ECE 6123
Advanced Signal Processing: Compressive Sensing and Sparseness

Peter Willett

Fall 2017

1 Sparse Representations

Why do image processors transform an image – via multi-resolution (wavelet) transform, discrete Fourier transform (2D-DFT) or its variant the discrete cosine transform (DCT) – prior to coding it for data compaction? It seems fairly intuitive: while in the original (image) domain the energy is distributed evenly amongst all pixels, in the transform domain this is no longer true. For example, it is common to find most of the energy in low spatial frequency components (larger image objects) and much less at higher frequencies (fine detail); and it makes sense to expend many bits to quantize the former and rather fewer to deal with the latter. In fact, *inverse water-filling* from rate-distortion arguments in information theory tell us to do exactly that, and even to ignore completely (no bits at all) components that are smaller than some threshold.

Taken to its limit, this describes a representation that is *sparse*. We may wish to write

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{e} \tag{1}$$

in which $\mathbf{x}$ is the observation vector[1] of dimension $M \times 1$, $\mathbf{A}$ is an a-priori fixed "dictionary" matrix[2] of dimension $M \times N$ (generally $M \ll N$), $\mathbf{s}$ is a vector of dimension $N \times 1$ that contains only S ($S \ll M$) non-zero elements and $\mathbf{e}$ is a small "noise" vector to account for the inaccuracy in the representation. This is clearly quite appealing: to code (approximately, anyway) the data vector $\mathbf{x}$ all we need is a few (S) elements of $\mathbf{s}$, since presumably the *de*-coder already knows $\mathbf{A}$.

[1] This is not quite a correct thing to write in all cases, but wait for Compressive Sensing to go into more detail.

[2] This is an *over-complete* dictionary: there are more columns of $\mathbf{A}$ than should be necessary to span the space, meaning that these columns are necessarily linearly dependent.

So how do we do this? Let's begin by considering the problem

$$\min_{\mathbf{s}} \{\|\mathbf{s}\|_2\} \text{ such that } \|\mathbf{x} - \mathbf{A}\mathbf{s}\|_2 \leq \varepsilon \tag{2}$$

in which $\|\cdot\|_2$ refers to the Euclidean distance (2-norm). With a few Lagrange multipliers and use of the Woodbury formula we have

$$\mathbf{s} \;=\; (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{A}^T\mathbf{x} \tag{3}$$

where

$$\left\|(\mathbf{I} + \lambda^{-1}\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{x}\right\|_2 = \varepsilon \tag{4}$$

solves implicitly for λ. This is complicated and not what we want anyway – the point in showing it is to demonstrate that there will be no special sparsity associated with the problem, since $\mathbf{s}$ will in general be fully populated. A real sparse solution is found from

$$\min_{\mathbf{s}} \{\|\mathbf{s}\|_0\} \text{ such that } \|\mathbf{x} - \mathbf{A}\mathbf{s}\|_2 \leq \varepsilon \tag{5}$$

where the 0-norm is the number of non-zero elements. The problem with this solution is that in general one may need to test each of the 2^N combinations of non-zero elements of $\mathbf{s}$, and that is clearly not an option for computational reasons. Actually you can do OK with a greedy algorithm, that amounts to finding the best column of $\mathbf{A}$, then next-best, and so on: this is called matching-pursuit (MP) and has complexity $\mathcal{O}(MS)$. A variant of MP that works a little better is *orthogonal* matching pursuit (OMP): after a new column of $\mathbf{A}$ is discovered all non-zero coefficients in the set so far discovered are re-computed. This reduces the error somewhat, and the complexity remains approximately the same.

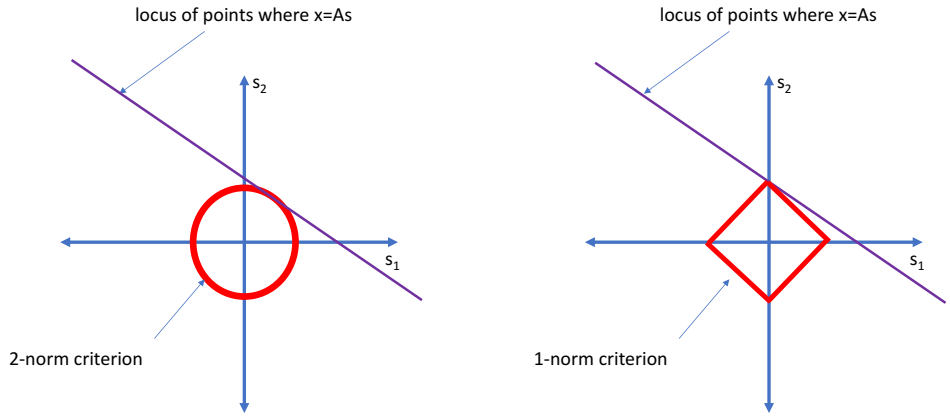
However, researchers have found that an in-between solution is perhaps preferable. Consider the problem

$$\min_{\mathbf{s}} \left\{\|\mathbf{s}\|_p\right\} \text{ such that } \mathbf{x} = \mathbf{A}\mathbf{s} \tag{6}$$

where of course

$$\|\mathbf{s}\|_p \;=\; \left(\sum_{m=1}^{M} |s[m]|^p\right)^{\frac{1}{p}} \tag{7}$$

This assumes that an exact solution (not necessarily sparse) exists, but the notion suggests the following cartoon.



On the left is what we have with $p = 2$ – minimizing the standard Euclidean norm does not encourage a sparse solution. On the right is the L_1 ("Manhattan distance") norm, and it is seen that a sparse solution is indeed the most likely result, since it will be at a "corner" of the constraint set. Actually a reformulation of (6) is what is actually posed:

$$\min_{\mathbf{s}} \left\{ \|\mathbf{x} - \mathbf{A}\mathbf{s}\|_2^2 \right\} \text{ such that } \|\mathbf{s}\|_1 \leq \varepsilon \tag{8}$$

where

$$\|\mathbf{s}\|_1 = \sum_{m=1}^{M} |s[m]| \tag{9}$$

Problem (8) is solved iteratively by a classical subgradient technique from statistics, and is called Least Absolute Shrinkage and Selection Operator (LASSO). And there seems to be some success with (6) using $p = 0.5$.

2 Compressive Sensing

Consider a communications application in which the channel is being probed. The channel is made up of multiple paths, such that

$$h[n] = \sum_{k=1}^{K} \alpha_k \delta[n - n_k] \tag{10}$$

and it is assumed here that the sampling rate is high enough that Nyquist rate sampling can incorporate all possible delays – hence we use discrete time to represent it. Naturally, you want to characterize the channel. One

approach is simply to measure $h[n]$ – perhaps by inserting a very narrow pulse – and to look for peaks. One may need a lot of samples.

Another approach would be to apply a sinusoid of frequency ω_1; one observes

$$H(\omega_1) \;=\; \sum_{k=1}^{K} \alpha_k e^{-j\omega_1 t_k} \tag{11}$$

If one applies another sinusoid of frequency ω_2 one observes $H(\omega_2)$; and so on. Notice that all paths $\{\alpha_k, t_k\}$ contribute to all observations; and that one really needs only $L \geq K$ probing frequencies in order to have enough information to characterize the channel. Notice that we can write

$$L \text{ frequencies} \left\{ \begin{pmatrix} H(\omega_1) \\ H(\omega_2) \\ \vdots \\ H(\omega_L) \end{pmatrix} \right. = \tag{12}$$

$$\underbrace{\begin{pmatrix} e^{-j\omega_1 0} & e^{-j\omega_1} & e^{-j\omega_1 2} & \dots & e^{-j\omega_1(N-1)} \\ e^{-j\omega_2 0} & e^{-j\omega_2} & e^{-j\omega_2 2} & \dots & e^{-j\omega_2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ e^{-j\omega_L 0} & e^{-j\omega_L} & e^{-j\omega_L 2} & \dots & e^{-j\omega_L(N-1)} \end{pmatrix}}_{N \text{ samples} = N \text{ possible paths}} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \alpha_1 \\ 0 \\ \vdots \\ 0 \\ \alpha_2 \\ 0 \\ \vdots \end{pmatrix}$$

where the vector on the RHS is clearly sparse: we could write (12) as (1) with $\mathbf{e} = 0$ and the n^{th} column of the dictionary matrix containing the response of a path at time sample n to the various probing frequencies. Notice how many fewer samples (channel-probings) are needed.

At a somewhat more abstract level what we have done is to posit that we have

$$\mathbf{x} \;=\; \mathbf{A}\mathbf{s} \tag{13}$$

with a sparse $\mathbf{s}$, but that in fact we observe

$$\mathbf{y} \;=\; \mathbf{B}\mathbf{x} \tag{14}$$

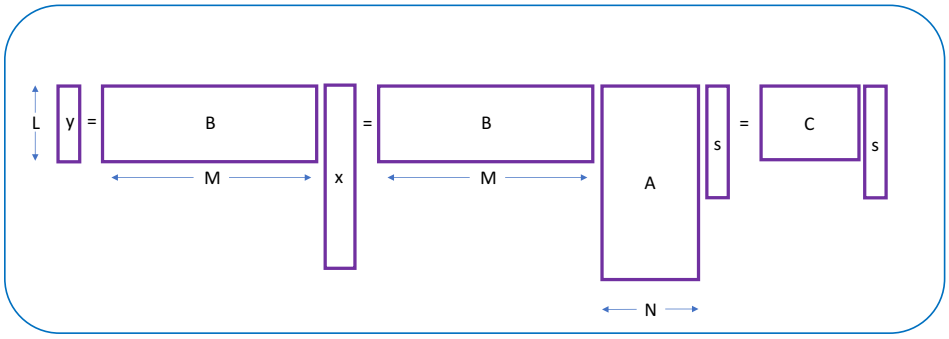
meaning that we have

$$\mathbf{y} \;=\; \mathbf{C}\mathbf{s} \tag{15}$$

where

$$\mathbf{C} = \mathbf{BA} \tag{16}$$

In the channel-probing example just given, $\mathbf{x}$ is the impulse response, which we do not know but would like to know. The *sparse* vector $\mathbf{s}$ is made up of mostly 0's but also the α's in the appropriate locations. These locations are delayed impulses, meaning that the i^{th} column of $\mathbf{A}$ is really just an impulse at the i^{th} delay – that is, it is $\delta[n - n_i]$, where n increments down the column. We do not observe $\mathbf{x}$ directly, of course; instead we observe $\mathbf{Bx}$ which is the response of the impulse response at a particular frequency (see (12)). We observe this at several frequencies; that is, we observe $\mathbf{y}$ which is made up of $H(\omega)$'s.

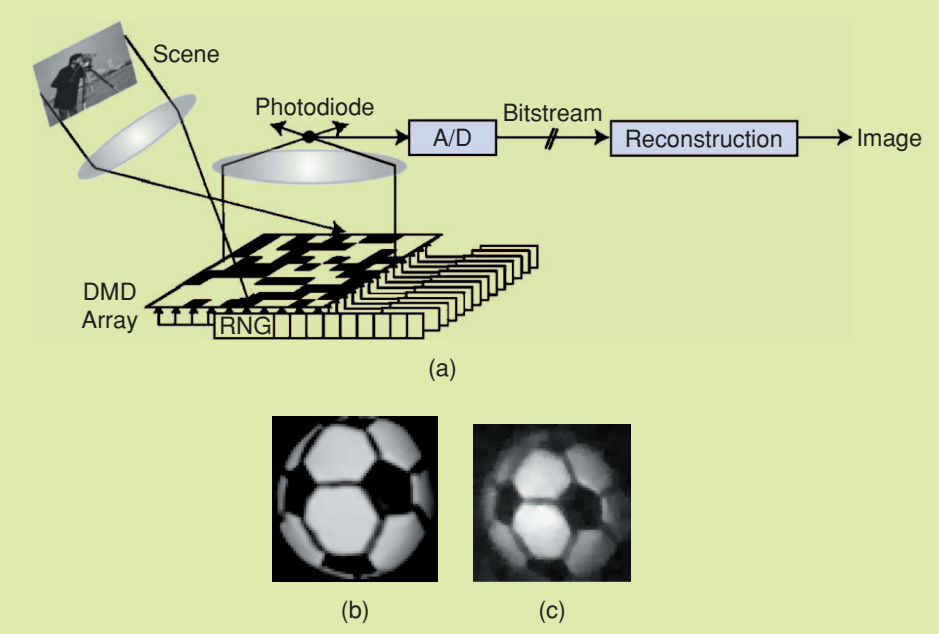


The notional figure is as above. The figure assumes that $N \ll M$ but this need not be true and we could have N quite large – in that case the matrix $\mathbf{C}$ would be "fat."

The above figure leads us to the rather interesting idea of the "single-pixel" camera. In the digital communications notional example the columns of $\mathbf{C}$ – the *dictionary* that we are trying to build our response from – is constructed carefully. That is, each column corresponds to the response that we would observe at the frequencies probed for a specific delay. But indeed much of the more recent success of such compressive sensing has come via columns that are designed haphazardly – using Matlab's *randnormal* function, for example.

Suppose that each row of $\mathbf{B}$ represents a pseudo-random photographic *mask*, and the corresponding element in the observation vector $\mathbf{y}$ is the amount of light received at the single pixel (a photo-diode?) as the true image $\mathbf{x}$ is applied to the mask. The matrix $\mathbf{A}$ can be anything, and often is assumed itself to be random. The after solving the sparse problem $\mathbf{y} = \mathbf{Cs}$ the resultant vector $\mathbf{s}$ is applied to the dictionary matrix $\mathbf{A}$ to render an approximation to the *actual image* $\mathbf{x}$. It seems to work quite well.

Please see the figure below, taken from "Compressive Sensing" by R. Baraniuk, *IEEE Signal Processing Magazine*, pp. 118-124, July 2007.



[FIG3] (a) Single-pixel, compressive sensing camera. (b) Conventional digital camera image of a soccer ball. (c) 64×64 black-and-white image $\widehat{x}$ of the same ball ($N = 4{,}096$ pixels) recovered from $M = 1{,}600$ random measurements taken by the camera in (a). The images in (b) and (c) are not meant to be aligned.

In order that this work we need to make sure that we have the correct sparse vector $\mathbf{s}$ of sparseness S. Suppose that we have $\mathbf{y} = \mathbf{C}\mathbf{s}_1$ and $\mathbf{y} = \mathbf{C}\mathbf{s}_2$ for $\mathbf{d} = (\mathbf{s}_1 - \mathbf{s}_2) \neq \mathbf{0}$. Then we know that $\mathbf{C}\mathbf{d} = \mathbf{0}$ which implies that there is some group of $2S$ columns of $\mathbf{C}$ that are linearly dependent. To avoid this, we must insist that all such collections of $2S$ columns of $\mathbf{C}$ be linearly-independent. This amounts to the (more complicated) *restricted isometry property* (RIP) that gives fairly exact results.