CS 229r: Algorithms for Big Data Fall 2013

Lecture 19 — November 5, 2013

Prof. Jelani Nelson *Scribe: Kristan Temme*

1 Overview

In this lecture we will finish sparse recovery with ℓ_1/ℓ_1 - guarantee and start with the introduction of the matrix completion problem. The matrix completion problem is a problem that resides at the intersection of compressive sensing and numerical linear algebra. Then, on thursday we will finish the matrix completion problem, so that we can start with a new topic next week. Next week will be looking at external memory algorithms. Afterwards we will be looking at MapReduce and Hadoop.

1.1 ℓ_1/ℓ_1 - recovery

Last week we already introduced combinatorial expanders. These expanders give rise to $(\epsilon, k)-RIP_1$ - matrices, which preserve ℓ_1 - norm, i.e. $(1-\epsilon)\|x\|_1 \leq \|\Pi x\|_1 \leq (1+\epsilon)\|x\|_1$ for all x which are k-sparse. As of now, the best known iterative ℓ_1/ℓ_1 - sparse recovery algorithm is by Indyk and Ružić [IR08]. The algorithm is a so callad EMP (Expander Matching Persuit) - algorithm. A general survey of sparse recovery algorithms can be found in [GI10] by Gilbert and Indyk. This survey also contains the SMP (Sparse Matching Persuit) algorithm of Berinde, Indyk and Ružić [BIR08] which we will be discussing today.

2 SMP (Sparse Matching Persuit)

The SMP algorithm turns out to be simpler to analyze and in practice performs better than [IR08], however this observation has not been proven. We will cover the analysis presented in [BIR08], though after lecture Indyk informed the instructor that a newer analysis appears in a very recent book of Foucart and Rauhut [FR13, Section 13.4] which is both simpler and more general than the original analysis of [BIR08]. It is more general in that it also applies to the *model-based* compressed sensing setup, in which one has the extra information that the sparsity pattern which x (approximately) obeys comes from a family of possibile sparsity patterns whose size is much smaller than $\binom{n}{k}$ (knowing this information allows one to save on the number of measurements).

Set up:
We consider bipartite expander graphs $G = (U, V, E)$. G is said to have left degree d if every right vertex in U gets mapped to at most d vertices in V. We choose the convention that

- $|U| =$ n is the signal dimension
- $|V| =$ m is the measurement dimension.

Definition 1. *A bipartite, left d regular graph $G = (U, V, E)$ is called (s, ϵ, d)-expander, if every subsets $S \subset U$ of at most s vertices has at least*

$$\Gamma(S) = (1 - \epsilon)d|S| \tag{1}$$

neighbors. $\Gamma(S) = \{v \in V | (\exists u \in S)\ s.th\ (u, v) \in E\}$ denotes the left neighborhood of the set S. The adjacency matrix of the G will be denoted by $\Pi \in \{0, 1\}^{m\times n}$.

The ℓ_1/ℓ_1 - recovery guarantee for the recovered signal $\tilde{x}$ is a bound on the deviation from the exact signal x in terms of the best k-sparse truncation x^k of the form

$$\|x - \tilde{x}\|_1 \leq C\|x - x^k\|_1 \tag{2}$$

The constant C differs depending on whether a EMP algorithm or a SMP algorithm was used. One has that for EMP $C = 1 + \epsilon$, whereas for SMP $C = \mathcal{O}(1)$.

The trick that is used to give bounds in terms of the k-sparse vector x^k is to treat $\Pi(x - x^k)$ as an error in the recovery. The analysis of the SMP algorithm only uses the expanding properties of Π as a black box and does not use the fact that the matrix Π is RIP_1. It would be interesting to know if once could find a proof that only makes use of RIP_1 and does not refer to the expander directly.

2.1 The SMP algorithm

Given $\Pi x + e = b$, where x is the signal b the measurement output and e an error term, the algorithm of [BIR08] is as follows:

1. **Initialize** :
   (a) $j = 0$
   (b) $x_j = 0$
2. **Repeat T times**:
   (a) $j = j + 1$
   (b) $c = b - \Pi x^{j-1}$
   (c) $u^* = u^*(c)$, where $[u^*]_j = \text{median}(c_{\Gamma(j)})$
   (d) $u^j = H_{2k}(u^*)$
   (e) $x^j = x^{j-1} + u^j$
   (f) $x^j = H_k(x^j)$

Note that x^j will be always k-sparse after each iteration of the algorithm. Therefore the multiplication in (b) Πx^{j-1} is very fast.

A good intuition for this algorithm comes from the count median (min) sketch of lecture 5. For this problem we had an ℓ_∞/ℓ_1 guarantee of the form $\|x - \tilde{x}\|_\infty \leq \mathcal{P}(w^{-1})\|x - x^{w/z}\|_1$ as long as $z = \Omega(\log(n))$.

In the following we will assume that the signal is in fact k -sparse, i.e. $x = x^k$. If this is not the case, we can always redefine the error e as

$$\Pi x + e = \Pi x^k + \underbrace{\Pi(x - x^k) + e}_{\tilde{e}}. \tag{3}$$

Moreover note that

$$\|\tilde{e}\|_1 \leq \|e\|_1 + \|\Pi(x - x^k)\| \leq \|e\|_1 + d\|(x - x^k)\|. \tag{4}$$

Theorem 2 (SMP - algorithm [BIR08])**.** *The vector x^T deviates from the exact signal x after T iterations by*

$$\|x - x^T\|_1 \lesssim 2^{-T}\|x\|_1 + \frac{\|\tilde{e}\|_1}{d}. \tag{5}$$

2.2 Proof of SMP - Theorem

The proof will be broken up into smaller subproblems. We will show that after different stages of the algorithm we have have the following bounds

1. After (d) :

$$\|u^j - (x - x^{j-1})\|_1 \leq \frac{\|x - x^{j-1}\|_1}{4} + \frac{C}{d}\|\tilde{e}\|_1$$

2. After (e) :

$$\|x - x^j\|_1 = \|(x - x^{j-1}) - u^j\|_1 \leq \frac{\|x - x^{j-1}\|_1}{4} + \frac{C}{d}\|\tilde{e}\|_1$$

3. After (f) :

$$\|x - x^j\|_1 \leq \frac{\|x - x^{j-1}\|_1}{2} + \frac{2C\|\tilde{e}\|_1}{d}$$

Subproblem 2 implies 3:

The last inequality 3. will imply the claim of the Theorem by iteration. We therefore first show that 2. implies 3.

Lemma 3. *Let $x = x^k$ and x' arbitrary. Then*

$$\|H_k(x') - x\|_1 \leq 2\|x' - x\|.$$

Proof:
Define $S = \text{supp}(x)$ and $T = \text{supp}(H_k(x'))$. Note that we have that $|T| = |S| = k$ for which then $|S - T| = |T - S|$. Since the entries are grouped so that the largest k are supported on T we have $\|x'_{S-T}\|_1 \leq \|x'_{T-S}\|_1$. Now,

$$\begin{aligned}
\|H_k(x') - x\|_1 &= \|x'_T - x\|_1 \\
&= \|x_{S-T}\|_1 + \|x'_{T-S}\|_1 + \|(x - x')_{T\cap S}\|_1 \\
&\leq \|(x - x')_{T\cap S}\|_1 + \|x'_{S-T}\|_1 + \|x'_{S-T} - x_{S-T}\|_1 + \|x'_{T-S}\|_1 \\
&\leq \|(x - x')_{T\cap S}\|_1 + \|x'_{S-T} - x_{S-T}\|_1 + 2\|x'_{T-S}\|_1.
\end{aligned}$$

The last inequality is due to the aforementioned discussion. Moreover since $x_{T-S} = 0$ we have that

$$\begin{aligned} \|H_k(x') - x\|_1 &\leq \|(x - x')_{T\cap S}\|_1 + \|x'_{S-T} - x_{S-T}\|_1 + 2\|x'_{T-S} - x_{T-S}\|_1 \\ &\leq 2\|x' - x\|_1 \end{aligned}$$

□

Moreover, we have that 2 follows directly from the bound on 1.

Subproblem 1**:**

Before we prove subproblem 1, we need state two relevant lemmata. In what follows we set $u = x - x^{j-1}$. Therefore u is always $2k$ sparse. Furthermore we have for notational convenience that

$$|u_1| \geq |u_2| \geq \ldots \geq |u_{2k}|,$$

so that $\mathrm{supp}(u) = S = [2k]$.

Lemma 4. *Let Π be the adjacency matrix of a $(2k, \epsilon, d)$ - bipartite expander, u^* as defined in the algorithm step (c), then*

- **A:**
$$\|(u^* - u)_S\|_1 \leq \epsilon\|u\|_1 + \frac{\|\tilde{e}\|_1}{d}$$

- **B:** *Let $B \subset \overline{S}$ with $|B| \leq 2k$, then*
$$\|u^*_B - u_B\|_1 \leq \|u^*_B\|_1 \leq \epsilon\|u\|_1 + \frac{\|\tilde{e}\|_1}{d}$$

.

These two lemmata imply 1. Let $T = \mathrm{supp}(H_{2k}(u^*))$, then we have

$$\begin{aligned} \|H_{2k}(u^*) - u\|_1 &= \|u^*_T - u\|_1 \\ &= \|(u^* - u)_{S\cap T}\|_1 + \|u^*_{T-S}\|_1 + \|u_{S-T}\|_1 \\ &\leq \|(u^* - u)_{S\cap T}\|_1 + \|u^*_{T-S} - u_{T-S}\|_1 + \|u_{T-S}\|_1 + \|u_{S-T}\|_1 \\ &\leq \|(u^* - u)_{S\cap T}\|_1 + \|u^*_{T-S} - u_{T-S}\|_1 + 2\|u_{T-S}\|_1 \\ &= \underbrace{\|(u^* - u)_S\|_1}_{\text{apply A}} + 2\underbrace{\|u_{T-S}\|_1}_{\text{apply B}} \end{aligned}$$

Choose ϵ appropriately and 1. follows.

Let us first define $\Pi u = v + v''$ and furthermore we write $c = v + \overbrace{v'' + \tilde{e}}^{v'}$. Moreover, we will need the following claims. The proof of Claim C is similar to the proof of Theorem 1 of [BGI+08], and the proofs of Claims D and E can be found in [BIR08].

Claim 5. *We have*

1. *Claim C:*
   $\|v''\|_1 \leq 2\epsilon d \|u\|_1$

2. *Claim D:*
   $median(a+b) \leq quant_{1/4}(a) + quant_{1/4}(b)$

3. *Claim E:*
   Suppose $a_1, \ldots, a_s$ *and* $b_1, \ldots, b_t$ *are positive with* $\{a_1, \ldots, a_s\} \subset \{b_1, \ldots, b_t\}$ *and define* $c(i) = |\{j : a_j \geq b_i\}|$. *Then if*

$$\forall i \quad c(i) \leq \alpha i \implies \|a\|_1 \leq \alpha \|b\|_1$$

Proof of lemma A:

We write $u_i^* = \text{median}(v_{\Gamma(i)} + v'_{\Gamma(i)})$, where $\Gamma(i)$ corresponds to the elements in the expander of $i \in$ "left". Then

$$\begin{aligned}
\|(u - u^*)_S\|_1 &= \sum_{i \in S} |\text{median}(v_{\Gamma(i)} + v'_{\Gamma(i)}) - u_i| \\
&= \sum_{i \in S} |\text{median}(v_{\Gamma(i)} - u_i^d + v'_{\Gamma(i)})| \\
\text{where we have defined:} \quad & u_i^d = \underbrace{(u_i, \ldots, u_i)}_{d \text{ times}} \\
&\leq \sum_{i \in S} \text{median}(|\underbrace{u_{\Gamma(i)} - u_i^d}_{w^i}| + |v'_{\Gamma(i)})|) \\
&\leq \sum_{i \in S} \text{quant}_{1/4}(|w^i|) + \sum_{i \in S} \underbrace{\text{quant}_{1/4}(|v'_{\Gamma(i)}|)}_{\text{c.f. Claim 6}} \\
&\lesssim \sum_{i \in S} \text{quant}_{1/4}(|w^i|) + \frac{\|v'\|_1}{d}.
\end{aligned}$$

Note that Claim C yields $\|v'\|_1 \lesssim 2\epsilon d \|u\|_1 + \|\tilde{e}\|_1$ Due to lack of time, we skipped the bound on $\text{quant}_{1/4}(|w^i|)$. The arguments are essentially very similar and can be found in the paper [BIR08] □

Claim 6 is the central point that makes use of the expansion properties of Π .

Claim 6. *Let* P *be any set of* $s \leq 2k$ *coordinates, then*

$$\sum_{i \in P} quant_{1/4}(|v'_{\Gamma(i)}|) \lesssim \frac{\|v'\|_1}{d}$$

Proof:
Let c(j) be the number of $i \in P$ s.th. i has at least $\frac{d}{4}$ neighbors in the $\{1, \ldots, j\}$. Then we have due to the expansion property

$$\begin{aligned}
c(j)\left(\frac{d}{4} - \epsilon d\right) &\leq j \\
\implies c(j) &\leq \frac{1}{d\left(\frac{1}{4} - \epsilon\right)} j \leq \frac{8}{d} j
\end{aligned}$$

for an appropriately chosen ϵ. Moreover, with $(b_1, \ldots, b_m) = (|v'_1|, \ldots, |v'_m|)$ and $(a_1, \ldots, a_s) = (\text{quant}_{1/4}(|v'_{\Gamma(i)}|))_{i \in P}$ we can use Claim E. $\square$.

3 Introduction to the matrix completion problem

A motivation for the matrix completion problem comes from user ratings of some products which are put into a matrix M. The entries M_{ij} of the matrix correspond to the j'th user's rating of product i. We assume that there exists an ideal matrix that encodes the ratings of all the products by all the users. However, it is not possible to ask every user his opinion about every product. We are only given some ratings of some users and we want to recover the actual ideal matrix M from this limited data. So matrix completion is the following problem

Problem: Suppose you are given some matrix $M \in \mathbb{R}^{n_1 \times n_2}$. Moreover, you also are given some entries $(M_{ij})_{ij \in \Omega}$ with $|\Omega| \ll n_1 n_2$.

Goal: We want to recover the missing elements in M.

This problem is impossible if we don't make any additional assumptions on the matrix M since the missing M_{ij} could in principle be arbitrary. We will discuss a recovery scheme that relies on the following three assumptions.

1. M is (approximately) low rank.
2. Both the columns space and the row space are "incoherent". We say a space is incoherent, when the projection of any vector onto this space has a small ℓ_2 norm.
3. If $M = U \Sigma V^T$ then all the entries of UV^T are bounded.
4. The subset Ω is chosen at random.

Under these assumptions we will show next time, that there exists an algorithm that needs a number of entries in M bounded by $|\Omega| \leq (n_1 + n_2)\, r\, \text{poly}\,(\log(n_1 n_2)) \cdot \mu$. Here μ captures to what extent properties 2 and 3 above hold. One would naturally consider the following recovery method for the matrix M:

$$\begin{aligned} &\text{minimize} \quad \text{rank}(X) \\ &\text{subject to:} \quad X_{ij} = M_{ij} \;\; \forall i, j \in \Omega. \end{aligned}$$

Unfortunately this optimization problem is NP-hard. We will therefore consider the following alternative optimization problem in trace norm:

$$\begin{aligned} &\text{minimize} \quad \|X\|_{tr} \\ &\text{subject to:} \quad X_{ij} = M_{ij} \;\; \forall i, j \in \Omega, \end{aligned}$$

where the trace norm is defined as the sum of the singular values of M, i.e. $\|X\|_{tr} = \sum_i \sigma_i(M)$. This problem is an SDP (semi-definite program), which you will prove on pset 8, and can be solved in time polynomial in $n_1 n_2$.

References

[BGI+08] Radu Berinde, Anna C Gilbert, Piotr Indyk, H Karloff, and Martin J Strauss. Combining geometry and combinatorics: A unified approach to sparse signal recovery. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 798–805. IEEE, 2008.

[BIR08] Radu Berinde, Piotr Indyk, and Milan Ružić. Practical near-optimal sparse recovery in the ℓ_1 norm. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 198–205. IEEE, 2008.

[FR13] Simon Foucart, Holger Rauhut. A Mathematical Introduction to Compressive Sensing. *Applied and Numerical Harmonic Analysis*, 2013.

[GI10] Anna Gilbert and Piotr Indyk. Sparse recovery using sparse matrices. *Proceedings of the IEEE*, 98(6):937–947, 2010.

[IR08] Piotr Indyk and Milan Ružić. Near-optimal sparse recovery in the l1 norm. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 199–207. IEEE, 2008.