COMS E6998: Algorithms for Massive Data (Spring'25) Sep 24, 2025

Lecture 7: Iterative Hard Thresholding

Instructor: *Alex Andoni* Scribe: *Xian Jiang*

1 Review

Recall that in compressed sensing we have a vector $x \in \mathbb{R}^n$ that is approximately (at most) k sparse, in the sense that the error

$$\text{Err}_1^k(x) \triangleq \min_{x' \text{ is } k\text{-sparse}} \|x - x'\|_1$$

is small. A is an $m \times n$ matrix of measurements A. Given A and $y = Ax$, we can recover a vector $x^\star$ from $y = Ax$ such that

$$\|x - x^\star\|_2 \leq O(1) \cdot \min_{x' \text{ is } k\text{-sparse}} \|x - x'\|_2 \triangleq Err_2^k(x)$$

We can do $x^\star = L_1(y)$, using the following L_1 algorithm.

L_1 algorithm:

$$\text{Minimize } \|x^*\|_1$$

$$\text{such that } Ax^* = y$$

As long as A RIP for some parameters [Donoho; Candès-Romberg-Tao], it would be sufficient to achieve guarantees of the type eq. (1) with norm 1 for this algorithm. The L_1 algorithm runs in polynomial time since it is a linear program. Is there a faster algorithm with similar guarantees in less run time?

2 Iterative Hard Thresholding (IHT)

The Iterative Hard Thresholding algorithm allows us to achieve similar guarantees with less run time. Here we give a brief overview of the algorithm.

Algorithm 1 Iterative Hard Thresholding (IHT).

function IHT$(A, y(= Ax), k, T)$
  $x^1 \leftarrow (0, ..., 0)$
  for $t = 1 \cdots T$ **do**
    $x^{t+1} = P_k\big(x^t + A^\top(y - Ax^t)\big)$
  end for
  return x^{T+1}
end function

Explanation of algorithm: The previous guess x^t is probably not good enough. If we do not get y from Ax^t, then we want to subtract by the amount that does not give the measurement $(y - Ax^t)$, but this is in the wrong space (it is in the space of measurements). We project it back to the space of signals using $A^\top$, which is $n \times m$. Since we want our x^{t+1} to be k-sparse, we can use a projection operator $P_k : \mathbb{R}^n \to \mathbb{R}_n$ that takes the vector and leaves only k largest entries.

Theorem 1 (Blumensath, Davies '09)**.** *If A is $(3k, \varepsilon = \frac{1}{8})$-RIP, suppose that $y = Ax + e$, where e is some error (we cannot use the previous result because it requires exact $y = Ax$), then x^{T+1} satisfies*

$$\|x^{T+1} - x\|_2 \leq O(1)\left[2^{-T}\|x\|_2 + \frac{\mathrm{Err}_1^k(x)}{\sqrt{k}} + \|e\|_2\right].$$

The guarantee above is called an L2/L1-guarantee, and it is stronger than the L1/L1-guarantee from the previous lecture.

The error that we get decreases exponentially with the number of iterations.

We will prove a slightly weaker theorem in class:

Theorem 2. *Let $x \in \mathbb{R}^n$ be k-sparse and A is $(3k, \varepsilon = \frac{1}{12})$-RIP, then suppose that $y = Ax$, then*

$$\|x^{T+1} - x\|_2 \leq 2^{-T}\|x\|_2,$$

where x^{T+1} is the output of the IHT algorithm.

Intuition: First observe that $\|Ax^t\|_2^2 \approx \|x^t\|_2^2 \in (1 \pm \varepsilon)\|x^t\|_2^2$ because A is $(3k, \varepsilon = \frac{1}{12})$-RIP. Formally, we have

$$(x^t)^\top A^\top A x^t \in (1 \pm \varepsilon)(x^t)^\top I x^t$$

which implies

$$|(x^t)^\top (I - A^\top A x^t)| \leq \varepsilon \|x^t\|_2^2.$$

If A is a very good RIP matrix, then as $\varepsilon \to 0$, $A^\top A \approx I$ on vector $x^t \Rightarrow A^\top A x^t = x^t$.

$$x^{t+1} = P_k(x^t + A^\top(Ax - Ax^t)) = P_k(x^t + A^\top A(x - x^t)) \approx P_k(x^t + (x - x^t)) = x,$$

Proof. We define $r^t := x - x^t$, $a^{t+1} := x^t + A^\top(y - Ax^t)$. We will show that $\|r^{t+1}\|_2 \leq \frac{1}{2}\|r^t\|_2$, which suffices to prove our theorem. The intuition of the proof is that we can write

$$a^{t+1} = x^t + A^\top(y - Ax^t) = x^t + A^\top A(x - x^t),$$

and this is an approximation of x when $A^\top A \approx I$ (in which we would obtain $\approx x^t + I(x - x^t) = x$).

Let $B^t := \mathsf{supp}(x) \cup \mathsf{supp}(x^t) \supseteq \mathsf{supp}(r^t)$ (with $|B^t| \leq 2k$). Denote $B = B^{t+1} = \mathsf{supp}(x) \cup \mathsf{supp}(x^{t+1})$, $B^- := B^t = \mathsf{supp}(x) \cup \mathsf{supp}(x^t)$. We now have

$$\begin{aligned}
\|r^{t+1}\|_2 &= \|x - x^{t+1}\|_2 \\
&= \|x_B - x_B^{t+1}\|_2 \\
&= \|x_B - a_B^{t+1} + a_B^{t+1} - x_B^{t+1}\|_2 \\
&\leq \|x_B - a_B^{t+1}\|_2 + \|a_B^{t+1} - x_B\|_2 \\
&= 2\|x_B - a_B^{t+1}\|_2.
\end{aligned}$$

Let $A_B = A$ with columns not in B zeroed out, we have

$$a_B^{t+1} = (x_B^t + A^\top A r^t)_B = x_B^t + A_B^\top A r^t,$$

which implies

$$\begin{aligned}\|r^{t+1}\|_2 &\le 2\|x_B - x^t - A_B^\top A r^t\|_2 \\ &= 2\|r_B^t - A_B^\top A r^t\|_2 \\ &\le 2\|r_B^t - A_B^\top A_B r_B^t\|_2 + 2\|A_B^\top A r^t_{B^t \setminus B}\|_2\end{aligned}$$

Claim 3. $\|r_B^t - A_B^\top A_B r_B^t\|_2 \le \varepsilon - \|r_B^t\|_2$

Proof. $\forall$ $2k$-sparse $z(:= r_B^t)$,

(reason we need $3k$ RIP is because we require $\mathsf{supp}(x)$, $\mathsf{supp}(x^t)$, and $\mathsf{supp}(x^{t+1})$)

$$\begin{aligned}\|z_B - A_B^\top A_B z_B\|_2 &= \|(I_B - A_B^\top A_B) - z_B\|_2 \\ &\le \|I_B - A_B^\top A\|_2 \cdot \|z_B\|_2 \\ &\le \max_{u \in \mathbb{R}^n,\ \|u\|=1} \left[u_B^\top (I_B - A_B^\top A_B) u_B\right] \cdot \|z_B\|_2 \\ &\le \varepsilon \cdot \|u_B\|_2 \cdot \|z_B\|_2 = \varepsilon \cdot \|z_B\|_2\end{aligned}$$

where the last inequality follows from the RIPness of A. □

Claim 4.

$$\|A_B^\top A_{B^t \setminus B} \cdot r_{B^t \setminus B}\|_2 \le 2\varepsilon \|r_{B^t \setminus B}\|.$$

Overall,

$$\|r^{t+1}\|_2 \le 2(\varepsilon \|r_B^t\|_2 + 2\varepsilon \|r^t_{B^t \setminus B}\|_2) \le 6\varepsilon \|r^t\|_2 \le \tfrac{1}{2}\|r^t\|_2,$$

assuming $\varepsilon \le \frac{1}{12}$.

After t iterations:

$$\|r^{t+1}\|_2 \le 2^{-t}\|r^1\|_2 = 2^{-t}\|x\|_2.$$

□

2.1 Runtime

Runtime of this algorithm is $T \cdot O(nm) = O(Tnk \cdot \lg n)$, where $T = \lg \frac{\|x\|_2}{\delta}$ if we want $\|r^{T+1}\|_2 \le \delta$. Can we get $RT \ll n$ (sublinear regime)? With our current approach, this is not possible, but with a structured A, this is possible in time $k \cdot (\log n)^{O(1)}$ (for slightly different recovery guarantees).