We can make this mathematically precise as follows. Stack the m random test vectors $\phi_1, \phi_2, \dots, \phi_m$ on top of one another as rows in the $m \times n$ matrix Φ . If the number of rows obeys

$$m \gtrsim S \log n, \quad (5)$$

then with probability very, very close to one, the matrix obeys what is known as a *uniform uncertainty principle* (UUP) [5]. (Here we mean that $m \geq \text{Const} \cdot S \log n$ for some known constant. We are not trying to hide anything using this notation, the constants involved are almost always small; see in particular [13].) The UUP states that for any S -sparse vector h , the energy of the measurements Φh will be comparable to the energy of h itself:

$$\frac{1}{2} \cdot \frac{m}{n} \cdot \|h\|_2^2 \leq \|\Phi h\|_2^2 \leq \frac{3}{2} \cdot \frac{m}{n} \cdot \|h\|_2^2. \quad (6)$$

We call this is an uncertainty principle because the proportion of the energy of h that appears as energy in the measurements is roughly the same as the under-sampling ratio m/n . While h is entirely concentrated on a small set, it is spread out more or less

evenly in the measurement domain.

To see how the UUP relates to sparse recovery, suppose that (6) holds for sets of size $2S$ [we will have to double the constant in (5)]. We measure our S sparse vector as above: $y = \Phi \alpha_0$. Is there any other S -sparse (or sparser) vector $\alpha' \neq \alpha_0$ that has the same measurements? The answer is no. If there were such a vector, then the difference $h = \alpha_0 - \alpha'$ would be $2S$ -sparse and have $\Phi h = 0$, two properties made incompatible by the UUP.

To recover an S sparse α_0 from $y = \Phi \alpha_0$ (inverting the measurement process), we solve an optimization problem. From the arguments above, we know that α_0 is the sparsest vector which maps to y . Thus it must be the solution to the following optimization problem:

$$\min_{\alpha} \#\{i : \alpha(i) \neq 0\} \quad \text{subject to} \quad \Phi \alpha = y. \quad (7)$$

The functional $\#\{i : \alpha(i) \neq 0\}$ is simply the number of nonzero terms in the candidate vector α ; in the literature this is sometimes referred to as the ℓ_0 norm (although is not a vector norm, strictly speaking).

The problem with (7) is that solving it directly is infeasible. It is combinatorial and NP-hard [17]. Fortunately, there is a convex program that works almost as well:

$$\min_{\alpha} \|\alpha\|_{\ell_1} \quad \text{subject to} \quad \Phi \alpha = y. \quad (8)$$

The only difference between (7) and (8) is the substitution of sum of magnitudes in place of size of support. But (8) is far easier to solve; it can be recast as a linear program [7] and solved using any number of modern techniques [1].

Even though (7) and (8) are fundamentally different, they produce the same answer in many interesting situations. Under essentially the same assumptions (the constant in (5) is slightly larger), the program (8) will also recover an S -sparse α_0 from its measurements $y = \Phi \alpha_0$. This fact is

of utmost importance, and it is what promotes sparse recovery from an intellectual curiosity to an idea which can have broad impact.

THE GEOMETRY OF ℓ_1 MINIMIZATION

We can get some geometric intuition for why ℓ_1 is an effective substitute for sparsity by turning to the sketches in Figure 3. Part (a) illustrates the ℓ_1 ball in \mathbb{R}^2 of a certain radius. Note that it is anisotropic; it is “pointy” along the axes (compare to the standard Euclidean ℓ_2 ball, which is spherical and thus completely isotropic). Part (b) diagrams the ℓ_1 recovery program (also in \mathbb{R}^2): the point labeled α_0 is a “sparse” vector (only one of its components are nonzero) of which we make one measurement; the line labeled H is the set of all α that share the same measurement value.

The task for (8) is to pick out the point on this line with minimum ℓ_1 norm. To visualize how (8) accomplishes this, imagine taking an ℓ_1 ball of tiny radius and gradually expanding it until it bumps into H . This first point of intersection is by definition the vector that solves (8). The combination of the anisotropy of the ℓ_1 ball and the flatness of the space H results in this intersection occurring at one of the points, precisely where sparse signals are located.

Compare to what would happen if we replaced the ℓ_1 norm with the ℓ_2 norm (which would make the recovery a least-squares problem). Figure 3(c) replaces the diamond-shaped ℓ_1 ball with the spherical and perfectly isotropic ℓ_2 ball. We can see that the point of first intersection of H and the expanding ℓ_2 ball does not have to be sparse at all. In high dimensions this difference becomes very dramatic. Despite the seemingly innocuous difference in the definitions of the ℓ_1 and ℓ_2 norms (sum of magnitudes versus sum of magnitudes squared), they are totally different creatures.

ℓ_1 RECOVERY AND UNCERTAINTY PRINCIPLES

The precise arguments demonstrating that (8) recovers sparse signals (given that Φ obeys the uncertainty principle) are a little more involved than those we made for the combinatorial recovery below (7), but they have much of the same flavor. We will give only an outline of the reasoning here; interested readers can consult [4], [12], and [24] for the details. We want to show that if α_0 is sparse, then for all α'

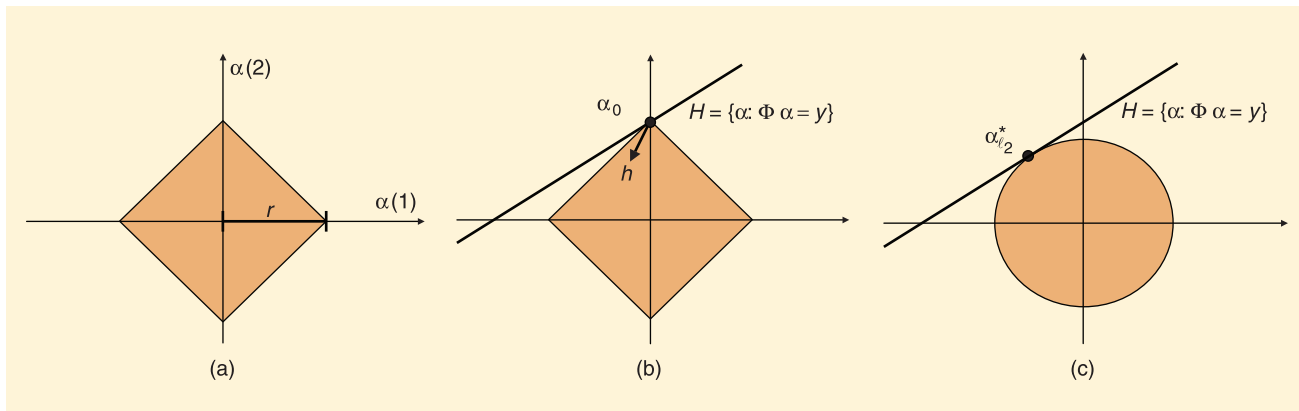
with $\|\alpha'\|_{\ell_1} \leq \|\alpha_0\|_{\ell_1}$ we have $\Phi\alpha' \neq \Phi\alpha_0$. Returning to our diagram, we can see that (8) will recover α_0 if the line H does not “cut through” the ℓ_1 ball at α_0 . Another way to say this is that for every h in the cone of descent from the facet of the ℓ_1 ball on which α_0 lives (meaning $\|\alpha_0 + h\|_{\ell_1} \leq \|\alpha_0\|_{\ell_1}$), we will have $\Phi h \neq 0$.

The key, just as in the combinatorial case, is that all descent vectors h are concentrated on the same (relatively small) set as α_0 . Of course, they do not have to be supported exactly on this set, but the “pointiness” of the ℓ_1 ball at the low-dimensional facet on which α_0 lives severely constrains how descent vectors can behave. We have seen that (6) directly implies that vectors supported on sets with size proportional to S cannot be in the null space of Φ ; showing that this extends to vectors which are merely concentrated on such sets is what [4], [12], and [24] accomplish.

RECOVERY OF SPARSE TRANSFORMS

In general, we are not as interested in reconstructing signals that are by themselves sparse but rather are sparse in some known transform domain. Making this transition is straightforward; instead of (8), we use (3) in the introduction, with an orthonormal Ψ representing the transform in which we expect our signals of interest to be sparse. Of course, now we need to take measurements that are incoherent in the Ψ domain;

THIS SHORT, CONVERSATIONAL ARTICLE IS MEANT AS AN INTRODUCTION TO CS AND RECOVERY VIA CONVEX PROGRAMMING.



[FIG3] Geometry of ℓ_1 recovery. (a) ℓ_1 ball of radius r ; the orange region contains all $\alpha \in \mathbb{R}^2$ such that $|\alpha(1)| + |\alpha(2)| \leq r$. (b) Solving the min $-\ell_1$ problem (8) allows us to recover a sparse α_0 from $y = \Phi\alpha_0$, as the anisotropy of the ℓ_1 ball favors sparse vectors. Note that the descent vectors h pointing into the ℓ_1 ball from α_0 will be concentrated on the support of α_0 . (c) Minimizing the ℓ_2 norm does not recover α_0 . Since the ℓ_2 ball is isotropic, the min $-\ell_2$ solution α_{r2}^* will in general not be sparse at all.

instead of measuring random combinations of pixels, we should, strictly speaking, measure random combinations of basis functions. It is often the case, though, that measurements of random combinations of the pixels behave enough like random combinations of the basis functions so that they are functionally indistinguishable in practice.

DISCUSSION

This short, conversational article is meant as an introduction to CS and recovery via convex programming. In many places we have sacrificed precision to develop intuition. The mathematical theory underlying CS, however, is deep and beautiful, and draws from diverse fields including harmonic analysis, convex optimization, random matrix theory, statistics, approximation theory, and theoretical computer science. To conclude, we will briefly touch on several interesting topics we did not discuss above.

FIXED MEASUREMENT SYSTEMS

We do not always have complete control over the types of measurements we make. If our equipment make measurements in the U domain and our model is that signals are sparse in the Ψ domain, the number of samples we need to reconstruct now depends on how different the U and Ψ domains are. Recovery results exist [2], [20] that are similar to those using random matrices, but with the additional dependence on a *coherence parameter* that quantifies how concentrated elements of ψ are in the U domain (and vice versa).

STABILITY

In practice, signals and images are never perfectly sparse, and we can always expect to have some amount of measurement error. Because the recovery procedure is nonlinear, there is a natural worry that it is unstable. This is not the case at all. The program (3) does a very good job of recovering signals that are only approximately sparse [4], [5]. When the measurements y are perturbed, there are various ways to *relax* [4], [6], [24] the program (3) so that the recovery error is on the same order as the measurement error.

COMPUTATIONS

While the actual acquisition process (i.e., the “analog encoding”) is trivial—we simply take inner products—solving the recovery program is not. Fortunately, there have been drastic advances in the field of convex optimization [1] that make solving these types of programs feasible on the scale that we are interested in (hundreds of thousands of measurements and millions of pixels). There is a good deal of computation involved in these algorithms, but it manageable. A good rule of thumb for these methods is that solving the ℓ_1 minimization program (3) is about 30–50 times as expensive as solving the least-squares problem (2).

Finally, although we have couched the entire discussion in this paper in terms of imaging, CS is applicable to a wide variety of problems. The moral is very general: a good signal representation can fundamentally aid the acquisition process.

AUTHOR

Justin Romberg (jrom@ece.gatech.edu.) attended Rice University, where he received the B.S. (1997), M.S. (1999), and Ph.D. (2003) degrees in electrical engineering. In the fall of 2003, he joined the Applied and Computational Mathematics Department at Caltech as a Postdoctoral Scholar, where he worked on the theoretical foundations of compressive sampling. He spent the fall of 2004 as a Fellow at UCLA's Institute for Pure and Applied Mathematics. In the fall of 2006, he joined the faculty at Georgia Tech as an assistant professor in the School of Electrical and Computer Engineering.

REFERENCES

- [1] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [2] E. Candès and J. Romberg, “Sparsity and incoherence in compressive sampling,” *Inverse Prob.*, vol. 23, no. 3, pp. 969–986, June 2007.
- [3] E. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
- [4] E. Candès, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Comm. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [5] E. Candès and T. Tao, “Near-optimal signal recovery from random projections and universal encoding strategies?,” *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5406–5245, Dec. 2006.
- [6] E. Candès and T. Tao, “The Dantzig selector: Statistical estimation when p is much smaller than n ,” submitted for publication.
- [7] S.S. Chen, D.L. Donoho, and M.A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1999.
- [8] R. Coifman, F. Geshwind, and Y. Meyer, “Noiselets,” *Appl. Comp. Harmon. Anal.*, vol. 10, no. 1, pp. 27–44, 2001.
- [9] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.
- [10] R.A. DeVore, “Nonlinear approximation,” in *Acta Numerica*, vol. 7, pp. 51–150, 1998.
- [11] D.L. Donoho, “Compressed sensing,” *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [12] D.L. Donoho and X. Huo, “Uncertainty principles and ideal atomic decomposition,” *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.
- [13] D.L. Donoho and J. Tanner, “Neighborliness of randomly-projected simplices in high dimensions,” *Proc. Natl. Acad. Sci. USA*, vol. 102, no. 27, pp. 9452–9457, 2005.
- [14] D.L. Donoho, M. Vetterli, R.A. DeVore, and I. Daubechies, “Data compression and harmonic analysis,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 2435–2476, Oct. 1998.
- [15] J. Lee, A. Bandyopadhyay, I.F. Baskaya, R. Robucci, and P. Hasler, “Image processing system using a programmable transform imager,” in *Proc. IEEE ICASSP*, vol. 5, Philadelphia, PA, Mar. 2005, pp. 101–104.
- [16] S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed. San Diego, CA: Academic, 1999.
- [17] B.K. Natarajan, “Sparse approximate solutions to linear systems,” *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, 1995.
- [18] W.B. Pennbaker and J.L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Springer-Verlag, 1993.
- [19] R. Robucci, L.K. Chiu, J. Gray, J. Romberg, P. Hasler, and D. Anderson, “Compressive sensing on a CMOS separable transform image sensor,” submitted for publication.
- [20] M. Rudelson and R. Vershynin, “On sparse reconstruction from Fourier and Gaussian measurements,” submitted for publication.
- [21] J. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.
- [22] A. Skodras, C. Christopoulos, and T. Ebrahimi, “The JPEG2000 still image compression standard,” *IEEE Signal Processing Mag.*, vol. 18, pp. 36–58, Sept. 2001.
- [23] D. Takhar, J.N. Laska, M.B. Wakin, M.F. Duarte, D. Baron, S. Sarvotham, K.F. Kelly, and R.G. Baraniuk, “A new compressive imaging camera architecture using optical-domain compression,” in *Proc. SPIE Conf. Computational Imaging IV*, San Jose, CA, Jan. 2006, pp. 43–52.
- [24] J. Tropp, “Just relax: Convex programming methods for identifying sparse signals in noise,” *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 1030–1051, 2006.
- [25] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

SP