1 Sparse Matrices & RIP

We have seen in the homework that no sparse matrices have RIP-2, i.e. \( \forall \) \( k \)-sparse \( x \),
\[
\|Ax\|_2 = (1 \pm \epsilon)\|x\|_2.
\]

But we can have sparse matrices have RIP-1: \( \forall \) \( k \)-sparse \( x \)
\[
\|Ax\|_1 = (1 \pm \epsilon)\|x\|_1
\]

**Constructions** Consider random \( A \in \{0, 1\}^{m \times n} \) subject to \( d \) = \( O(\log n) \) entries of 1 per column

\( A \) has (normalized) RIP-1: \( \forall \) \( k \)-sparse \( x \),
\[
(1 - \epsilon)d\|x\|_1 \leq \|Ax\|_1 \leq d\|x\|_1
\]

**Lemma 1.** \( A \in \{0, 1\}^{m \times n} \) is RIP-1 with sparsity \( d \) if and only if \( A \) is adjacency matrix of a \( d \)-regular bipartite expander (with \( n \) nodes on left and \( m \) nodes on right).

Bipartite expander: \( \forall S \subseteq [n] \) on left, \( |S| \leq k \), \( |N(S)| \geq (1 - \epsilon)d|S| \).

**Claim 2.** With random graph: \( d \gtrsim \frac{1}{\epsilon} \log \frac{n}{k} \), \( m \gtrsim \frac{1}{\epsilon^2} \log \frac{n}{k} \) = \( \frac{1}{\epsilon}kd \) suffices. We also have explicit graph with \( d = \log n(\log \frac{k}{\epsilon})^{1+\frac{1}{\epsilon}} \), \( m = k^{1+\alpha}d^{2} \) that satisfies RIP-1.

**Lemma 3.** Random Graph with \( d \gtrsim \frac{1}{\epsilon} \log \frac{n}{k} \), \( m \gtrsim \frac{1}{\epsilon} \log \frac{n}{k} = \frac{1}{\epsilon}kd \) is an expander with high probability.

**Proof.**
\[
P[\text{random graph is not expander}] \\
= P[\exists S, |S| = k, |N(S)| < (1 - \epsilon)d|S|] \\
\leq \binom{n}{k}P[\exists S, |S| = k \text{ has } |N(S)| \leq (1 - \epsilon)kd]
\]

Consider the following balls and bins problem: \( kd \) balls placed randomly among \( \frac{kd}{\epsilon} \) bins.
\[
P[\text{bin } i \text{ is empty}] = (1 - \frac{\epsilon}{kd})^{kd} \approx \exp(-\epsilon)
\]
So

\[ \mathbb{E} [\text{# of non-empty bins}] = \frac{kd}{\epsilon} (1 - \exp(-\epsilon)) \approx kd(1 - O(\epsilon)), \]

which is good. But we need high probability bounds.

Define \( X_j \) the indicator of the event that the \( j \)-th ball collides with previous balls. We have

\[ \mathbb{P} [X_j = 1 \mid \text{balls 1, \cdots, } j - 1] \leq \epsilon. \]

We can then apply Chernoff bound as

\[
\mathbb{E} \left[ \exp \left( \sum_{j \in [kd]} \lambda X_j \right) \right] = \prod_{j \in [kd]} \mathbb{E} [\exp(\lambda X_j) \mid \text{balls 1, \cdots, } j - 1] \leq (\exp(\lambda) + 1 - \epsilon)^{kd}.
\]

With multiplicative Chernoff bound, we have

\[ \mathbb{P} \left[ \sum_{j \in [kd]} X_j \geq 2\epsilon kd \right] \leq \exp \left( -\frac{\epsilon kd}{3} \right), \]

and thus

\[ \mathbb{P} [|N(S)| \leq (1 - 2\epsilon kd)] \leq \exp \left( -\Theta \left( k \log \frac{n}{k} \right) \right). \]

By choosing proper constant and union bound, we have the desired result with high probability.

### 2 Sequential Sparse Matching Pursuit

Given \( y = Ax, x \) is \( k \)-sparse. We want to do the \( \ell_1 \) sparse recovery, by picking \((\alpha, i)\), s.t. \( \hat{x} + \alpha e_i \) is a bit closer to \( x \) than 0. A natural way is picking \((\alpha, i)\) minimizes

\[ \| (y - A\hat{x}) - A(\alpha e_i) \|_1 = \| (y - A\hat{x}) - \alpha a_i \|_1 \quad (A = (a_1, a_2, \cdots, a_n)) \]

*Can we repeat the \( \ell_1 \) minimization to do the sparse recovery?*

**Lemma 4.** Let \( Z = \sum_{i \in k} Z_i \), s.t. \( \sum \| Z_i \|_1 \leq \frac{1}{1-\epsilon} \| z \|_1 \), then \( \exists i, \text{ s.t. } \| z - z_i \|_1 \leq (1 - \frac{1-2\epsilon}{k})\| z \|_1. \)

As \( y = \sum x_i a_i \) and \( \| y \|_1 \geq d(1 - \epsilon)\| x \|_2 = (1 - \epsilon) \sum \| x_i a_i \|_1 \). We have

\[ \| y - \alpha a_i \|_1 \leq \left( 1 - \frac{1}{2k} \right) \| y \|_1. \]

Define \( y^{(2)} = y - \alpha a_i \) the residual after first round. And we have

\[ \| y^{(2)} - \alpha^{(2)} a^{(2)} \| \leq \left( 1 - \frac{1}{2k + 2} \right) \| y^{(2)} \|_1. \]
Algorithm 1: Sequential Sparse Matching Pursuit (SSMP)

**INPUT:** \( y = Ax + u \in \mathbb{R}^m \), \( A \) a random sparse RIP-1 binary matrix.

Initialize \( x^{(1)} = 0 \).

for \( l = 1, \ldots , L = \Theta(\log \|x\|_1 \|u\|_1) \) do

for \( t = 1, \ldots , 16k \) do

Pick \((\alpha, i)\) via minimizing \( \|y - Ax^{(r)} - \alpha a_i\|_1 \).

\( x^{(r)}_t \leftarrow x^{(r)}_t + \alpha a_i \).

end for

\( x^{(r+1)} = H_k(x^{(r)}_{16k}) \).

end for

After \( r \) repetitions with RIP-1 of order \((k + r)\), we have

\[
\|y^{(r)}\| \leq \frac{\sqrt{(2k + 1)(2k + 2r - 1)}}{2k + 2r} \approx \frac{1}{\sqrt{c}}.
\]

if \( r = ck \). But we can do hard thresholding:

\[
\|x - H_k(x^{(r)})\|_1 \leq \|x - x^{(r)}\|_1 + \|x^r - H_k(x^{(r)})\|_1 \leq 2\|x - x^{(r)}\|_1
\]

With the discussion above, we know that each of the inner loop have that

\[
\|x - x^{(r)}_{16k}\|_1 \leq \frac{1}{4}\|x - x^r\|_1,
\]

and after the hard thresholding, we have

\[
\|x - x^{(r+1)}\|_1 \leq \frac{1}{2}\|x - x^{(r)}\|_1.
\]

**Theorem 5.** If \( A \) has \((O(k), \frac{1}{4})\)-RIP, for Sequential Sparse Matching Pursuit, we have

\[
\|\hat{x}^L - x\|_1 \leq 2^{-L}\|x\|_1 + O(\|u\|_1)
\]

For time complexity, we first focus on the inner loop of the algorithm. A naive implementation would require \( O(n \log n) \) time for solving the minimization in the inner loop (i.e. the \( n \) part comes from searching through basis \( e_i \) and log \( n \) part comes from determining proper \( \alpha \)). The overall complexity would be \( O(kn \log^2 n) \).

However, notice that from the random graph construction, each time we add a new \( \alpha e_i \), it would affect \( d \) elements of \( y \), which in turn will affect the estimation of \( O(\frac{nd}{k}) \) basis \( e_i \). Therefore the complexity of the minimization in the inner product is around \( O(\frac{n^2}{k} \log^2 n) \), which leads to an overall complexity of \( O(n \log^{O(1)} n) \) which is nearly linear in \( n \).