Special Topics in Theoretical Computer Science February 21, 2019

Lecture 12: Fast Algorithm for Compressive Sensing

Lecturer: Huy Lê Nguyễn *Scribe: Xuangui Huang*

Last time we saw compressive sensing (noisy sparse vector recovery) using ℓ_1 minimization, named as "Basis Pursuit." The goal is to recover a hidden vector x that is k-sparse with noise, after observing measurements $y = \Pi x$. What we did is to solve the following linear programming:

$$\begin{gathered} \min \|z\|_1 \\ \text{subject to } \Pi z = y \ . \end{gathered}$$

More generally, if there is a "post-measurement noise" e s.t. $\|e\|_2 \leq \alpha$ and $y = \Pi x + e$, we can change the above LP into the following convex optimization:

$$\begin{gathered} \min \|z\|_1 \\ \text{subject to } \|\Pi z - y\|_2 \leq \alpha \ . \end{gathered}$$

Similar to what we did last time, we can prove that this convex optimization can successfully recover x if Π is RIP. Recall that we used x^k to denote the vector obtained from picking top k coordinates of x and zeroing out the rest.

Theorem 1. *If Π satisfies $(\varepsilon, 2k)$-RIP for sufficiently small $\varepsilon > 0$, we can get z s.t.*

$$\|z - x\|_2 \leq O\left(\frac{1}{\sqrt{k}} \left\|x - x^k\right\|_1 + \|e\|_2\right) . \tag{1}$$

However, algorithms for convex optimization, such as interior-point method and ellipsoid method, are usually very slow so it is impractical to use them to recover a vector x that has millions of entries.

In this lecture we introduce a fast algorithm for compressive sensing: the "Iterative Hard Thresholding" (IHT) algorithm by Blumansath and Davies[1], based on the CoSaMP algorithm by Needell and Tropp[2]. The algorithm is very simple:

Algorithm 1 Iterative Hard Thresholding

$x^{(1)} \leftarrow 0$
for t **from** 1 **to** T **do**
  $a^{(t+1)} \leftarrow x^{(t)} + \Pi^\top \left(y - \Pi x^{(t)}\right)$
  $x^{(t+1)} \leftarrow H_k\left(a^{(t+1)}\right)$
end for

The H_k operator above is the "hard thresholding operator," which keeps the top k coordinates (in absolute value) of the operand and zeros out the rest. This algorithm uses this operator iteratively, thus earning the name IHT. Observed that $x^{(t)}$'s are guaranteed to be k-sparse, we can prove the "correctness" of this algorithm by the following theorem.

Theorem 2. *If* Π *satisfies* $(\varepsilon, 3k)$*-RIP for* $\varepsilon \le \frac{1}{4\sqrt{2}}$*, then for any* $t \ge 1$*,*

$$\left\|x^{(t+1)} - x\right\|_2 \le O\left(2^{-t}\left\|x\right\|_2 + \left\|x - x^k\right\|_2 + \frac{1}{\sqrt{k}}\left\|x - x^k\right\|_1 + \left\|e\right\|_2\right). \tag{2}$$

The last two terms $\frac{1}{\sqrt{k}}\left\|x - x^k\right\|_1$ and $\left\|e\right\|_2$ also appear in Equation (1). In the following we will see that the second term $\left\|x - x^k\right\|_2$ is bounded by the third term. To make the first term $2^{-t}\left\|x\right\|_2$ small enough, we need the total number of iterations T to be $\log\left\|x\right\|_2$, roughly the total number of bits of all the numbers in the input, which the runtime of the convex optimization method also depends on (polynomially if we use ellipsoid method). The only difference is that here the dependence of runtime on T is explicit and linear whilst for the convex optimization it is implicit and polynomial. Therefore HIT algorithm can not only outperform the convex optimization in time but also achieve similar accuracy of solution.

Claim 1. $\left\|x - x^{2k}\right\|_2 \le \frac{1}{\sqrt{k}}\left\|x - x^k\right\|_1$.

Proof. We use the "Shelling trick," which was already used in previous lectures for several times. We sort coordinates of x in descending order of absolute value, obtaining $x_1, x_2, \ldots, x_n$ s.t. $|x_1| \ge |x_2| \ge \cdots \ge |x_n|$. Then we divide them into groups of size k, denoted as $B_1, B_2, \ldots$. For any $j \in B_t$ we have $|x_j| \le \min_{i \in B_{t-1}}\{|x_i|\} \le \frac{\left\|x_{B_{t-1}}\right\|_1}{k}$, thus $\left\|x_{B_t}\right\|_2 \le \frac{1}{\sqrt{k}}\left\|x_{B_{t-1}}\right\|_1$, therefore by triangle inequality we have

$$\left\|x - x^{2k}\right\|_2 \le \sum_{t \ge 3}\left\|x_{B_t}\right\|_2 \le \sum_{t \ge 2}\frac{1}{\sqrt{k}}\left\|x_{B_t}\right\|_1 = \frac{1}{\sqrt{k}}\left\|x - x^k\right\|_1. \tag{3}$$

□

Looking back into $y = \Pi x + e$, we can instead write it as $y = \Pi x^{2k} + \Pi\left(x - x^{2k}\right) + e = \Pi x^{2k} + e'$ if we define $e' = \Pi\left(x - x^{2k}\right) + e$. In other words, we can take the noise out of x and put it into the post-measurement noise.

Claim 2. $\left\|e'\right\|_2 \le \left\|e\right\|_2 + \frac{1+\varepsilon}{\sqrt{k}}\left\|x - x^{2k}\right\|_1$.

Therefore, to prove Theorem 2, it is sufficient to prove that for $2k$-sparse x,

$$\left\|x^{(t+1)} - x\right\|_2 \le O\left(2^{-t}\left\|x\right\|_2 + \left\|e'\right\|_2\right). \tag{4}$$

Proof of Claim 2.

$$\begin{aligned}
\|e'\|_2 &\le \|e\|_2 + \left\|\Pi(x - x^{2k})\right\|_2 && \text{(by triangle inequality)}\\
&\le \|e\|_2 + \sum_{t\ge 3} \left\|\Pi x_{B_t}\right\|_2 && \text{(by triangle inequality)}\\
&\le \|e\|_2 + \sum_{t\ge 3} (1+\varepsilon) \left\|x_{B_t}\right\|_2 && \text{(by RIP and k-sparsity of x_{B_t})}\\
&\le \|e\|_2 + (1+\varepsilon)\frac{1}{\sqrt{k}} \left\|x - x^k\right\|_1 . && \text{(similar to Eq. (3))}
\end{aligned}$$

$\square$

Now we want to prove Eq. (4) for sparse x. To make life easier we use k as $2k$ and e as e'. Intuitively, in each iteration of the IHT algorithm, we have

$$\begin{aligned}
x^{(t+1)} &= H_k\big(a^{(t+1)}\big)\\
&= H_k\Big(x^{(t)} + \Pi^\top \left(\Pi x + e - \Pi x^{(t)}\right)\Big)\\
&\approx H_k\Big(x^{(t)} + I\left(x - x^{(t)}\right) + \Pi^\top e\Big) && \text{(by RIP)}\\
&\approx x + e.
\end{aligned}$$

Denote $\Gamma^* = \mathsf{supp}(x)$, $\Gamma^{(t)} = \mathsf{supp}\big(x^{(t)}\big)$, $B^{(t)} = \Gamma^* \cup \Gamma^{(t)}$, and $r^{(t)} = x - x^{(t)}$. Now we have

$$\begin{aligned}
\left\|r^{(t+1)}\right\|_2 &= \left\|x - x^{(t+1)}\right\|_2\\
&= \left\|x_{B^{(t+1)}} - x^{(t+1)}_{B^{(t+1)}}\right\|_2 && \text{(by def. of $B^{(t+1)}$)}\\
&\le \left\|x_{B^{(t+1)}} - a^{(t+1)}_{B^{(t+1)}}\right\|_2 + \left\|a^{(t+1)}_{B^{(t+1)}} - x^{(t+1)}_{B^{(t+1)}}\right\|_2 && \text{(by triangle inequality)}\\
&\le 2\left\|x_{B^{(t+1)}} - a^{(t+1)}_{B^{(t+1)}}\right\|_2 && \text{(by optimality of H_k)}\\
&= 2\left\|x_{B^{(t+1)}} - x^{(t)}_{B^{(t+1)}} - \Pi^\top_{B^{(t+1)}}\left(y - \Pi x^{(t)}\right)\right\|_2\\
& && \text{(where Π_B is Π but with columns not in B zeroed out)}\\
&= 2\left\|r^{(t)}_{B^{(t+1)}} - \Pi^\top_{B^{(t+1)}}\left(\Pi r^{(t)} + e\right)\right\|_2\\
&= 2\left\|r^{(t)}_{B^{(t+1)}} - \Pi^\top_{B^{(t+1)}}\Pi r^{(t)}_{B^{(t+1)}} - \Pi^\top_{B^{(t+1)}}\Pi r^{(t)}_{\overline{B^{(t+1)}}} - \Pi^\top_{B^{(t+1)}} e\right\|_2\\
&\le 2\left\|\left(I - \Pi^\top_{B^{(t+1)}}\Pi_{B^{(t+1)}}\right) r^{(t)}_{B^{(t+1)}}\right\|_2 && \text{(since $\Pi r^{(t)}_{B^{(t+1)}} = \Pi_{B^{(t+1)}} r^{(t)}_{B^{(t+1)}}$)}\\
&\quad + 2\left\|\Pi^\top_{B^{(t+1)}}\Pi_{B^{(t)}\setminus B^{(t+1)}} r^{(t)}_{B^{(t)}\setminus B^{(t+1)}}\right\|_2 && \text{(similarly, by $r^{(t)}_{\overline{B^{(t+1)}}} = r^{(t)}_{B^{(t)}\setminus B^{(t+1)}}$)}\\
&\quad + 2\left\|\Pi^\top_{B^{(t+1)}} e_{B^{(t+1)}}\right\|_2 .
\end{aligned}$$

Claim 3. $\left\|\left(I - \Pi_{B^{(t+1)}}^{\top} \Pi_{B^{(t+1)}}\right) r_{B^{(t+1)}}^{(t)}\right\|_2 \le \varepsilon \left\|r_{B^{(t+1)}}^{(t)}\right\|_2$.

Proof. For any vector v with $\mathsf{supp}(v) \subseteq B^{(t+1)}$, we have

$$\begin{aligned}\left|\left\langle v, \left(I - \Pi_{B^{(t+1)}}^{\top} \Pi_{B^{(t+1)}}\right) r_{B^{(t+1)}}^{(t)}\right\rangle\right| &= \left|\left\langle v, r_{B^{(t+1)}}^{(t)}\right\rangle - \left\langle \Pi_{B^{(t+1)}} v, \Pi_{B^{(t+1)}} r_{B^{(t+1)}}^{(t)}\right\rangle\right| \\ &\le \varepsilon \left\|v\right\|_2 \left\|r_{B^{(t+1)}}^{(t)}\right\|_2,\end{aligned}$$

by Cauchy-Schwarz inequality and RIP. Take $v = \left(I - \Pi_{B^{(t+1)}}^{\top} \Pi_{B^{(t+1)}}\right) r_{B^{(t+1)}}^{(t)}$. □

We will finish our proof in next lecture.

References

[1] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

[2] Deanna Needell and Joel A Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and computational harmonic analysis*, 26(3):301–321, 2009.