1 Overview

Last lecture, we talked about adaptive sparse recovery. Today, we will talk about RIP-1, expanders and SSMP.

2 Sparse Matrix with RIP-1

Recalled from homework 3, we showed that 0-1 matrices that satisfy the RIP-2 cannot be very sparse. Alternatively, there exists a lower bound \( m = \Omega(k^2) \) that an 0-1 matrix \( A \in \mathbb{R}^{m \times n} \) with \( d = O(\log n) \) ones per column satisfies RIP-2. But \( A \) can satisfy the following RIP-1 property [BGIKS08]. Today we will show deterministic sparse recovery and fast-embedding via sparse matrices that satisfy RIP-1 property.

**Definition 1.** A has RIP-1 of \((k, \epsilon)\) if \( \forall k\)-sparse \( x \), \( (1 - \epsilon)\|x\|_1 \leq \|Ax\|_1 \leq \|x\|_1 \).

**Definition 2.** \( G = (U, V, E) \) is a bipartite graph with left-degree \( d \).

**Claim 3.** A random sparse binary matrix with \( m = \frac{1}{\epsilon^2}k\log^2 n, d = \frac{1}{\epsilon} \log^2 n \) satisfies RIP-1 (after scaling by \( \frac{1}{d} \), \( d \) is degree of sparse matrix that is number of ones by column).

**Claim 4.** More generally, an adjacency matrix of expander graph \( G(n, m, k, d, \epsilon) \) has \((k, 2\epsilon)\) RIP-1.

**Theorem 5.** A random sparse binary matrix with \( m = \frac{1}{\epsilon^2}k\log^2 n, d = \frac{1}{\epsilon} \log^2 n \) is a \((k, 2\epsilon)\) expander with high probability.

**Proof.** \( \forall S \) of size \( k \), \( m = \frac{1}{\epsilon^2}k\log^2 n, d = \frac{1}{\epsilon} \log^2 n \). \( N(S) \) has \( dk \) balls in \( m \) bins. Consider all \( kd \) edges, define \( V_1, V_2, \cdots, V_{kd} \in [m] \) and be in i.i.d. Let \( E_i \) denotes the event that any ball \( V_i \) collides with previous balls, that collides with \( V_1, V_2, \cdots, V_{i-1} \). We have

\[
Pr[E_i] \leq \frac{i - 1}{m} \leq \frac{kd}{m} \leq \epsilon
\]

\[
Pr[\sum_{j=1}^{kd} E_j > \epsilon kd] \leq e^{-\Omega(\epsilon kd)}
\]

\[
Pr[|N(S)| \leq (1 - 2\epsilon)dk] \leq e^{-\Omega(\epsilon kd)}
\]
Set $d = O\left(\frac{1}{2} \log \frac{n}{k}\right)$

$$\Rightarrow Pr[|N(S)| \leq (1-2\epsilon)dk] \leq e^{-\Omega(\log \frac{n}{k})} \leq \frac{1}{(\frac{n}{k})^{2\epsilon}} \leq \frac{1}{(\frac{n}{k})}$$

By union bound, all sets $S$ of size $k$ expands with $1-\frac{1}{(\frac{k}{n})}$ probability. For sets of size less than $k$, if size is $k'$, it still works with $1-\frac{1}{(\frac{k'}{n})}$. So by union bound over $k'$, all sets of size $\leq k$ expands with good probability.

Claim 6. There exists explicit expander constructions for $\forall \alpha > 0, d = \log n \cdot \left(\log \frac{\epsilon}{\alpha}\right)^{1+\frac{1}{\alpha}}, m = k^{1+\alpha}d^2$.

3 L1 minimization with RIP-1

If we have RIP-1 matrices, we can use L1 minimization to do sparse recovery. Given $y = Ax + e$, pick $\hat{x} = \text{argmin} \|\hat{x}\|_1 \text{ s.t.} \|A\hat{x} - y\|_1 \leq \Delta$.

Theorem 7. If $A$ has $(k, 2\epsilon)$ RIP-1, $\|e\|_1 \leq \Delta$, then we have $\|\hat{x} - x\|_1 \preceq 2\Delta$.

Set $z = \hat{x} - x$, we have $\|Az - e\|_1 = \|A(\hat{x} - x) - e\|_1 = \|A\hat{x} - y\|_1 \leq \Delta$. By triangle inequality, we have $\|Az\|_1 \leq \|Az\|_1 + \|e\|_1 \leq 2\Delta$. We only need $\|z\|_1 \preceq \|Az\|_1$, so that $\|\hat{x} - x\|_1 \preceq 2\Delta$ follows.

Lemma 8. If $Az = 0, \forall |S| = k$, then $\|z_s\|_1 \leq \frac{2\epsilon}{1-2\epsilon}\|z\|_1$.

Proof. Partition $[n]$ into $S_0 \cup S_1 \cup \cdots \cup S_L, |S_i| = k$ in decreasing order of $z_i$, and $S = S_0$. $m = \frac{1}{2} k \log \frac{n}{k}$, $d = \frac{1}{2} k \log \frac{n}{k}$. Then $|N(S)| \approx d|S| = dk = O(em)$. 

2
Pick $A' = \text{rows of } A$ corresponding to $N(S)$, $Az = 0 \Rightarrow A'z = 0$.

$$0 = \|A'z\|_1 = \|A'z_s + \sum_{l \geq 1} (A'z_{s_l})\|_1 \geq \|A'z_s\|_1 - \|\sum_{l \geq 1} (A'z_{s_l})\|_1 \text{ (by triangle inequality)} \geq \|A'z_s\|_1 - \sum_{l \geq 1} \|A'z_{s_l}\|_1 \text{ (by triangle inequality)} \geq (1 - 2\epsilon) \cdot d \cdot \|z_s\|_1 - \sum_{l \geq 1} (\# \text{ edges from } S_l \text{ to } N(S)) \cdot \max_{i \in s_l} |z_i| \text{ (by definition of RIP-1)}$$

$$\mathbb{E}[^{\# \text{ edges from } S_l \text{ to } N(S)}] = \frac{d}{m} \cdot |N(S)| \cdot k < \frac{dk}{m} \cdot dk < \frac{dk}{m} \cdot me = \epsilon mk \text{ (by } dk \epsilon m)$$

w.h.p[^{\# \text{ edges from } s_l \text{ to } N(S)}] < 2\epsilon mk

$$\geq (1 - 2\epsilon) \cdot d \cdot \|z_s\|_1 - 2\epsilon mk \cdot \sum_{l \geq 1} \|z_{s_{l-1}}\|_1 \text{ (by decreasing order of } z_i) \geq (1 - 2\epsilon) \cdot d \cdot \|z_s\|_1 - 2\epsilon \cdot d \cdot \|z\|_1 \Rightarrow \|z_s\|_1 \leq \frac{2\epsilon}{1 - 2\epsilon} \|z\|_1$$

4 Sequential Sparse Matching Pursuit

Suppose $x$ is sparse, given $y = Ax$ ($A$ is random sparse RIP-1 binary matrix), for each $i$, how to estimate $x_i$? We can minimize $\|y - A(\hat{x}_i \cdot e_i)\|_1$, it turns out that $\hat{x}_i = \text{median}(y_i)$ (similar to 'count-sketch').

**Algorithm: SSMP**

1) Let $x^{(0)} = 0$

2) For $r = 1, 2, \cdots, T = O(\log(\|x\|_1/\|e\|_1))$

   a) For $t = 1, 2, \cdots, 10k$

      - $\hat{x}_i \leftarrow \text{median}_{j \in N(i)} (y - Ax^{(r)})_j$
      - Let $i$ be the largest term of $\hat{x}$
      - Let $x^{(r)} \leftarrow x^{(r)} + \hat{x}_i e_i$

   b) Let $x^{(r+1)} = H_k(x^{(r)})$ (the top $k$ values of $x^{(r)}$)

3) Report $x' = x^{(T)}$

The idea of proving SSMP is using the Lemma below: Given $y = Ax$,
Lemma 9. There exists \( i \) s.t. \( \| y - A\hat{x}_i \cdot e_i \|_1 \leq (1 - \frac{1}{10k})\|y\|_1 \)

Therefore each step \( x^{(r)} = x^{(r)} + \hat{x}_i e_i \) decreases \( \| y - Ax^{(r)} \| \) by \( 1 - \frac{\Omega(1)}{k} \), after \( O(k) \) steps, we have

\[
\| y - Ax^{(r)} \| \leq \frac{1}{10} \| y - Ax^{(r-1)} \|
\]

repeat \( \log(\frac{\|x\|_1}{\epsilon}) \) times converges to \( \| y - Ax^* \|_1 \leq \epsilon \).

A more detailed proof can be found at [BI09].

How fast is SSMP? It will take \( \log(\frac{\|x\|_1}{\epsilon}) \) times sequentially add \( O(k) \) single terms to minimize \( \| y - Ax^* \|_1 \).

Naive update will take \( O(knd) \) per loop. Can do better with \( O(dn\log(n)) \)

We can spend \( nd \) times for the first update and will find all the neighbors in \( y \). For every successive time, it modifies \( d \) elements of \( y \) by looking at all neighbors of these.

\[ \Rightarrow \frac{d^2 n}{m} \approx \frac{dn}{k} \log(n). \]

References
