Compressed Sensing

Martin Leslie

Department of Mathematics
University of Arizona

May 13, 2010

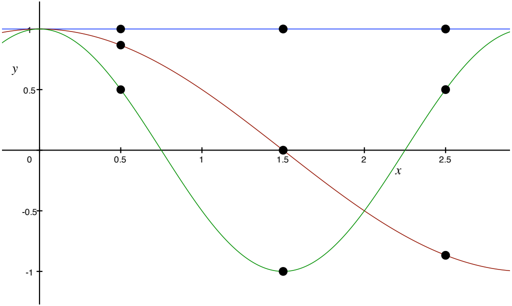
Lossy Compression of images

- Think of a grayscale image as a vector of pixel intensities $x \in \mathbb{R}^m$.
- The idea is to choose $m \times m$ matrix F whose rows are basis for $\mathbb{R}^m$ for which $\theta = Fx$ has many small entries (i.e. θ is approximately sparse).
- Then only need to store the K largest magnitude coefficients θ_i.

Discrete Cosine Transform

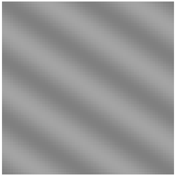
- Sample m cosine functions of different periods at m places to make a basis for $\mathbb{R}^m$. For $m = 3$ we could let

$$F = \begin{pmatrix} 1 & 1 & 1 \\ \sqrt{3}/2 & 0 & -\sqrt{3}/2 \\ 0.5 & -1 & 0.5 \end{pmatrix}$$

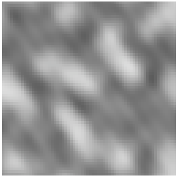


Examples

original $(m = 4096)$



$K = 2$



$K = 100$

$K = 500$

$K = 1000$

$K = 2000$

Why it works

- We can think of

$$\theta_k = \sum_j x_j \cos\left(\frac{\pi k}{m}\left(j + \frac{1}{2}\right)\right)$$

as a discrete version of the real part of the Fourier coefficient

$$a_k = \int_{-\infty}^{\infty} f(t) e^{ikt}\, dt.$$

- The Riemann–Lebesgue Lemma says that $a_k \to 0$ as $k \to \infty$. So eventually most θ_k are small.

Sparsity

- Define

$$\begin{aligned}\|\theta\|_0 &= \lim_{p \to 0} \|\theta\|_p^p \\ &= \lim_{p \to 0} |\theta_1|^p + \ldots + |\theta_m|^p \\ &= \#\{\text{non-zero elements of } \theta\}\end{aligned}$$

- So $\|\theta\|_p$ is a measure of sparsity for small p. We will use $\|\theta\|_0$ and $\|\theta\|_1$. We'll say θ is k–sparse if $\|\theta\|_0 \leq k$.

Compressed Sensing

- Instead of taking in all m coefficients from our camera and then compressing them down to n numbers, couldn't we only take n coefficients in to start with?

Compressed Sensing

- Instead of taking in all m coefficients from our camera and then compressing them down to n numbers, couldn't we only take n coefficients in to start with?
- The idea: Let H be a $n \times m$ matrix whose entries are normal random numbers chosen such that columns have expected ℓ^2 norm 1.
- Now $y = H\theta = HFx$ is in $\mathbb{R}^n$.
- Physically HF corresponds to n measurements while F corresponds to m measurements.

Recovering x from $y = HFx$

- We are trying to solve $H\theta = y$ for H an $n \times m$ matrix with $n << m$. (Then $x = F^{-1}(\theta)$.)
- This is a massively under-determined system of linear equations (many more variables than equations) so has a lot of solutions.
- To determine the correct one we need to use the fact that θ is sparse.

Is this even possible?

Yes. For example: assume H has the property that every set of $2k$ columns from it are linearly independent. Then any k–sparse vector x can be reconstructed uniquely from $y = Hx$.

Proof.

If x, x' are both k–sparse then $x - x'$ is $2k$–sparse. But since $Hx = y$ and $Hx' = y$ we have $H(x - x') = 0$. Then since the columns of H corresponding to non-zero elements of $x - x'$ are linearly independent we have $x - x' = 0$. □

Basis Pursuit

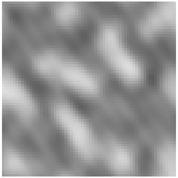
- First idea is to find sparsest θ (i.e. minimum $\|\theta\|_0$) such $H\theta = y$. Unfortunately this problem is computationally infeasible.
- Instead we find θ with minimum $\|\theta\|_1$ such that $H\theta = y$. This type of problem is efficiently solvable by a linear programming technique called basis pursuit.

Theoretical basis for this idea

- Candes and Tao (2004) showed that this is a good idea.
- The paper shows that if all sets of k columns of H are sufficiently close to being orthogonal then the answer from basis pursuit is the original θ.
- They also showed that a randomly chosen H as described earlier will satisfy the above condition with overwhelming probability.
- The big result in our context is that using compressed sensing we can use on the order of $K \log(m/K)$ measurements to reconstruct an image with same quality as choosing the K largest magnitude coefficients.

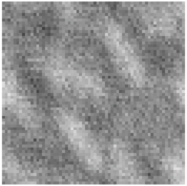
Pictures

original $(m = 4096)$

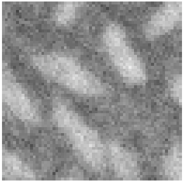


$K = 100$

$K = 500$



$n = 500$



$n = 1000$

$n = 2000$

Single Pixel Camera

- A digital camera with only one pixel has been developed at Rice.
- It implements Compressed Sensing by using random filters to make measurements of intensity of light through them. Then reconstructs.

Original

$m = 4096$, $n = 800$

Connection to LDPC codes

- Low Density Parity Check (LDPC) codes are a form of error correcting codes that approach capacity and have efficient message passing decoding algorithms.
- LDPC codes can also be decoded using linear programming techniques.
- Dimakis & Vontobel (2009) showed that H matrices over $\mathbb{F}_2$ that are good for LDPC codes are good for compressed sensing considering H as a $\{0, 1\}$ matrix over $\mathbb{R}$.

Interesting questions

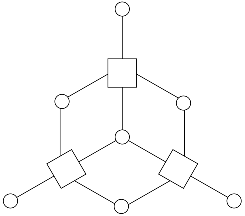
- Can you find a deterministic H that works?
- Is there a message passing algorithm that works almost as well as basis pursuit but is faster?

Message passing algorithm

- Think of our $\{0,1\}$ matrix H as the adjacency matrix of a bipartite graph. For example :

$$H = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

looks like



- The squares represent y_j's and the circles represent θ_i's.

One idea (Berinde, Indyk, Ruzic)

- First the squares send circles received measurements:

$$m_{j \to i} = y_j.$$

- Circles send to a square what the circle thinks it represents:

$$m_{i \to j} = \text{median}_{j'}(m_{j' \to i}).$$

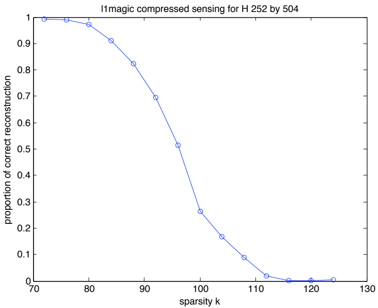
- Squares send circles what that circle needs to be to make the square have correct value:

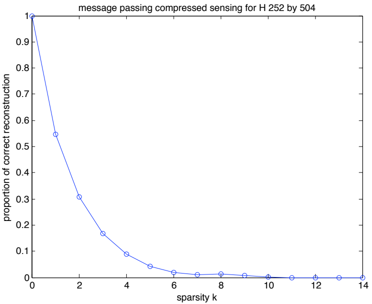
$$m_{j \to i} = y_j - \sum_{i' \neq i} m_{i' \to j}.$$

- Repeat last two steps until the circles make a vector which satisfies $H\theta = y$.

How well does it work?

- Asymptotically this require number of measurements of the same order as basis pursuit while running faster. Practically it doesn't work so well:





An idea for nonnegative signals (Chandar, Shah, Wornell)

- Send messages that represent the minimum, m, and maximum, M, values:

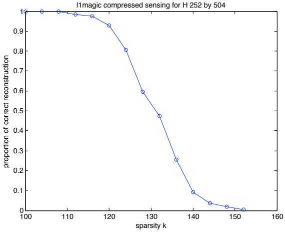
$$m_{j\to i} = 0 \qquad M_{j\to i} = y_j$$

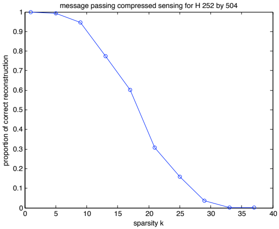
$$m_{i\to j} = \max_{j'}(m_{j'\to i}) \qquad M_{i\to j} = \min_{j'}(M_{j'\to i})$$

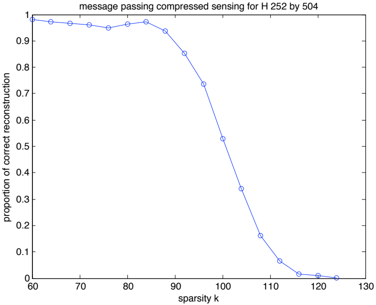
$$m_{j\to i} = y_j - \sum_{i'\neq i} M_{i'\to j} \qquad M_{j\to i} = y_j - \sum_{i'\neq i} m_{i'\to j}.$$

- Repeat last two steps until messages converge and $m_{i\to j} = M_{i\to j}$.

How well does it work?







References

- Decoding by Linear Programming, Emmanuel Candes, Terence Tao, http://arxiv.org/abs/math/0502327
- The single pixel camera website is http://dsp.rice.edu/cscamera
- LP Decoding meets LP Decoding: A Connection between Channel Coding and Compressed Sensing, Alexandros G. Dimakis, Pascal O. Vontobel, http://arxiv.org/abs/0910.1121
- Practical Near-optimal Sparse Recovery in the L1 Norm, R. Berinde, P. Indyk and M. Ruzic, http://people.csail.mit.edu/indyk/smp.pdf
- A Simple Message-Passing Algorithm for Compressed Sensing, Venkat Chandar, Devavrat Shah, Gregory W. Wornell, http://arxiv.org/abs/1001.4110