15-859 Algorithms for Big Data — Fall 2022

Problem Set 1

Due: Thursday, September 29, before class

Please see the following link for collaboration and other homework policies:
http://www.cs.cmu.edu/afs/cs/user/dwoodruf/www/teaching/15859-fall22/grading.pdf

Problem 1: Subspace Embeddings for Other Norms (13 points)
In class, we considered the notion of a $(1+\epsilon)$-approximation ℓ_2-subspace embedding, which is random matrix S for which, with probability at least $9/10$, $\|SAx\|_2^2 = (1 \pm \epsilon)\|Ax\|_2^2$ simultaneously for all x. In fact, we showed that if S is a random $s \times n$ Gaussian matrix, then this holds provided that S has $s = O(d/\epsilon^2)$ rows. Suppose we are instead interested in a p-norm subspace embedding, meaning $\|SAx\|_p^p = \sum_{i=1}^{s} |(SAx)_i|^p = (1 \pm \epsilon) \sum_{i=1}^{n} |(Ax)_i|^p = \|Ax\|_p^p$ simultaneously for all x, with say probability at least $9/10$. We will compute p-norm subspace embeddings for even integers $p \geq 2$ as follows:

1. For a vector $y \in \mathbb{R}^d$, let $y^{\otimes p} \in \mathbb{R}^{d^p}$ be the vector with $(i_1, \ldots, i_p)$-th entry equal to $y_{i_1} \cdot y_{i_2} \cdots y_{i_p}$, where $i_\ell \in \{1, 2, \ldots, d\}$ for each $\ell \in \{1, 2, \ldots, p\}$. Show that for any $x \in \mathbb{R}^d$, we have $\langle x^{\otimes p}, y^{\otimes p} \rangle = (\langle x, y \rangle)^p$.

2. For any even integer $p \geq 2$, construct an $n \times d^{p/2}$ matrix M satisfying for all $x \in \mathbb{R}^d$,
$$\|Ax\|_p^p = \|Mx^{\otimes p/2}\|_2^2.$$
Prove that if S is an appropriately scaled Gaussian Matrix with $O(d^{p/2}\epsilon^{-2})$ rows, then with probability $\geq 9/10$, for all x,
$$\|SMx^{\otimes p/2}\|_2^2 = (1 \pm \epsilon)\|Ax\|_p^p.$$
For constant even p, your algorithm to construct the matrix M and compute the matrix product $S \cdot M$ should run in time $n \cdot \mathrm{poly}(d)$.

3. An $n \times d$ matrix A is a Vandermonde matrix if for each row A_i, there is a $y_i \in \mathbb{R}$, for which $A_i = (1, y_i, y_i^2, y_i^3, \ldots, y_i^{d-1})$. For a vector $x \in \mathbb{R}^d$, note that $\langle A_i, x \rangle = \sum_{j=0}^{d-1} x_j y_i^j$ is the degree-$(d-1)$ polynomial with coefficient vector x and evaluated at the point y_i. Show that for the special case of a Vandermonde matrix and any constant even integer p, if S is an appropriately scaled Gaussian matrix with $O(dp/\epsilon^2)$ rows, then with probability $\geq 9/10$, for all x,
$$\|SMx^{\otimes p/2}\|_2^2 = (1 \pm \epsilon)\|Ax\|_p^p.$$

Problem 2: Randomized Rounding for Sparsification (12 points)
Consider the following algorithm to sparsify an $n \times n$ input matrix A:

1. Let $\hat{A} = A$ and replace all entries of $\hat{A}$ that are smaller in absolute value than $\epsilon/2n$ with 0. You can assume that A has at least one entry with absolute value $\geq \epsilon/2n$.
2. Set $s = O(\|A\|_F^2 \cdot n\epsilon^{-2} \ln n)$
3. For $t = 1, \ldots s$, randomly sample indices (i_t, j_t) of $\hat{A}$ with for all i, j,
$$\Pr[(i_t, j_t) = (i, j)] = \frac{(\hat{A}_{i,j})^2}{\|\hat{A}\|_F^2} =: p_{i,j}.$$
4. Output $\tilde{A} = \frac{1}{s} \sum_{t=1}^{s} \frac{\hat{A}_{i_t, j_t}}{p_{i_t, j_t}} e_{i_t} e_{j_t}^T$, where e_i is the i-th standard unit vector in $\mathbb{R}^n$.

Argue that with probability at least $1 - 1/n$,
$$\|A - \tilde{A}\|_2 \leq \epsilon.$$

As a hint, it may help to bound $\|A - \hat{A}\|_2$ and $\|\hat{A} - \tilde{A}\|_2$ separately and apply the triangle inequality. It may also be helpful to apply the matrix Chernoff bound that we learned in class to bound $\|\hat{A} - \tilde{A}\|_2$. See the generalization of symmetric Chernoff bound we saw in class to the case of arbitrary rectangular random matrices here (`https://en.wikipedia.org/wiki/Matrix_Chernoff_bound#Matrix_Bennett_and_Bernstein_inequalities`).

Problem 3: Sparse Deterministic Matrix Product (13 points)
Given $n \times n$ matrices A and B, we define sparse matrices $\bar{A}$ and $\bar{B}$ as follows. For each row A_{i*} of A, for each entry $A_{i,k}$, if $|A_{i,k}| > \frac{\epsilon}{2}\|A_{i*}\|_1$, then set $\bar{A}_{i,k} = A_{i,k}$, otherwise set $\bar{A}_{i,k} = 0$. Here for a vector y, $\|y\|_1 = \sum_{i=1}^{d} |y_i|$, where d is the dimension of y.

Similarly, for each column B_{*j} of B, for each entry $B_{k,j}$, if $|B_{k,j}| > \frac{\epsilon}{2}\|B_{*j}\|_1$, then set $\bar{B}_{k,j} = B_{k,j}$, otherwise set $\bar{B}_{k,j} = 0$. Show that
$$\|AB - \bar{A} \cdot \bar{B}\|_1 \leq \epsilon\|A\|_1 \cdot \|B\|_1,$$
where for an $n \times n$ matrix C, $\|C\|_1 = \sum_{i=1}^{n} \|C_{i*}\|_1$.

As a hint, you may want to first establish a similar claim for vectors rather than matrices.

Problem 4: Computing the Best Cost Regression (12 points)
Given an $n \times d$ matrix A with $n \geq d$, as well as an $n \times m$ matrix B, you would like to estimate: $\min_{x,i} \|Ax - B_{*i}\|_2$ up to a multiplicative factor of 2, where B_{*i} is the i-th column of B. Show how to solve this problem with probability $\geq 9/10$ in $(\mathrm{nnz}(A) + \mathrm{nnz}(B) + d^4)\mathrm{poly}(\log m)$ time.

As a hint, try to use one of the Sketching Distributions discussed in the class to approximate the optimal cost of one of the regression problems with high probability and then use a union bound to simultaneously approximate the optimal costs of all the regression problems. Your algorithm may have to compute column norms of a matrix that cannot be computed in the given time budget—think how to instead just approximate the column norms quickly.