New York University Tandon School of Engineering
Computer Science and Engineering

CS-GY 9223D: Homework 4.
Due Wed., December 16th, 2020, 11:59pm.

Collaboration is allowed on this problem set, but solutions must be written-up individually. Please list collaborators for each problem separately, or write "No Collaborators" if you worked alone.

Problem 1: Sketches for Cut Estimation

(5 pts) Let $G(V,E)$ be a graph with vertex set V and edge set E. Suppose $|V| = n$ and G has edge vertex incidence matrix $\mathbf{B} \in \mathbb{R}^{m\times n}$. Let $\mathbf{\Pi} \in \mathbb{R}^{k\times m}$ be a random Johnson-Lindenstrauss matrix with k rows. Suppose we sketch G by forming the $O(kn)$ matrix $\mathbf{\Pi B}$.

Given any vertex set $S \subseteq V$, prove that it is possible to estimate $\text{cut}(S, V \setminus S)$ up to $(1 \pm \epsilon)$ error with probability $1-\delta$ using only the information in $\mathbf{\Pi B}$, as long as $k = O(\log(1/\delta)/\epsilon^2)$.

Problem 2: Matrix Concentration from Scalar Concentration

(20 pts) This problem asks you to prove a simplified (and slightly weaker) version of the matrix concentration result used in Lecture 10. Construct a random *symmetric* matrix $R \in \mathbb{R}^{n\times n}$ by setting $R_{ij} = R_{ji}$ to $+1$ or -1, uniformly at random. Prove that, with high probability,

$$\|R\|_2 \leq c\sqrt{n \log n},$$

for some constant c. This is much better than the naive bound of $\|R\|_2 \leq \|R\|_F = n$.

Here are a few hints that might help you along:

- Recall that for a matrix R, $\|R\|_2 = \max_{x\in\mathbb{R}^n} \frac{\|Mx\|_2}{\|x\|_2}$. When R is symmetric, it also holds that $\|R\|_2 = \max_{x\in\mathbb{R}^n} \frac{|x^T Rx|}{x^T x}$.
- Try to first bound $\frac{|x^T Rx|}{x^T x}$ for one particular x – you might want to use a Hoeffding bound, or the Rademacher concentration bound from Lecture 12.
- Then try to extend the result to hold for all x, simultaneously using an ϵ-net argument.

Problem 3: 18th Century Style Compressed Sensing

(10 pts) In Lecture 13 it was mentioned that there exist simpler compressed sensing schemes that work when noise/numerical precision is not an issue. Let $q_1, \ldots, q_n \in \mathbb{R}$ be any set of *distinct* numbers. E.g. we could choose $[q_1, \ldots, q_n] = [1, \ldots, n]$. Consider the sensing matrix $A \in \mathbb{R}^{2k\times n}$:

$$A = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ q_1 & q_2 & q_3 & \ldots & q_n \\ (q_1)^2 & (q_2)^2 & (q_3)^2 & \ldots & (q_n)^2 \\ \vdots & \vdots & \vdots & & \vdots \\ (q_1)^{2k-1} & (q_2)^{2k-1} & (q_3)^{2k-1} & \ldots & (q_n)^{2k-1} \end{bmatrix}$$

This A does not obey any sort of RIP property. Nevertheless, show that, if $x \in \mathbb{R}^n$ is a k sparse vector – i.e. $\|x\|_0 \leq k$ – then we can recover x from Ax. You don't need to give an efficient algorithm. Just argue that for any given $y \in \mathbb{R}^{2k}$, there is at most one k-sparse x such that $y = Ax$. (Hint: Use that a non-zero degree p polynomial cannot have more than p roots. You may also want to use that the column and row rank of a matrix are always equal.)

Problem 4: Sparse Recovery for Dense Vectors – BONUS

(5 pts extra credit) A compressed sensing scheme typically recovers x from a linear sketch Ax whenever x is k-sparse. When x is not k-sparse, there is no guarantee about what is returned. E.g., for the measurement matrix A described above, for any specified k, there exists an algorithm $Decode(y)$ which returns x if $y = Ax$ for a k-sparse x. If $y \neq Ax$ for some k-sparse x, $Decode(y)$ can return anything. In this problem we consider an method that will still return *something useful* when x is not k-sparse.

In particular, your goal is to design a measurement matrix $B \in \mathbb{R}^{c \log n \times n}$, where c is a constant, such that *for any* x (i.e. not necessarily sparse) it is possible to recover a single index/value pair (i, x_i) with $x_i \neq 0$ from Bx with constant probability (e.g. with success probability 9/10). Your algorithm can return *any* (i, x_i) as long as $x_i \neq 0$. **Hint: One possible B takes the form:**

$$B = \begin{bmatrix} AD_0 \\ AD_1 \\ AD_2 \\ \dots \\ AD_s \end{bmatrix}$$

where $D_1, \dots, D_s$ are carefully (and randomly) constructed diagonal matrices and A is the matrix from Problem 3 with $k = O(1)$.

Problem 5: Communicating in the Dark is Easier with Shared Random Coins

(10 pts) Suppose Jesse holds a subset of elements $J \subseteq \{1, \dots, n\}$. Leslie holds another subset $L \subseteq \{1, \dots, n\}$. Jesse and Leslie do not know what elements the other holds. Using as little communication as possible, Jesse wants to figure out if she or Leslie hold any unique elements – i.e. if there is any $j \in J \cup L - J \cap L$.

Show that, for some constant c, Leslie can send Jesse a single message of $O(\log^c n)$ bits that allows her to find such a j if one exists, with constant success probability.

You can assume that Jesse and Leslie decide on a strategy in advance, and that they have access to an unlimited source of shared random bits (e.g. that are published by some third party).

Hint: You might want to use the result from Problem 4. Even if you do not solve Problem 4, you can use the existence of a solution (e.g. a measurement matrix B and a recovery algorithm achieving the described goal.)

This result should surprise you! Even if Leslie *knew* all of Jesse's elements, $O(\log n)$ bits would be needed to communicate if they hold any unique elements. Here we are saying that nearly the same communication complexity can be achieved with *no prior knowledge* of J.