CS 294-165 Sketching Algorithms — Fall 2020

Problem Set 3

Due: 11:59pm, Friday November 20th

See homework policy at `http://www.sketchingbigdata.org/fall20/syllabus/`
Each problem is worth the same number of points (except Problem 3).

Problem 1: Michael's counter-example. A matrix Π is said to satisfy the (C, k)-*restricted nullspace property* if for any $\eta \in \ker(\Pi) \setminus \{0\}$ (i.e. $\Pi\eta = 0$), it holds that $\|\eta\|_1 \le C\|\eta_{\bar{S}}\|_1$, where $S \subset [n]$ has size k, $\bar{S}$ is the complement of S, and $\eta_{\bar{S}}$ is the projection of η onto the coordinates in $\bar{S}$. It is known that if Π satisfies the restricted nullspace property for small enough C, then performing measurements according to such a Π is sufficient to recover $\tilde{x}$ satisfying $\|x - \tilde{x}\|_1 \le O(1) \cdot \|x_{tail(k)}\|_1$ (in fact via solving basis pursuit).

Now, it is known that if Π is the (normalized) adjacency matrix of a left-regular bipartite expander with appropriate parameters, then Π satisfies (ε, k)-RIP$_1$: i.e. $\|\Pi x\|_1 = (1 \pm \varepsilon)\|x\|_1$ for all k-sparse x. It is also known that if Π is such an adjacency matrix, it satisfies the $(1 + \varepsilon, k)$-restricted nullspace property, though known proofs of this fact use the expander properties directly rather than going through the RIP$_1$ abstraction. Show that this is necessary, and that the abstraction cannot be used: show that for any pair of constants $k > 1$ and $\varepsilon > 0$, there is an $m \times n$ matrix Π which satisfies (ε, k)-RIP$_1$, but not the null space property with $C < 2$. You may assume in your construction that n is sufficiently large in terms of k and $1/\varepsilon$ if you wish.

Problem 2: Reductions between recovery guarantees An ℓ_2/ℓ_1 *recovery scheme* is a measurement matrix Π together with an algorithm R such that for any $x \in \mathbb{R}^n$,

$$\|x - R(\Pi x)\|_2 \le O(1/\sqrt{k}) \cdot \|x_{tail(k)}\|_1.$$

Show that if (Π, R) is an ℓ_2/ℓ_1 recovery scheme, then there exists an ℓ_1/ℓ_1 recovery scheme (Π, R') satisfying

$$\|x - R'(\Pi x)\|_1 \le O(1) \cdot \|x_{tail(k)}\|_1$$

s.t. the running time to compute $R'(\Pi x)$ is at most $O(n)$ plus the time to compute $R(\Pi x)$. That is to say, achieving the ℓ_2/ℓ_1 recovery guarantee is stronger than ℓ_1/ℓ_1.

Problem 3: (1 point) How much time did you spend on this problem set? If you can remember the breakdown, please report this per problem. (sum of time spent solving problem and typing up your solution)