CS 395T: Sublinear Algorithms	Fall 2016

Lecture 15 — Oct. 18, 2016

Prof. Eric Price — *Scribe: Ziyang Tang & Mei Wang*

1 Overview

In last class, we discussed Compressed Sensing with Interactive Hard Thresholding (IHT). In this lecture, we continue to study IHT algorithm first and then we start to look at "Model-Based" Compressed Sensing.

2 Recall for IHT Algorithm

Definition 1. *Suppose matrix $A \in \mathbb{R}^{m\times n}$ has $(k,\epsilon)RIP$. If for all k-sparse x, $\|Ax\|_2^2 = (1\pm\epsilon)\|x\|_2^2$. Then this is equivalent to saying as $\|(A^\top A - I)_{S*S}\| \leq \epsilon$ for any $S \leq k$, $S \in [n]$.*

Theorem 2. *If A has $(3k, 0.1)RIP$, given $y = Ax + e$ where x is k-sparse, then IHT recovers $\hat{x}$ with $\|\hat{x} - x\|_2 \leq \|e\|_2$.*

This theorem indicates that if there is no error, you can recover it exactly. If there is some small error ϵ, you will recover with this small norm error.

So how this theorem expected to work? Assume $z = A^\top y = A^\top Ax + A^\top e$. Let $H_k(z) \in \mathbb{R}^n :=$ closest k-sparse approximation for any $z \in \mathbb{R}^n$. Then $\hat{x} = H_k(z)$ has

$$\|\hat{x} - x\| \leq O(\epsilon)\|x\| + \|(A^\top e)_S\|.$$

By looking at Definition 1, for any $\omega \in \mathbb{R}^{|S|}$, $\|A_S\omega\| \leq \sqrt{1+\epsilon}\|\omega\|$, which is equal to saying that the singular values of A_S are less than or equal to $\sqrt{1+\epsilon}$. The same property holds for $A_S^\top$. So we can get

$$\begin{aligned}\|\hat{x} - x\| &\leq O(\epsilon)\|x\| + \sqrt{1+\epsilon}\|e\| \\ &\leq O(\epsilon)\|x\| + 2\|e\|.\end{aligned}$$

This argue means we can get closer, where two means we can get down to the noise level fast.

Now we will restate the algorithm in order to repeat. Suppose we have current guess $x^{(i)}$, then we set

$$\begin{aligned}x^{(i+1)} &= H_k(A^\top(yAx^{(i)}) + x^{(i)}) \\ &= x^{(i)} + A^\top e + A^\top A(x - x^{(i)})t.\end{aligned}$$

Run this equation repeatedly until you get to $\|x^{(l)} - x\| \leq \|e\|$ after $log(\frac{\|x\|}{\|e\|})$ iterations.

If A has $(3k, 0.1)$ RIP, this holds with $O(k \log \frac{n}{k})$ Gaussian rows. Basically, this is optimal in general.

So what number should we expect of the Gaussian rows? $\Omega(k)$ words is necessary. In order to match "compression", how is the basic requirement of compression?

$$\underbrace{\text{(identify support)}}_{k\log(\frac{n}{k})\text{ bits}} + \underbrace{\text{(tell values on support)}}_{k\text{ words}}$$

If we don't have noise, we can do that in just k words. But if we want to be robust to noise, we cannot avoid the $k\log(\frac{n}{k})$ issue. This is quite similar to the lower bound for moment estimation by Gaussian noise last week.

In some circumstances, we don't need $k\log(\frac{n}{k})$ bits to describe k words, because we don't expect our vector to be an arbitrary sparse vector. We can do better with **structural sparsity**, which is also known as "Model-Based" Compressed Sensing.

3 "Model-Based" Compressed Sensing

Definition 3. *Suppose x is k-sparse and supp$(x) \in \mathcal{F}$, where $\mathcal{F}$ = set of supports. Then $m = O(k + \log|F|)$ suffices for "Model-Based" Compressed Sensing.*

We first see some examples of $\mathcal{F}$.

Example 1: Block Sparsity:

If the signal comes in blocks, that is the support of x S consists of several blocks.

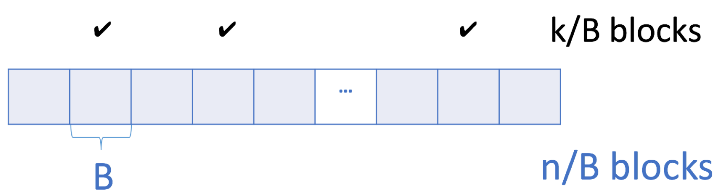

Figure 1: Block Sparsity

Suppose the number of the block is B, then we can pick n/B disjoint block, and we need to choose k/B block out of these blocks, thus the total support number of size k should be $|\mathcal{F}| = \binom{n/B}{k/B}$. So m is approximately $log\binom{n/B}{k/B} \approx \frac{k}{B}\log\binom{n}{k}$.

Example 2: Tree Sparsity

If we want to find a subset of connected tree that has k nonzeros containing root, how large is the set of Tree Sparse Supports?

Well, one way to think of a connected subset of tree is to do a tranverse of a tree, and to see how many different way can we represent a traverse over k connected vertices. If we do a traverse of all the connected support of the tree, then for each vertex at some root of a subtree, there are 4 choices of next layer of its child information, that is none of its child are in the support, one of its child is in the support and both of its child are in the support. So we need 2 bits for every vertex for a tree to represent the traverse, that is totally at most $2k$ bits. So $|\mathcal{F}| \leq 2^{2k}$, which means $O(k + log|\mathcal{F}|) = O(k)$ suffices.

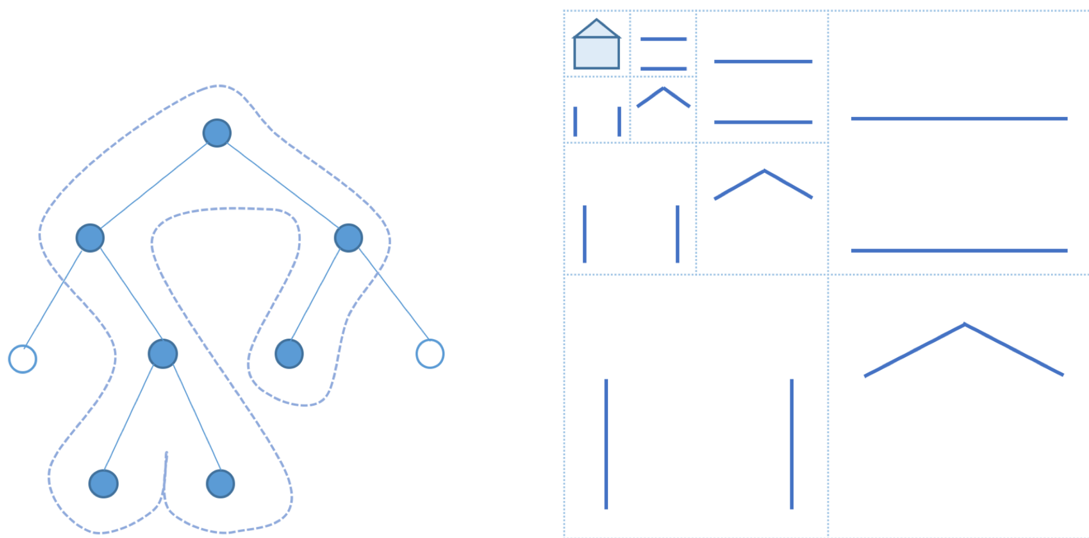

Figure 2: Tree Sparsity

Tree sparsity can also be used in sparse images. As we can see in the right part of Figure. 2, the sparse structure with linear operations of the figure can be modeled as a tree model. Our analysis of this tree sparsity example can be applied in image processing field.

If we assign nonnegative value on every vertex and would like to find the maximum k-sparse support on a tree, table 1 shows the complexity of this specific problem (which we will prove in homework).

Table 1: algorithm complexity

nk^2	easy DP
nk	harder DP
$n \log n$	$O(1)$ approximation

Now we generalize the "Model-Based" compressed sensing.

Theorem 4. *Suppose A has $(\mathcal{F}, 1)$ model RIP. If for any x with $supp(x) \in \mathcal{F}$, we have*

$$\|Ax\|_2^2 = (1 \pm \epsilon)\|x\|_2^2,$$

Then Guassian Matrix has this for $M = O(k + \log|\mathcal{F}|)$ with constant ϵ.

So how to change the algorithm to Model-Based Compressed Sensing?

$$H_{\mathcal{F}}(z) := \underset{minimize\|\hat{z}-z\|}{\hat{\text{z}}} \text{ with } supp(\hat{z}) \in \mathcal{F}$$

Proof. Analysis uses RIP on $supp(x^{(i)}) \cup supp(x^{(i+1)}) \cup supp(x)$. So we just need RIP on $\mathcal{F} \otimes \mathcal{F} \otimes \mathcal{F}$ with size less than $\|\mathcal{F}\|^3$. If A has $(\mathcal{F}^3, 1)$ RIP, then it has $O(k + log|\mathcal{F}|)$ Gaussian rows.

We need for any z, $\|x - H_{\mathcal{F}}(z)\| \leq \|x - z\|$.

needs an analysis that doesn't just pair up elements in $supp(x)$ and $supp(H_k(x))$. $\square$