15-451 Algorithms, Spring 2019
Practice Problems

Weighted Multiplicative Spanners. We saw a greedy algorithm for finding a multiplicative spanner of an unweighted graph in lecture. Recall a k-multiplicative spanner $H = (V, E')$ of a given unweighted graph $G = (V, E)$ on n nodes, is a subgraph (so $E' \subseteq E$) for which for all pairs u, v of vertices in V, $d_G(u, v) \leq d_H(u, v) \leq k \cdot d_G(u, v)$. In this problem we will find a multiplicative spanner $H = (V, E')$ in a *weighted* graph $G = (V, E)$ on n nodes, where each edge $e \in E$ has a positive edge weight w_e. Consider the following algorithm:

1. Initialize E' to $\emptyset$
2. Let $E = \{e_1 = \{u_1, v_1\}, e_2 = \{u_2, v_2\}, \ldots, e_m = \{u_m, v_m\}\}$ be such that
$$w_{e_1} \leq w_{e_2} \leq w_{e_3} \leq \cdots \leq w_{e_m}.$$
3. For $i = 1, 2, \ldots, m$,
   (a) If the distance between u_i and v_i in $H = (V, E')$ is more than $k \cdot w_e$, then add the edge e_i to E', otherwise discard the edge.
4. Output $H = (V, E')$.

1. Argue that H is a k-multiplicative spanner.

   Solution: Consider any pair u, v of vertices in V. For H to be a k-multiplicative spanner, it must be that $d_H(u, v) \leq k \cdot d_G(u, v)$ (note that trivially $d_H(u, v) \geq d_G(u, v)$ for all u, v). Let $P = (e_{i_1}, e_{i_2}, \ldots, e_{i_r})$ be an arbitrary shortest path in G between u and v. Then for each edge $e_{i_j} = \{u_{i_j}, v_{i_j}\}$ along P, either $e_{i_j} \in E'$ and so $d_H(u_{i_j}, v_{i_j}) \leq w_{e_{i_j}}$ (in fact, equality holds, as otherwise there would be a shorter path from u to v), or $d_H(u_{i_j}, v_{i_j}) \leq k w_{e_{i_j}}$ by definition of the algorithm. Since $d_G(u, v) = \sum_{j=1}^{r} w_{e_{i_j}}$, it follows that $d_H(u, v) \leq \sum_{j=1}^{r} k \cdot w_{e_{i_j}}$. Since u, v were arbitrary, it follows that H is a k-multiplicative spanner.

2. Argue that for any choices of the weights w_e, the girth (minimum cycle length) of H is at least $k + 2$.

   Solution: Suppose the girth were at most $k + 1$, and consider the last edge $e = \{u, v\}$ the algorithm adds to H along some cycle C of length at most $k + 1$. Since the algorithm added e to H, it must have been that before adding e, $d_H(u, v) > k \cdot w_e$. Since we process the edges of G in non-decreasing order of weights though, each of the edges in $C \setminus \{e\}$ has weight at most w_e. Consequently, $C \setminus \{e\}$ must have at least $k + 1$ edges, as otherwise the path from u to v along $C \setminus \{e\}$ would have total weight at most $k \cdot w_e$, a contradiction. But this implies C is a cycle of length at least $k + 2$, a contradiction.

3. What is an upper bound on the number of edges in H?

   Solution: From lecture, if $k = 2t$ or $k = 2t - 1$, a graph with girth at least $k + 2$ has at most $O(n^{1+1/t})$ edges.

The Variance of CountSketch. Recall in lecture we introduced the CountSketch, which is a random linear map S from $\mathbb{R}^n$ to $\mathbb{R}^k$, for $k = \Theta(1/\epsilon^2)$, defined as follows. Let $h : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, k\}$ be a 2-wise independent hash function, and $\sigma : \{1, 2, \ldots, n\} \to \{-1, 1\}$ be a 4-wise independent hash function. Then for $i = 1, 2, \ldots, k$, we have $(Sx)_i = \sum_{j \text{ s.t. } h(j)=i} \sigma(j) x_j$, where x is the n-dimensional input vector.

In lecture, we showed $\mathbf{E}[\|Sx\|^2] = \|x\|^2$, and claimed that $\mathbf{Var}[\|Sx\|^2] = O(\|x\|^4/k)$. We saw that these statements, by Chebyshev's inequality, imply $\mathbf{Pr}[|\|Sx\|^2 - \|x\|^2| > \epsilon\|x\|^2] \le \frac{1}{10}$.

Prove that $\mathbf{Var}[\|Sx\|^2] \le \frac{2}{k}\|x\|_2^4$.

Solution: Write $\|Sx\|^2 = \sum_{j=1}^k \left(\sum_{i=1}^n \delta(h(i) = j)\sigma(i)x_i\right)^2$, where $\delta(h(i) = j)$ is 1 if $h(i) = j$, otherwise $\delta(h(i) = j)$ is 0. We are interested in $\mathbf{Var}[\|Sx\|^2] = \mathbf{E}[(\|Sx\|^2)^2] - (\mathbf{E}[\|Sx\|^2])^2$, and already know from lecture that $(\mathbf{E}[\|Sx\|^2])^2 = \|x\|_2^4$, where $\|x\|_2^2 = \sum_{i=1}^n x_i^2$. We bound $\mathbf{E}[\|Sx\|^4]$. Write $\|Sx\|^4$ as:

$$\|Sx\|^4 = \left(\sum_{j=1}^k \left(\sum_{i=1}^n \delta(h(i) = j)\sigma(i)x_i\right)^2\right)^2,$$

which, after expanding the squares, is:

$$\sum_{j_1,j_2=1}^k \sum_{i_1,i_2,i_3,i_4=1}^n \delta(h(i_1) = j_1)\delta(h(i_2) = j_1)\delta(h(i_3) = j_2)\delta(h(i_4) = j_2)\sigma(i_1)\sigma(i_2)\sigma(i_3)\sigma(i_4)x_{i_1}x_{i_2}x_{i_3}x_{i_4}.$$

By linearty of expectation, $\mathbf{E}[\|Sx\|^4]$ equals

$$\sum_{j_1,j_2=1}^k \sum_{i_1,i_2,i_3,i_4=1}^n \mathbf{E}[\delta(h(i_1) = j_1)\delta(h(i_2) = j_1)\delta(h(i_3) = j_2)\delta(h(i_4) = j_2)\sigma(i_1)\sigma(i_2)\sigma(i_3)\sigma(i_4)]x_{i_1}x_{i_2}x_{i_3}x_{i_4}.$$

Since $x_{i_1}, x_{i_2}, x_{i_3}$, and x_{i_4} are constants, and h and σ are independent, we can write this as

$$\sum_{j_1,j_2=1}^k \sum_{i_1,i_2,i_3,i_4=1}^n \mathbf{E}[\delta(h(i_1) = j_1)\delta(h(i_2) = j_1)\delta(h(i_3) = j_2)\delta(h(i_4) = j_2)] \cdot \mathbf{E}[\sigma(i_1)\sigma(i_2)\sigma(i_3)\sigma(i_4)] \cdot x_{i_1}x_{i_2}x_{i_3}x_{i_4}. \tag{1}$$

If i_1, i_2, i_3, i_4 are distinct, then by 4-wise independence of σ and the fact that $\mathbf{E}[\sigma(i_1)] = 0$, we have $\mathbf{E}[\sigma(i_1)\sigma(i_2)\sigma(i_3)\sigma(i_4)] = 0$. By similar reasoning, $\mathbf{E}[\sigma(i_1)\sigma(i_2)\sigma(i_3)\sigma(i_4)] = 0$ unless either 1) $i_1 = i_2 = i_3 = i_4$, or 2) $i_1 = i_2$ and $i_3 = i_4$ but $i_1 \neq i_3$, or 3) $i_1 = i_3$ and $i_2 = i_4$ but $i_1 \neq i_2$, or 4) $i_1 = i_4$ and $i_2 = i_3$ but $i_1 \neq i_2$. In each of these cases, $\mathbf{E}[\sigma(i_1)\sigma(i_2)\sigma(i_3)\sigma(i_4)] = 1$.

Case 1: if $j_1 \neq j_2$, $\mathbf{E}[\delta(h(i_1) = j_1)\delta(h(i_2) = j_1)\delta(h(i_3) = j_2)\delta(h(i_4) = j_2)] = 0$ since the same index i cannot hash to more than one bucket. If $j_1 = j_2$, then $\mathbf{E}[\delta(h(i_1) = j_1)\delta(h(i_2) = j_1)\delta(h(i_3) = j_2)\delta(h(i_4) = j_2)] = 1/k$, so (1) simplifies to $\sum_{j=1}^{k}\sum_{i=1}^{n}(1/k)x_i^4 = \sum_{i=1}^{n} x_i^4 = \|x\|_4^4$.

Case 2: $\mathbf{E}[\delta(h(i_1) = j_1)\delta(h(i_2) = j_1)\delta(h(i_3) = j_2)\delta(h(i_4) = j_2)] = 1/k^2$, and so (1) simplifies to $\sum_{j_1,j_2}^{k}\sum_{i_1 \neq i_3 = 1}^{n}(1/k^2)x_{i_1}^2 x_{i_3}^2 = \sum_{i_1 \neq i_3 = 1}^{n} x_{i_1}^2 x_{i_3}^2 \leq \|x\|_2^4 - \|x\|_4^4$. Here $\|x\|_2^2 = \sum_{i=1}^{n} x_i^2$.

Case 3: if $j_1 \neq j_2$, $\mathbf{E}[\delta(h(i_1) = j_1)\delta(h(i_2) = j_1)\delta(h(i_3) = j_2)\delta(h(i_4) = j_2)] = 0$ since the same index i cannot hash to more than one bucket. If $j_1 = j_2$, then (1) simplifies to $\sum_{j=1}^{k}\sum_{i_1 \neq i_2}^{n}(1/k^2)x_{i_1}^2 x_{i_2}^2 \leq \frac{1}{k}\|x\|_2^4$.

Case 4: is analogous to case 3. For completeness: if $j_1 \neq j_2$, $\mathbf{E}[\delta(h(i_1) = j_1)\delta(h(i_2) = j_1)\delta(h(i_3) = j_2)\delta(h(i_4) = j_2)] = 0$ since the same index i cannot hash to more than one bucket. If $j_1 = j_2$, then (1) simplifies to $\sum_{j=1}^{k}\sum_{i_1 \neq i_2}^{n}(1/k^2)x_{i_1}^2 x_{i_2}^2 \leq \frac{1}{k}\|x\|_2^4$.

Summing over the four cases, (1) is upper bounded as $\|x\|^4 + \frac{2}{k}\|x\|^4$. Hence, $\mathbf{Var}[\|Sx\|^2] = \mathbf{E}[(\|Sx\|^2)^2] - (\mathbf{E}[\|Sx\|^2])^2 \leq \frac{2}{k}\|x\|_2^4$.

Locality Sensitive Hashing (LSH) for Jaccard Similarity In lecture we looked at LSH for Hamming distance on the Hamming cube. Here we look at the Jaccard measure: choose a random permutation π on the universe U. For a set $S \subseteq U$, the LSH for Jaccard measure is simply $h(S)$ =First element in S according to permutation π. Consider two sets S_1 and S_2. The Jaccard measure between them is $J(S_1, S_2) = |S_1 \cap S_2|/|S_1 \cup S_2|$.

1. Argue that $\mathbf{Pr}[h(S_1) = h(S_2)] = J(S_1, S_2)$.

   Solution: There are $|S_1 \cup S_2|$ items in the union. In the permutation π defined by h, the first element in S_1 and the first element in S_2 are necessarily in $S_1 \cup S_2$. If these elements are in $S_1 \cap S_2$, then they are the same element and $h(S_1) = h(S_2)$.

   Suppose we define distance as $D(S_1, S_2) = 1 - J(S_1, S_2)$.

2. Show that for any $r > 0$, if $D(S_1, S_2) < r$, then $\mathbf{Pr}[h(S_1) = h(S_2)] \geq 1 - r$.

   Solution: If $D(S_1, S_2) < r$, then $J(S_1, S_2) > 1 - r$, and we can apply the previous part.

3. Show that for any $r > 0$ and $c > 1$, if $D(S_1, S_2) \geq cr$, then $\mathbf{Pr}[h(S_1) = h(S_2) \leq 1 - cr$.

   Solution: If $D(S_1, S_2) \geq cr$, then $J(S_1, S_2) \leq 1 - cr$, and we can apply the previous part.

4. What is the expected query time and the space if you have n sets, as a function of c?

   Solution: From lecture the space is $O(n^{1+\rho}\log n)$ bits, plus the space to store the original n sets (the $\log n$ comes from storing a pointer to one of the original n sets), and the expected query time is $O(n^\rho \cdot |U|)$. Here $\rho = \frac{\log(1/(1-r))}{\log(1/(1-cr))}$.