Homework 4

Algorithms for Big Data
CS498ABD Spring 2019
Due: 10am, Friday, April 19th

Instructions:

- Unlike previous homeworks, you need only do **3 out of the 4** problems. (Of course you're encouraged to try and welcome to submit all 4!)
- Each home work can be done in a group of size at most two. Only one home work needs to be submitted per group. However, we recommend that each of you think about the problems on your own first.
- Homework needs to be submitted in pdf format on Gradescope. See `https://courses.engr.illinois.edu/cs374/fa2018/hw-policies.html` for more detailed instructions on Gradescope submissions.
- Follow academic integrity policies as laid out in student code. You can consult sources but cite all of them including discussions with other class mates. Write in your own words. See the site mentioned in the preceding item for more detailed policies.

Problem 1. Fast JL Recall the JL Lemma where we pick a random $m \times n$ matrix Π and show that for $m = O(1/\epsilon^2)$, with at least $2/3$ probability,

$$(1-\epsilon)\|x\|_2^2 \leq \|\Pi x\|_2^2 \leq (1+\epsilon)\|x\|_2^2. \tag{1}$$

- Imagine picking Π as follows: for each $i \in \{1, \ldots, n\}$ we pick a uniformly random number $h_i \in \{1, \ldots, m\}$. We then set $\Pi_{h_i,i} = \pm 1$ for each $i \in \{1, \ldots, n\}$ (the sign is chosen uniformly at random from $\{-1, 1\}$), and all other entries of Π are set to 0. This Π has the advantage that in turnstile streams, we can process updates in constant time. Show that using this Π still satisfies the conditions of Equation 1 with $2/3$ probability for $m = O(1/\epsilon^2)$.
- Show that the matrix Π from the first part can be specified using $O(\log n)$ bits such that Equation 1 still holds with at least $2/3$ probability, and so that given any $i \in \{1, \ldots, n\}$, $\Pi_{h_i,i}$ and h_i can both be calculated in constant time. *Hint:* Use limited independence hash functions to generate the h_i.

Exercise 2: Improved net argument for subspace embeddings Recall that in oblivious subspace embeddings we want to preserve lengths of all vectors in a subspace of dimension d (assuming vectors are in dimenstion R^n where $n > d$). For this we showed that a JL matrix with $m = O(d/\epsilon^2)$ rows suffices via a net argument. More formally the claim is that there is a fixed set Q of $\exp(O(d))$ vectors such that preserving their lengths to a $(1 \pm \epsilon)$ factor suffices to preserve lengths of all vectors in that subspace (we then use a union bound). In lecture we describe a construction that yielded a net of size $\exp(d \log d)$ which is weaker. In this problem you will see the stronger bound via the following two parts.

- Define $Q_\gamma = \{w : w \in \frac{\gamma}{\sqrt{d}}\mathbb{Z}^d, \|w\|_2 \leq 1\}$ for $\gamma \in (0, 1)$. Prove $|Q_\gamma| \leq e^{d \cdot f(\gamma)}$ for some function $f(\gamma)$ (which needn't be optimized).

  Hint: Given $z \in Q_\gamma$ define a cube C_z centered at z with side length $\gamma/\sqrt{d}$. Note these cubes are all disjoint, then use a volume argument (you may use that an ℓ_2 ball of radius r in $\mathbb{R}^d$ has volume $(C_d \cdot r/\sqrt{d})^d$ for some constant C_d which is $\Theta(1)$ as d grows).

- Show that if for some $A \in \mathbb{R}^{d \times d}$ we have $|u^T A v| \leq \epsilon$ for all $u, v \in Q_\gamma$, then $|x^T A x| \leq \epsilon/(1-\gamma)^2$ for all $x \in \mathbb{R}^d$ of unit ℓ_2 norm.

  Hint: Write $y = (1-\gamma)x$ and round down the coordinates of y to obtain $z \in Q_\gamma$. Argue that $y \in C_z$ and use that any point in a convex polytope can be written as a convex combination of the vertices of that polytope.

- Finish up the details to argue that JL matrix with $m = O(d/\epsilon^2)$ rows is yields an oblivious subspace embedding with constant probability.

Exercise 3: LSH for Hamming Distance In class we saw an LSH scheme for nearest neighbor search for n binary strings of length d in the hamming distance metric. The scheme was based on a decision version where for a given r the data structure would be able to answer with good probability whether there is a point in the data base with distance at most r from q or whether every point is at least $(1+\epsilon)r$. The final data structure is composed of $O(\log d/\epsilon)$ data structures for different values of r. Do we need this reduction to the decision version? Read Charikar's paper on similarity search for a variant of the basic scheme that avoids this in a simple way. Describe and analyze the scheme in your own words.

Problem 4. Matchings with additional constraint We saw an algorithm in the semi-streaming model for finding a constant factor approximation to the maximum cardinality and maximum weight matching problem. Now consider the following variant. We are given a graph $G = (V, E)$. Moreover each edge has a color from $\{1, 2, ..., k\}$ and each color i has an integer upper bound b_i. The goal is to find a maximum cardinality matching M which satisfies the additional constraint that the number of edges in M from a color class i is at most b_i. Assume that you are given the b_i values ahead of time and the each edge when it arrives in the stream specifies its end points and its color. Describe a constant factor approximation for this problem in the semi-streaming setting. **Extra credit:** Develop a constant factor for the weighted case.