

Due: Friday 2/6, 11:59pm on Gradescope

Please follow the homework policies on the course website.

1. (8 pt.) **Processes and CPUs**

Suppose that in a distributed system, we have N CPUs and P processes. Each process is independently and uniformly allocated to a CPU. However, if multiple processes are allocated to the same CPU, the CPU will choose one of them at random to complete; the remaining processes allocated to that CPU will not be completed.

- (a) (2 pt.) What is expected number of processes that will be completed?
- (b) (6 pt.) Suppose that $P \geq N$, and denote the total number of completed processes by C . Use **Poissonization** to prove that $\Pr[C \leq \frac{N}{2}] \leq e^{-\Omega(N)}$.

[**Note:** There may be ways to do this problem that don't involve Poissonization, but we want you to use it to get practice with it. That is, you should prove the statement by analyzing the case when the number of processes is an appropriate Poisson random variable. Don't forget the de-Poissonization step!]

- (c) (0 pt.) [**Optional: this won't be graded.**] Let μ be your answer from part (a). Under what conditions on P and N can you use Poissonization to show that $\Pr[C \leq (1 - \delta)\mu] \leq e^{-\Omega_\delta(N)}$ for any $\delta > 0$, where the Ω_δ notation means that you are allowed to have constants that depend on δ hidden inside the big-Omega.

- 2. (4 pt.) [**Bourgain's embedding for ℓ_p .**] We showed that Bourgain's embedding allows us to embed an arbitrary metric space (X, d) with $|X| = n$ into (\mathbb{R}^k, ℓ_1) with target dimension k being $O((\log n)^2)$ and distortion being $O(\log n)$. Moreover, the embedding can be computed efficiently using a randomized algorithm. Prove that the exact same embedding computed by the randomized algorithm also achieves $O(\log n)$ distortion with high probability when the target metric is ℓ_p for $p > 1$. We encourage you to emphasize only the differences from the proof in the lecture notes rather than copying the entire proof.

[**HINT:** Let $f : X \rightarrow \mathbb{R}^k$ denote the relevant embedding. For any two points $x, y \in X$, we showed that $\|f(x) - f(y)\|_1 \leq k \cdot d(x, y)$. Can we say something similar about $\|f(x) - f(y)\|_p$?]

[**HINT:** For any two points $a, b \in \mathbb{R}^k$ and $p > 1$, it holds that $\|a - b\|_p \geq k^{(1/p)-1} \|a - b\|_1$. This is a special case of Hölder's inequality.]

3. [**Fast approximate regression**] In this problem, we will see an application of (fast¹) JL to numerical linear algebra; in particular, to approximate regression.

Recall the standard *regression* problem: given $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$ (with $n \geq d$, so that A has full column rank), the goal is to find

$$x^* = \operatorname{argmin}_{x \in \mathbb{R}^d} \|Ax - b\|_2^2.$$

Traditional solvers for this take time about $O(nd^2)$ when $n \gg d$.

In this problem we will find a faster algorithm to find an \hat{x} that is an *approximately* optimal solution, with high probability.

- (a) (6 pt.) Consider the following definition:

Definition 1 (Subspace Embedding). *A distribution on matrices $\Phi \in \mathbb{R}^{m \times n}$ is a (ϵ, δ) -subspace embedding for subspaces of dimension k if the following holds:*

For any k -dimensional subspace $U \subseteq \mathbb{R}^n$, with probability at least $1 - \delta$ over the choice of Φ ,

$$(1 - \epsilon)\|u\|_2^2 \leq \|\Phi u\|_2^2 \leq (1 + \epsilon)\|u\|_2^2$$

for all $u \in U$.

Suppose that Φ is a random matrix in $\mathbb{R}^{m \times n}$ so that:

- The distribution of Φ is an $(\epsilon = 0.01, \delta = 0.01)$ subspace embedding for subspaces of dimension k .
- $m = O(k)$.
- For any vector $y \in \mathbb{R}^n$, you can do the matrix-vector multiplication² Φy in time $O(n \log n)$, assuming that $k \ll n^{1/3}$.

Use Φ to solve the following problem: Suppose that $d \ll n^{1/3}$. Design an algorithm that does the following:

- Given $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$, find $\hat{x} \in \mathbb{R}^d$ so that, with probability at least 0.99,

$$\|A\hat{x} - b\|_2^2 \leq 1.1 \min_{x \in \mathbb{R}^d} \|Ax - b\|_2^2$$

- Your algorithm should run in time $O(d^3 + dn \log n) = O(dn \log n)$. In particular, this is faster than $O(nd^2)$ if $d \gg \log n$.

Clearly state your algorithm, and prove that it has the desired properties, using the properties of Φ . You may assume that you can solve $\tilde{n} \times \tilde{k}$ regression problems in time $O(\tilde{n}\tilde{k}^2)$ for any \tilde{n}, \tilde{k} .

¹You can also do a similar application with sparse JL.

²Note that you don't store Φ explicitly; you can sample and store a small representation of it very quickly, and then when it's time to do a matrix-vector multiplication, you can use your small representation to do that in time $O(n \log n)$.

- (b) (0 pt.) [Optional: this part will not be graded.] In this optional part, you will come up with the distribution Φ from part (a). The starting point is *fast* JL transforms. We mentioned in the lectures that there are JL transforms that are faster to multiply by than a dense Gaussian matrix. Here's a formal statement about such a Fast JL transform. (Notice that this gives a guarantee about how much Φ distorts any particular vector x . To get a JL guarantee for all pairs of points x, y in a set X , take a union bound on top of this statement).

Theorem 1 (Implicit in (Ailon, Chazelle, 2009)). *Fix $n > m$. There is a distribution on matrices $\Phi \in \mathbb{R}^{m \times n}$ so that the following both hold:*

- *There is a constant c so that for all $\epsilon > 0$ and any $x \in \mathbb{R}^n$,*

$$\Pr_{\Phi}[(1 - \epsilon)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \epsilon)\|x\|_2^2] \geq 1 - 2 \exp(-cm\epsilon^2).$$

- *For $x \in \mathbb{R}^n$, a matrix-vector multiplication of the form $\Phi \cdot x$ can be done in time $O(n \log n + \min\{nm, m^3\})$.*

Show (using appropriate constraints on the values of n and d) that the distribution on matrices Φ in Theorem 1 satisfies the requirements on Φ in part (a).

[**HINT:** You may want to look at the “Bonus” material on sparsity that we are skipping this quarter; it’s available on the course website. In particular, try making an ϵ -net of a fixed subspace U and taking a union bound over all points in the net, mimicking what we did to prove that our JL matrix has the RIP.]