15-859 Algorithms for Big Data — Fall 2022

Problem Set 3

Due: Tuesday, November 8, 11:59pm

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: ℓ_1-Median Subspace Embedding (25 points)
In class we saw an ℓ_1-subspace embedding for an $n \times d$ matrix A using a random $m \times n$ matrix of i.i.d. Cauchy random variables (divided by m i.e., each entry of S is a standard Cauchy random variable divided by m) S with $m = O(d \log d)$. Namely, with probability at least $9/10$,

$$\Omega(\|Ax\|_1) = \|SAx\|_1 = O(d \log d)\|Ax\|_1, \tag{1}$$

simultaneously for all vectors $x \in \mathbb{R}^d$. Unfortunately, the $O(d \log d)$ factor can be large for some applications. In this problem we will improve this factor by instead looking at the *median operation*. Namely, let S be an $m \times n$ matrix of i.i.d. Cauchy random variables (divided by m) for $m = O(d \log(d/\epsilon)/\epsilon^2)$. Show that with probability at least $9/10$,

$$\frac{1-\epsilon}{m}\|Ax\|_1 \le \|SAx\|_{\text{med}} \le \frac{1+\epsilon}{m}\|Ax\|_1, \tag{2}$$

simultaneously for all $x \in \mathbb{R}^d$. Here for a vector y, $\|y\|_{\text{med}}$ denotes the median of the absolute values of its entries. You can assume that $1/\text{poly}(d) < \epsilon < c$ for a small enough constant c.

To help you prove this statement, here are some properties you may find useful:

1. You can continue to assume that (1) holds even with $m = O(d \log(d/\epsilon)/\epsilon^2)$.
2. It suffices to prove (2) for any basis of A that we would like. Choose a special kind of basis that we used in class for ℓ_1.
3. It may be helpful to use (1) to argue that $\|SAx\|_\infty \le \text{poly}(d)\|Ax\|_1$ for all x.
4. A γ-net for the unit ℓ_1-ball, intersected with the column span of A, has size at most $\gamma^{O(d)}$. You can use this fact without proof, as it follows a similar volume argument that we used in class for ℓ_2-nets.
5. It will be useful to define the notion of S being *good* for a vector Ax. Say that SAx is good if both

   $$|\{i \text{ such that } |(SAx)_i| < (1-\epsilon)\|Ax\|_1/m\}| \le \left(\frac{1}{2} - C\epsilon\right)m,$$

   and

   $$|\{i \text{ such that } |(SAx)_i| > (1+\epsilon)\|Ax\|_1/m\}| \le \left(\frac{1}{2} - C\epsilon\right)m,$$

   where $C > 0$ is a certain constant. Try to argue that the mapping S is good on all net vectors, and try to use this to conclude (2) holds for all vectors x.

HINT: Fix a vector x and try to show that with high probability $(1-\epsilon)\|Ax\|_1/m \le \|SAx\|_{\text{med}} \le (1+\epsilon)\|Ax\|_1/m$. Use the 1-stability property of the Cauchy Random variables and then show that median of independent Cauchy Random variables is highly concentrated. Next extend the argument using a union bound to all the net vectors and then to all the vectors.

Problem 2: F_2-Difference Estimator in a Stream (25 points)
In this problem we consider insertion-only streams, meaning that we just see positive updates to an underlying vector $x \in \{0, 1, 2, \ldots, M\}^n$ for some $M = \text{poly}(n)$. That is, no negative changes to coordinates of x are allowed. We will consider estimating $\|x\|_2^2$ up to a $(1+\epsilon)$-multiplicative factor.

Often x is *slowly-changing*, meaning that at some point in the stream $x = u$ and then at some later point in the stream $x = u + v$ for some $v \in \{0, 1, 2, \ldots, M\}^n$, and we have $\|u+v\|_2^2 - \|u\|_2^2 \leq \gamma\|u\|_2^2$ and $\|v\|_2^2 \leq \gamma\|u\|_2^2$ for some $0 < \gamma < 1$. You would like to estimate $\|u+v\|_2^2$ using your previous estimate $\|u\|_2^2$ and using very little space. Show how to estimate $\|u+v\|_2^2$ up to a $(1 \pm \epsilon)$-multiplicative factor with probability at least $9/10$, given a $(1 \pm \epsilon/2)$-approximation to $\|u\|_2^2$ and a sketch which uses space $O(\gamma\epsilon^{-2}\log n)$ bits. You can assume that you initialize $x = 0$ and after processing some number of updates you will have $x = u$ at which point you are given a $1 \pm \epsilon/2$ approximation to $\|u\|_2^2$. Then you start processing the rest of the updates and finally end up setting $x = u + v$ and at the end of the stream you want to output a $1 \pm \epsilon$ approximation to $\|u+v\|_2^2$.

Note that if $\gamma = \Theta(1)$, then there is no improvement over just estimating $\|u+v\|_2^2$ directly using the sketch from class. However, if for example, $\gamma = \Theta(\epsilon)$, then the memory is only $O(\epsilon^{-1}\log n)$ bits, which is a significant savings.

HINT: Treat the number of rows in your sketch as a variable and see what approximation guarantees you will get in terms of the number of rows. Then set the number of rows so as to obtain a $1 \pm \epsilon$ approximation.