CS 395T: Sublinear Algorithms	Fall 2016

Lecture 17 — October 25, 2016

Prof. Eric Price *Scribes: John Kallaugher & Quinten McNamara*

1 Overview

In this lecture, we focus on proving lower bounds for sparse recovery. We define the sparse recovery problem as: given $y = Ax$ and $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$, find $\hat{x}$ such that

$$\|\hat{x} - x\|_p \leq C \cdot \min_{\text{k-sparse } x_k} \|x - x_k\|_p$$

In this lecture we will provide and prove lower bounds on m for the following scenarios:

- A is deterministic, $p = 2$
- A is deterministic, $p = 1$
- A is randomized, $p = 1$

2 A is deterministic, $p = 2$

Consider the case that $y = 0$. The output must then be $\hat{x} = 0$. This means that for any x that exists in the null space of A (denoted $\mathcal{N}$), $\hat{x} = 0$ is a valid solution. Formally

$$x \in \mathcal{N} \implies Ax = y = 0 \implies \hat{x} = 0$$

Going back to the sparse recovery guarantee, we get

$$\begin{aligned}
\|\hat{x} - x\|_2 &\leq C \min_{\text{k-sparse } x_k} \|x - x_k\|_2 \\
\|\hat{x} - x\|_2^2 &\leq C^2 \min_{\text{k-sparse } x_k} \|x - x_k\|_2^2 \\
\sum_{i=1}^{n} x_i^2 &\leq C^2 \sum_{j \neq i} x_i^2, \forall j \in [n] \\
x_j^2 &\leq (C^2 - 1) \sum_{j \neq i} x_i^2 \\
x_j^2 &\leq (C^2 - 1)\|x\|_2^2 - x_j^2 \\
x_j^2 &\leq (1 - \frac{1}{C^2})\|x\|_2^2
\end{aligned}$$

This means that each x_j can not have much of the energy of x. Let $\alpha = 1 - \frac{1}{C^2}$, $\alpha < 1$.

$$x_j^2 \leq \alpha \|x\|_2^2$$

From this, we will show that the dimension of the null space has an upper bound of $\sqrt{\alpha} \cdot n$

Proof. Let $v_1, v_2, \ldots, v_{n-m}$ be the orthonormal basis for $\mathcal{N}$, and $e_1, e_2, \ldots e_n$ be the standard basis vectors. Denote the projection of e_i on the null space as $Proj_{\mathcal{N}}(e_i)$. Observe that

$$Proj_{\mathcal{N}}(e_i) = \sum_{j=1}^{n-m} v_j v_j^T e_i$$

$$< e_i, Proj_{\mathcal{N}}(e_i) > = \sum_{j=1}^{n-m} e_i^T v_j v_j^T e_i$$

Since the $Proj_{\mathcal{N}}(e_i)$ is in the null space, we can use the bound we just found above to provide an upper bounds for the inner product (since the inner product will just be the ith index of the projection).

$$(\sum_{j=1}^{n-m} e_i^T v_j v_j^T e_i)^2 \leq \alpha \|Proj_{\mathcal{N}}(e_i)\|_2^2$$

$$\sum_{j=1}^{n-m} e_i^T v_j v_j^T e_i \leq \sqrt{\alpha} \|Proj_{\mathcal{N}}(e_i)\|_2$$

$$\sum_{j=1}^{n-m} e_i^T v_j v_j^T e_i \leq \sqrt{\alpha}$$

Summing the inequality over $i \in [n]$

$$\sum_{i}^{n} \sum_{j=1}^{n-m} e_i^T v_j v_j^T e_i \leq n\sqrt{\alpha}$$

$$\sum_{j=1}^{n-m} \|v_j\|_2^2 \leq n\sqrt{\alpha}$$

$$n - m \leq n\sqrt{\alpha}$$

$$m \geq (1 - \sqrt{\alpha})n$$

$$m = \Omega(n)$$

This means we cannot have a deterministic sublinear algorithm that gives the ℓ_2/ℓ_2 guarantee.

□

3 A is deterministic, $p = 1$

3.1 Gilbert-Varshamov Bound

The bound states that $\forall q, k \in \mathbb{Z}, \epsilon \in \mathbb{R}, \epsilon < 1 - \frac{1}{q}$, $\exists S \subseteq [q]^k$ such that the minimum Hamming distance in S is $\geq \epsilon k$ and

$$\log |S| \geq (1 - H_q(\epsilon)) \cdot k \log q$$

where

$$H_q(\epsilon) = -\epsilon \log_q (\frac{\epsilon}{q-1}) - (1-\epsilon) log_q (1-\epsilon)$$

Proof. We will prove the Gilbert-Varshamov bound by using a volume covering argument.

$$|S| \geq \frac{q^k}{\sum\limits_{i=0}^{\epsilon k-1} \binom{k}{i}(q-1)^i}$$

where the denominator represents the size of a ball with radius $\epsilon k - 1$

$$\sum_{i=0}^{\epsilon k-1} \binom{k}{i}(q-1)^i \leq \binom{k}{\epsilon k}(q-1)^{\epsilon k} \approx (q/\epsilon)^{\epsilon k} \approx q^{\epsilon k \log_q \frac{q}{\epsilon}} \approx q^{k H_q(\epsilon)}$$

therefore

$$|S| \geq \frac{q^k}{q^{k H_q(\epsilon)}}$$

taking the log of both sides gives us

$$\log |S| \geq (1 - H_q(\epsilon)) k \log q$$

which is the Gilbert-Varshamov bound. □

Claim 1. *$\exists$ set $S \subseteq \{0,1\}^n$ of k-sparse vectors with minimum ℓ_1 distance k of size* $\log |S| \gtrsim k \log \frac{n}{k}$

Proof. Apply the Gilbert-Varshamov bound, setting $q = \frac{n}{k}$. Conceptually, this can be achieved by encoding each character in q as a one-hot vector, consequently each string in $[q]^k$ maps to a k-sparse vector. Furthermore, set $\epsilon = \frac{1}{2}$ in order to get a minimum Hamming distance of k, since each different character will create a difference of 2 (0 where a 1 should be and 1 where a 0 should be in the string vector).

$$\log |S| \geq (1 - H_{\frac{n}{k}}(\frac{1}{2})) k \log \frac{n}{k}$$

Furthermore, we know that $(1 - H_{\frac{n}{k}}(\frac{1}{2}))$ is a constant, thus

$$\log |S| \gtrsim k \log \frac{n}{k}$$

This means that the set S is very large, but is also well separated. □

3.2 Proof for lower bound of ℓ_1/ℓ_1

We need to show that we can recover x exactly using the set S, even in the presence of noise. Suppose we have sparse recovery for $C = 3$. Consider any $x' = x + w$ where $x \in S$ and $\|w\|_1 \leq \frac{k}{100}$. Using the guarantee, we have $\|\hat{x} - x'\|_1 \leq \frac{3k}{100}$. Therefore, by the triangle inequality we have $\|\hat{x} - x\|_1 \leq \frac{4k}{100}$, or a looser bound of $\|\hat{x} - x\|_1 \leq \frac{k}{2}$. Since the elements of S are k away from each other, we can round $\hat{x}$ to the nearest element of S and recover x exactly.

So we have several balls $x_i + B_1(\frac{k}{10})$ for each $x_i \in S$. Given we are maintaining a sketch, each ball is projected into a larger simplex shape when multiplied by A. Furthermore, we know that every ball in the projected space is disjoint and lie within the projection of the ball $B_1(k + \frac{k}{10})$ since the maximum ℓ_1 distance for each k-sparse binary vector is k and each ball can at most stretch $\frac{k}{10}$ further away. This ball is $A \cdot B_1(\frac{11k}{10})$ Therefore the most balls we can have is

$$|S| \leq \frac{Vol(A \cdot B_1(\frac{11k}{10}))}{Vol(A \cdot B_1(\frac{k}{10}))}$$

The balls in the numerator and denominator have the same shape, but different radius. Since scaling the radius scales the volume exponentially by the dimensionality, the ratio of the volumes must be 11^m (radii are scaled by 11). Thus we get

$$\begin{aligned} |S| &\leq 11^m \\ \log |S| &\leq m \\ m &\gtrsim k \log \frac{n}{k} \\ m &= \Omega(k \log \frac{n}{k}) \end{aligned}$$

4 A is randomized, $p = 1$

We will extend this bound to a randomized A by a reduction to the *Augmented Indexing Problem.*

Definition 2 (Augmented Index)**.** *In the Augmented Indexing problem on n bits, Alice has a string w of n bits, and Bob has an index i, and the bits $(w_j)_{j<i}$ of w that come before i. Alice must send a message to Bob (without knowing Bob's index) such that Bob can determine the value of w_i.*

It can be shown that, even in the randomized model (where Bob only needs to succeed with, say, 2/3 probability), Alice must send $\Omega(n)$ bits.

As in the deterministic case, we may use Gilbert-Varshamov to construct a set S of k-sparse vectors with binary coefficients, separated by at least $\frac{k}{10}$, with $|S| \gtrsim k \log \frac{n}{k}$.

Alice may encode a string of $R \log |S|$ (for some R to be determined later) bits as follows: She chooses R vectors X_j from S (encoding each block of $\log |S|$ bits by choosing a vector from S). She then combines these into the vector $\overline{X} = \sum_{j=0}^{R-1} \frac{X_j}{11^j}$. She then sends Bob the vector $A\overline{X}$, where A is the sketching matrix.

If Bob can successfully perform ℓ_1 recovery (with constant C, say — if C is a larger constant, we can deal with this by changing the constants in our construction of $\overline{X}$), he can recover X_1 exactly, as each X_j has 1-norm no more than k, so $|| \sum_{j=1}^{R-1} \frac{X_j}{11^j} ||_1 \leq \frac{k}{11} \frac{1}{1-\frac{1}{11}} = \frac{k}{10}$. If he successfully recovers X_1, he can then repeat the process on $A\overline{X} - AX_1$, and iterate until he recovers $X_{\left\lceil \frac{i+1}{\log |S|} \right\rceil}$, and therefore w_i.

There are two problems with this method. Firstly, if our failure probability is δ, by repeating the process we amplify the failure probability to δR. However, we are in the *Augmented* indexing model, and so we may assume that Bob knows the bits up to i. He can then use these to determine the vectors X_j for $j < \left\lceil \frac{i+1}{\log |S|} \right\rceil$, and so he needs only perform one succesful sparse recovery.

The second problem is that $A\overline{X}$ is a real-valued vector, and would therefore require infinite communication to send with arbitrary precision. We must therefore argue that this protocol succeeds even if we round $A\overline{X}$ to use only $O(\log n)$ bits per entry. If $Y = A\overline{X}$, we can think of our rounded vector as $Y' = Y + e$ where $||e|| \leq n^{10}$. (By using a sufficiently large constant times $\log n$ bits.)

WLOG, we may assume that A is orthonormal, as we may use the SVD to decompose A into $U^T \Sigma V$, and then use V as our sketching matrix, and then apply $U^T \Sigma$ to our sketch vector as a post-processing step. This will give us

$$Y' = Y + e = A(\overline{X} + e')$$

with $||e'|| \leq \frac{1}{n^9}$. So Bob will still be able to recover the X_j provided $R = O(\log n)$, as the size of the error will be amplified each time he multiplies $\overline{X}$ by 11, but this happens at most R times.

However, there is still a flaw in this argument. A sparse recovery algorithm must work for any fixed X, with good probability, with A chosen at random after X is fixed. But we have changed X dependent on A (as our rounding depends on A), so we no longer have that guarantee. This opens up the possibility that, for instance, our recovery algorithm fails if AX is rounded (which will happen with zero probability if A is chosen from a continuous distribution), but works otherwise. In order to evade this problem, Bob can add some random noise that is substantially larger than e but still too small to prevent correct recovery (say, with magnitude of order $\sqrt{||e||}$). As A is chosen independently of X, most A must work for most X, so this allows us to recover our guarantees.

This means Alice sends $O(m \log n)$ bits in total, and so as we know she needs to send $\Omega(R \log |S|) = \Omega(k \log n \log \frac{n}{k})$, we may conclude that

$$m \gtrsim k \log \frac{n}{k}$$

.