CS 395T: Sublinear Algorithms Fall 2016

Lecture 14 — October 13, 2016

Prof. Eric Price *Scribes: Cody Freitag, William Swartworth*

1 Compressed Sensing

In compressed sensing, the overall goal is to observe an object using as few measurements as possible. In today's lecture, we'll motivate compressed sensing with an application to observing images. Then we'll give the precise setup for compressed sensing. Lastly we'll give an interative algorithm to solve compressed sensing.

1.1 Single-Pixel Cameras

We will consider taking a picture as a motivating example. A normal camera uses 1000×1000 pixels with 1 photosensitive element per pixel. This is typically cheap for normal images since we can use silicon. However, silicon doesn't work if you want to capture infrared images, making it really expensive. As a result, we may try to decrease the number of photosensitive elements needed at the expense of taking a greater number of measurements per element.

An image can be represented as a vector $x = (x_1, \ldots, x_n)$ where entries of x correspond to individual pixels. Instead of getting a measurement for each pixel, we could instead get $\sum_i b_i x_i = \langle b, x \rangle$ for some $b \in \{0,1\}^n$ (where we think of the b_i's as lens filters on the camera). We could do this many times with different filters and about $\approx n/2$ ones per filter b. Then with around n different b, we can reconstruct x just by inverting the b matrix. So we would only need one photosensitive element, but we'd need n measurements now.

This leads to the fundamental problem in compressed sensing: choose a matrix A such that observing Ax allows x to be recovered.

1.2 Questions to Consider

Do we really need n measurements?

The answer is yes if we require exact recovery of every possible x. Many situations are more relaxed than this. For example it might be possible to recover x approximately from Ax, even if Ax is much smaller than x, if x is known to have some structure. In fact, if x is an image, then we expect to be able to compress it anyway (using JPEG for example). So it's reasonable to hope that Ax could be substantially smaller than x.

How do we model compressibility?

We can often think of compressible data as being sparse with respect to some basis. This is a familiar idea in the context of lossy compression. To compress an image we start by changing

basis—a typical choice is the wavelet basis. If we've chosen a good basis, then many entries will be negligibly small, and we approximate our image as a sparse vector in our chosen basis.

1.3 Compressed Sensing - Specifying the Problem

We begin by choosing $A \in \mathbb{R}^{m \times n}$ where generally m is much smaller than n.

Making our measurements gives us some vector $y = Ax + e$, where we assume that $||x||_0 \leq k$ (where the 0-norm is defined to be the sparsity). The vector e should be thought of as an error term which models imprecise measuring equipment.

Our goal is to recover x from y. In other words, we would like to find $\widehat{x} \approx x$ given y (and knowledge of the A that we chose).

This can be summarized as follows:

Choose $A \in \mathbb{R}^{m \times n}$
Observe $y = Ax + e$, where $||x||_0 \leq k$ and e is noise
Compute $\widehat{x} \approx x$ from (A, y)

2 Algorithms

Many algorithms exist for compressed sensing:

- Convex Optimization
  - L1 minimization, LASSO, Dantzig selector
- Iterative methods
  - CoSAMP, AMP, OMP, IHT

Many of these algorithms use a matrix A satisfying RIP($O(k)$, 0.1) to find $\widehat{x}$ with error $||\widehat{x} - x||_2 \lesssim ||e||_2$. Fortunately we can construct RIP($O(k)$, 0.1) in many ways:

- random gaussian with $m = O(k \log(n/k))$
- any JL matrix for $\delta = 2^{-O(k \log(n/k))}$
- matrices with low coherence [columns are $a_1, \ldots, a_n$ and $\frac{|\langle a_i, a_j \rangle|}{\sqrt{||a_i|| \cdot ||a_j||}} < \frac{1}{k}$ for all i, j. This is easy to check but requires $m > k^2$.]
- random rows of Fourier, $O(k \log n \log^2 k)$
- explicity $k^{2-\epsilon}$ row matrices exist.

We'll look at IHT (Iterative Hard Thresholding) today.

2.1 IHT

First suppose that A satisfies the $(3k, \epsilon)$-RIP.

This means that for all $3k$-sparse x in $\mathbb{R}^n$ we have

$$\begin{aligned} ||Ax||^2 &= (1+\epsilon)||x||_2^2 \\ x^T A^T A x &= (1+\epsilon) x^T x \\ |x^T (A^T A - I) x| &\leq \epsilon x^T x. \end{aligned}$$

In terms of the operator norm,

$$||(A^T A - I)_{S \times S}|| \leq \epsilon, \forall S \subset [n], |S| \leq 3k.$$

So the RIP implies that $A^T A$ is approximately the identity on small sets of coordinates.

In the compressed sensing problem, we're given $Y = Ax + e$, where x is $3k$-sparse, and $||e||$ is "small".

Let $z = A^T y = A^T Ax + A^T e$. Then $A^T e$ should be small noise, and $A^T Ax$ should be approximately x as long as x is sparse. So we expect that $||z - x||$ is small.

To be more precise, let S be any set of size at most $3k$, which we assume contains the support of x. Then we have

$$\begin{aligned} ||(z-x)_S|| &\leq ||(A^T A - I)_{S \times S} \cdot x_S|| + ||(A^T e)_S|| \\ &\leq \epsilon \cdot ||x|| + ||A^T_{S \times [m]}|| \cdot ||e||. \end{aligned}$$

The bound on $||(A^T e)_S||$ was shown in the next class.

Lemma 2.1. *Let $x, z \in \mathbb{R}^n$, with x k-sparse and with support H. Let S be the set of indices corresponding to the top k elements of z. Then $||x - z_S||_2^2 \leq 5 \cdot ||(x-z)_{H \cup S}||_2^2$.*

Proof. Pair up $i \in H \backslash S$ and $j \in S \backslash H$. We know $|z_j| > |z_i|$, so we just need $x_i^2 + z_j^2 \leq 5 \cdot ((x - z_i^2) + z_j^2)$. There are two cases. Either $|z_i| > |x_i|/2$, which implies $x_i^2 + z_j^2 \leq 4 z_i^2 + z_j^2 \leq 5 z_j^2$. Or, $|z_i| \leq |x_i|/2$, which implies $x_i^2 + z_j^2 \leq 4(x_i - z_i)^2 + z_j^2$. □

Now for $z = A^T y = A^T (Ax) + A^T e$, let T denote the top k elements of z. Applying the lemma to $T \cup \text{supp}(x)$ we get

$$\begin{aligned} ||x - z_T||_2 &\leq \sqrt{5} \cdot ||(x-z)_{T \cup S}||_2 \\ &\leq \sqrt{5}(\epsilon ||x|| + 3||e||). \end{aligned}$$

For $\epsilon = 0.1$, this is at most $||x||/4 + 3||e||$.

Now let $y' = A(x - z_T) + e = y' - A z_T$.

The same analysis as before says $z' = A^T y' = A^T A(x - z_T) + a^T e$ has

$$\begin{aligned} ||(z' - (x - z_T)) + S|| &= ||(x - (z_T + z'))_S|| \\ &\leq \epsilon||x - z_T|| + (1 + \epsilon)||e||. \end{aligned}$$

Set $x^{(2)}$ to be the top k elements of $z_T + z'$. According to the lemma,

$$\begin{aligned} ||x - x^{(2)}|| &\leq \sqrt{5}||x - (z_T + z')_{\text{supp}(x^{(2)}) \cup \text{supp}(x)}|| \\ &\leq \sqrt{5}\epsilon||x - z_T|| + (1 + \epsilon)||e|| \\ &\leq 1/4 \cdot (||x - x^{(1)}|| + 3||e||). \end{aligned}$$

Now we just repeat to a get a sequence of successively better recoveries of x. So set

$$z^{(t+1)} = A^T(y - Ax^{(t)}),$$

and $x^{(t+1)} = $ top k entries of $z^{(t+1)}$. Our analysis gives us the following:

$$||x^{(t+1)} - x||_{\leq} \frac{1}{4}||x^{(t)} - x|| + 3||e||.$$

Therefore if $||x^{(t+1)} - x|| > 12||e||$, then $||x^{(t+1)} - x|| \leq \frac{1}{2}||x^{(t)} - x||$. Hence, after $I = \log(||x||/||e||)$ iterations, we get a recovery $\widehat{x} = x^{(I)}$ with $||\widehat{x} - x|| \leq 12||e||$.

The running time of our algorithm is

$$O\left(\log \frac{||x||}{||e||} \cdot \text{matrix/vector multiply}\right).$$