15-851 Algorithms for Big Data — Spring 2025

Problem Set 3 Solutions

Problem 1: Geometric Mean Estimator for ℓ_1-Estimation

1. From class we have the property that if we have $\sum_i a_i \cdot C_i$ where each C_i is an i.i.d. Cauchy random variable, then this is distributed as $|a|_1 \cdot C$.

   So, we have that $(Sx)_a = |x|_1 \cdot C_1$, $(Sx)_b = |x|_1 \cdot C_2$, and $(Sx)_c = |x|_1 \cdot C_3$ where C_1, C_2, and C_3 are independent Cauchy random variables. We therefore have that

   $$\mathbf{E}[F_i] = |x|_1 \cdot \mathbf{E}[|C_1 C_2 C_3|^{1/3}].$$

   So we have to show that $\mathbf{E}[|C_1C_2C_3|^{1/3}]$ is a finite scalar that does not depend on x. Clearly $\mathbf{E}[|C_1C_2C_3|^{1/3}]$ does not depend on x. Note that $\mathbf{E}[|C_1C_2C_3|^{1/3}] = \mathbf{E}[|C_1|^{1/3}] \cdot \mathbf{E}[|C_2|^{1/3}] \cdot \mathbf{E}[|C_3|^{1/3}]$ since C_1, C_2, and C_3 are independent. We now need to show that it is finite. We have that

   $$\mathbf{E}[|C_1C_2C_3|^{1/3}] = \int\int\int \frac{|z_1z_2z_3|^{1/3}}{\pi^3(1+z_1^2)(1+z_2^2)(1+z_3^2)} dz_1dz_2dz_3$$

   which converges to some constant that is strictly greater than 0.

2. Again we have that

   $$F_i = ||x|_1^3 C_1C_2C_3|^{1/3} = |x|_1 \cdot |C_1C_2C_3|^{1/3}.$$

   We have that

   $$\mathbf{Var}(F_i) = |x|_1^2 \cdot \mathbf{Var}(|C_1C_2C_3|^{1/3}).$$

   By definition, we know that $\mathbf{Var}(|C_1C_2C_3|^{1/3}) = \mathbf{E}[|C_1C_2C_3|^{2/3}] - \mathbf{E}[|C_1C_2C_3|^{1/3}]^2 \leq \mathbf{E}[|C_1C_2C_3|^{2/3}]$. So we have

   $$\mathbf{Var}(|C_1C_2C_3|^{1/3}) \leq \int\int\int \frac{|z_1z_2z_3|^{2/3}}{\pi^3(1+z_1^2)(1+z_2^2)(1+z_3^2)} dz_1dz_2dz_3.$$

   This again converges to a constant.

3. Using the first part, we can see that

   $$\mathbf{E}[F] = \frac{3}{Ck}\sum_{i=1}^{k/3} \mathbf{E}[F_i] = \frac{3}{Ck} \cdot \frac{k}{3} \cdot C \cdot |x|_1 = |x|_1.$$

   Using the second part, we also have

   $$\mathbf{Var}[F] = \frac{9}{C^2k^2}\sum_{i=1}^{k/3} \mathbf{Var}[F_i] \leq \frac{9}{C^2k^2} \cdot \frac{k}{3} \cdot O(|x|_1^2) = O(|x|_1^2/k).$$

   Note that we can move the variance into the summation because all the F_i's are independent.

   Now, using Chebyshev's inequality will give us the result.

   $$\mathbf{Pr}(|F - |x|_1| \geq \varepsilon \cdot |x|_1) \leq \frac{O(|x|_1^2/k)}{|x|_1^2 \cdot \varepsilon^2} \leq 1/10$$

   for appropriate $k = O(\varepsilon^{-2})$.

Problem 2: Online Leverage Score Sampling plus Merge and Reduce

1. Let us say that the approximation factor we use in each coreset is γ. At each level of the tree, we suffer an additional multiplicative $(1 \pm \gamma)$ error in our approximation. So, since the height of the tree is $\Theta(\log(n/2k))$, we have that the final coreset is a $(1 \pm \log(n/2k) \cdot \gamma)$ multiplicative approximation.

   Recall that we could assume that $k < n^{0.9}$. Therefore, we have that $\log(n/2k) = \Theta(\log n)$, so we have that the final coreset is a $(1 \pm \log n \cdot \gamma)$ multiplicative approximation. Now, we want $\log n \cdot \gamma \leq \varepsilon$ to get the result. So, we should set $\gamma \leq \varepsilon / \log n$, giving us $k = O(d \log^2 n/\varepsilon^2)$.

   We now analyze the space. As per the problem statement, a coreset can be constructed in $O(kd \log n)$ space. In addition, storing a coreset requires $O(kd \log n)$ space since each row has d entries that each take $\log n$ bits to store. We keep at most one coreset per level of the binary tree at one time. So, the total space is
$$O(kd \log^2 n) = O(d^2 \log^4 n/\varepsilon^2).$$

2. (a) Let us maintain a $d \times d$ matrix M. Upon seeing a row r in the stream, you can store it exactly in $O(d \log n)$ space. Then we can compute $r^\intercal r$ using $O(d^2 \log n)$ space, and take $M = M + r^\intercal r$. Storing M only uses $O(d^2 \log n)$ space, and $M = A_{i-1}^\intercal A_{i-1}$.

   So, given the next row in the stream, we can exactly compute ℓ_i.

   (b) We will first show that the online leverage score upper bounds the actual leverage score of matrix $A' = [A; \sqrt{\lambda} I]$.

   We have that
$$\ell_i = \min(a_i^\intercal (A_{i-1}^\intercal A_{i-1} + \lambda I)^{-1} a_i, 1)$$
   and
$$\tau_i(A') = a_i^\intercal (A^\intercal A + \lambda I)^{-1} a_i.$$
   We want to show that
$$a_i^\intercal (A_{i-1}^\intercal A_{i-1} + \lambda I)^{-1} a_i \geq a_i^\intercal (A^\intercal A + \lambda I)^{-1} a_i.$$
   If we prove that $A^\intercal A + \lambda I \succeq A_{i-1}^\intercal A_{i-1} + \lambda I$, then we also have $(A_{i-1}^\intercal A_{i-1} + \lambda I)^{-1} \succeq (A^\intercal A + \lambda I)^{-1}$ as per the hint. This would give us what we want. We first see that
$$A^\intercal A = A_{i-1}^\intercal A_{i-1} + \sum_{j=i}^{n} a_j^\intercal a_j.$$
   Since for each j we have that $a_j^\intercal a_j$ is PSD, then we have that
$$A^\intercal A \succeq A_{i-1}^\intercal A_{i-1}.$$
   Now we can use the sampling without replacement bound on A'. As per the hint given on piazza, we have to sample each row of A' (not A). However, we can simply set p_i for all rows of A' that correspond to $\sqrt{\lambda} I$ to 1. Therefore, we simply add these rows to our sample.

   In the sampling without replacement bound, for $i \in [n]$, we take $u_i = \ell_i$ and for $i \in (n, n+d]$ we take $u_i = 1$. We also set $\gamma = \varepsilon/3$. So, we have that simultaneously for all x we have
$$\|TA'x\|_2^2 = (1 \pm \varepsilon/3)\|A'x\|_2^2.$$
   This gives us
$$\|TAx\|_2^2 + \lambda\|x\|_2^2 = (1 \pm \varepsilon/3)(\|Ax\|_2^2 + \lambda\|x\|_2^2).$$

   (c) We first prove that $\|Ax\|_2^2/\|x\|_2^2 \geq \sigma_{\min}^2$. Take the SVD of A. This gives us
$$\|Ax\|_2^2 = \|U\Sigma V^\intercal x\|_2^2 = \|\Sigma V^\intercal x\|_2^2.$$

Let us set $y = V^\intercal x$. So we further have

$$\|\Sigma V^\intercal x\|_2^2 = \|\Sigma y\|_2^2 = \sum_i \sigma_i^2 y_i^2 \geq \sigma_{\min}^2 \sum_i y_i^2 = \sigma_{\min}^2 \|y\|_2^2.$$

So we finally have

$$\frac{\|Ax\|_2^2}{\|x\|_2^2} \geq \frac{\sigma_{\min}^2 \|y\|_2^2}{\|x\|_2^2} = \frac{\sigma_{\min}^2 \|V^\intercal x\|_2^2}{\|x\|_2^2} = \frac{\sigma_{\min}^2 \|x\|_2^2}{\|x\|_2^2}.$$

So, we have by the previous part that

$$\|TAx\|_2^2 = (1 \pm \varepsilon/3)(\|Ax\|_2^2 \pm \varepsilon \sigma_{\min}^2(A)\|x\|_2^2) = (1 \pm \varepsilon/3)(\|Ax\|_2^2 \pm \varepsilon \|Ax\|_2^2).$$

This gives us the desired result.

(d) Notice that as proven in the previous part, our space usage to achieve a $(1 \pm \varepsilon)$ approximation after the merge-and-reduce scheme is $O(d^2 \log n/\varepsilon^2)$ times $\operatorname{poly}\log(r/2k)$ where r is the number of rows that are being passed in the stream. Recall that the number of rows is equivalent to the number of nonzeros in T. This is, by the sampling without replacement result, the number of rows, r, is

$$O(\varepsilon^{-2} \cdot \log d \cdot d \log(\|A\|_2^2/\lambda)).$$

Plugging in $\lambda = \varepsilon \cdot \sigma_{\min}^2$, we get that the number of nonzeros in T is

$$O(\varepsilon^{-2} \cdot \log d \cdot d \log(\kappa/\varepsilon)).$$

We know that k has a factor of d and ε^{-2}, so we have that

$$\operatorname{poly}\log(r/2k) = O(\operatorname{poly}\log(\log d \log(\kappa/\varepsilon))).$$

This gives us the final space result. For the error guarantee, we can simply combine parts 1 and $2c$.