CS 395T: Sublinear Algorithms Fall 2014

Lecture: Oct 23, 2014

Prof. Eric Price *Scribe: Tianyang Li*

1 Overview

- **adaptive algorithms for compressed sensing**
- **Fourier transform**

2 Adaptive algorithms for compressed sensing

2.1 Adaptive sparse recovery

For some $x \in \mathbb{R}^n$

- choose $v_1 \in \mathbb{R}^n$, observe $y_1 =< v_1, x >$
- then $v_2 \in \mathbb{R}^n$, get $y_2 =< v_2, x >$
  $\ldots$
- v_m, get $y_m =< v_m, x >$

Output $\hat{x}$ such that $\|x - \hat{x}\|_2 \leq (1+\epsilon) \min_{k-\text{sparse } x'} \|x - x'\|_2$ with $\frac{3}{4}$ probability.

Number of samples required:

- $\Omega(\frac{k}{\epsilon} + \log \log n)$ (known)
- $O(\frac{k}{\epsilon} + k \log \log \frac{n}{k})$
- $m = O(k \log \log \frac{n}{k})$

2.1.1

For a vector $x = (x_1 x_2 \ldots x_n)^T$ that satisfies $|x_i| \propto i^{-\alpha}$

- if $\alpha > 1$ then $\|x\|_1$ is bounded
- if $\alpha > 0.5$ then $\|x\|_2$ is bounded

2.1.2

We showed $\Omega(\log\log n)$ bound for $k = 1$, then by embedding k copies we can a bound for a general k.

2.2

$\|x - \hat{x}\|_1 \leq C \min_{k-\text{sparse} x'} \|x - x'\|_1$

Suppose $k = 1$

Let i^* be the largest coordinate $|x_{i^*}|$, and $|x_{i^*}| > R\|x_{-i^*}\|_1$ where x_{-i^*} is x with coordinate i^* set to 0.

Take

$$v_1 = (1, 1, 1, \dots)^T \qquad y_1 = \sum x_i \tag{1}$$

$$v_2 = (1, 2, 3, \dots)^T \qquad y_2 = \sum i x_i \tag{2}$$

If $R = \infty$, then $\frac{y_2}{y_1} = i^*$.

In general,

$$y_1 = x_{i^*}(1 \pm \frac{1}{R}) \tag{3}$$

$$y_2 = i^* \pm \frac{n x_{i^*}}{R} \tag{4}$$

Then we have

$$\frac{y_2}{y_1} = i^* \frac{1}{1 \pm \frac{1}{R}} \pm \frac{\frac{n}{R}}{1 \pm \frac{1}{R}} \tag{5}$$

$$= i^* \pm O(\frac{n}{R}) \tag{6}$$

2.3 An algorithm

Algorithm 1

permute x randomly
$\bar{i} \leftarrow \frac{n}{2}$, $\Delta \leftarrow \frac{n}{2}$
$S \leftarrow \{j \mid |\bar{i} - j| < \Delta\}$
given $R \leftarrow \Theta(1)$
$t \leftarrow 0$
while $\Delta \geq 1$ **do**
    $t \leftarrow t + 1$
    $\bar{i} \leftarrow \frac{\sum_{i \in S} i x_i}{\sum_{i \in S} x_i}$
    $\Delta \leftarrow O(\frac{\Delta}{R} 2^t)$
    $R \leftarrow \Theta(R^2 2^{-t})$
    $S \leftarrow \{j \mid |\bar{i} - j| < \Delta\}$
end while

At each stage $\bar{i} = i^* \pm \Delta$ (i.e. $i^* \in S$).

$$\|x_{S \setminus i^*}\|_1 \leq \|(i^* \pm 2\Delta) \setminus i^*\|_1 \tag{7}$$

$$\mathrm{E}\left[\|(i^* \pm 2\Delta) \setminus i^*\|_1\right] = \frac{4\Delta}{n-1} \|x_{-i^*}\|_1 \tag{8}$$

$$\Rightarrow \quad \frac{|x_{i^*}|}{\|x_{S \setminus i^*}\|_1} \text{ (at stage } t) \geq R_t \tag{9}$$

$\Rightarrow$ if Δ_t, R_t, $\bar{i}_t$ are "good" then so are Δ_{t+1}, R_{t+1}, $\bar{i}_{t+1}$.

$$R \leftarrow \frac{R^2}{2^t} \tag{10}$$

$$\log R \leftarrow 2 \log R - t \tag{11}$$

$\log R$ grows $\approx 2^t$

$$t = O(\log \log n)$$
$$\Rightarrow R \approx n$$

then $\Delta < 1 \Rightarrow S = -\{i^*\}$.

For a general k Hash $[n]$ to $[B]$ where $B = O(k)$, and solve individual buckets.

Each i is alone with probability $1 - \frac{k}{B} \geq \frac{3}{4}$.

$O(k \log \log \frac{n}{k}) \Rightarrow$ find $\frac{1}{2}$ of heavy hitter.

Repeat on rest of coordinates.

Time is

$$\begin{aligned} & k \log\log \frac{n}{k} + \frac{k}{2} \log\log \frac{n}{\frac{k}{2}} + \frac{k}{4} \log\log \frac{n}{\frac{k}{4}} + \dots \\ =& O(k \log\log \frac{n}{k}) \end{aligned}$$

2.4 An alternative algorithm

```
k' ← k                                        ▷ current sparsity estimate
while k' ≥ 1 do
    choose random n/k coordinates
    run the subroutine, set i*
    remove i* from set
    k' ← k' − 1/10
end while
```

Intuition As algorithm progresses, it's unlikely for k' to be less then the current sparsity parameter after i^* has been removed.

2.5

$L^1 \to L^2$ **for** $\|x - \hat{x}\|_2 \le (1+\epsilon) \min_{k-\textbf{sparse } x'} \|x - x'\|_2$

Basic idea Change $\bar{i}$ to $\frac{\sum \pm i x_i}{\sum_p m x_i}$ in Algorithm 1.

3 Discrete Fourier transform

- original $x \in \mathbb{C}^n$
- Fourier transform $\hat{x}$ denotes $\mathcal{F}(x) = Fx$

The discrete Fourier transform is given by

$$\hat{x}_i = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} \omega^{ij} x_j \tag{12}$$

$$F = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & 1 & \cdots \\ \omega & \omega^2 & \omega^3 & \cdots \\ \omega^2 & \omega^4 & \omega^6 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{13}$$

$$(F)_{ij} = \frac{1}{\sqrt{n}} \omega^{ij} \tag{14}$$

where

$$\omega = e^{-\frac{2\pi\sqrt{-1}}{n}} \tag{15}$$

F is unitary, and its inverse F^{-1} is

$$F^{-1} = F^H = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & 1 & \cdots \\ \omega^{-1} & \omega^{-2} & \omega^{-3} & \cdots \\ \omega^{-2} & \omega^{-4} & \omega^{-6} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{16}$$

$$(F^{-1})_{ij} = \frac{1}{\sqrt{n}} \omega^{-ij} \tag{17}$$

3.1 Properties

Convolution For $a = (a_1, a_2, \dots)^T \in \mathbb{C}^n$ and $b = (b_1, b_2, \dots)^T \in \mathbb{C}^n$

- $a \cdot b = (a_1 b_1, a_2 b_2, \dots)$
- $a * b = (c_1, c_2, \dots)$ where $c_i = \sum_{j=0}^{n-1} a_j b_{i-j}$ (convolution)

then

- $\widehat{a \cdot b} = \hat{a} * \hat{b}$
- $\widehat{a * b} = \hat{a} \cdot \hat{b}$

Parseval's theorem $\|x\|_2 = \|\hat{x}\|_2$

Plancherel's theorem $< x, y >=< \hat{x}, \hat{y} >$

References

[IPW11] Piotr Indyk, Eric Price, David P. Woodruff. On the Power of Adaptivity in Sparse Recovery. *FOCS 2011*.

[PW12] Eric Price, David P. Woodruff. Lower Bounds for Adaptive Sparse Recovery. *http://arxiv.org/abs/1205.3518*.