6.5240 Sublinear Time Algorithms October 17, 2022

Lecture 10

Lecturer: Ronitt A. Rubinfeld *Scribe: Nitin Kumar*

Last lecture, we looked at an algorithm for testing whether a dense graph is triangle-free. This algorithm was fairly slow - can we do better? Can we get a runtime in poly$(1/\epsilon)$? This lecture will be dedicated to showing the answer is no:

Theorem 1. *Any one-sided tester for a dense graph being triangle-free takes* $\Omega(1/\epsilon^{\Omega(\log 1/\epsilon)})$ *time.*

Call a graph *bad* if it is ϵ-far from being triangle-free (meaning $> \epsilon n^2$ edges must be deleted). As a refresher, a one-sided tester means that we always accept triangle-free graphs, and that we reject bad graphs with probability at least 2/3.

How do we show a bound like this? First, we bound the probability of rejecting a bad graph by the probability of finding a triangle among some randomly selected vertices (i.e., the probability that the induced subgraph of the vertices contains a triangle). Then, we will construct a bad graph G, but also requires many randomly selected vertices to have a large probability of finding a triangle. Finally, to make an arbitrarily large graph with the same properties, we will cleverly "blow up" G.

1. Making the Tester Nonadaptive

In Problem 3 of Homework 2, we showed that we can make a tester for any graph property that uses $O(q)$ queries into a nonadaptive tester that uses $O(q^2)$ queries by picking $O(q)$ random vertices and querying the induced subgraph of the random vertices. The idea is that the particular vertices the tester asks for are unimportant, since graph properties are preserved under isomorphism. Notably, this reduction preserves one-sided error, since it does not change the probabilities of acceptance and rejection.

Then, if there were a one-sided tester for triangle-freeness that runs in time $t(n)$, there is a nonadaptive one-sided tester that runs in $O(t^2)$ time. Further, since a one-sided tester must be certain there is a triangle in order to reject, there must be a triangle among $O(t)$ random vertices of any bad graph with probability at least 2/3. Thus, the following result implies Theorem 1:

Theorem 2. *There exist arbitrarily large bad graphs that require sampling* $\Omega(1/\epsilon^{\Omega(\log 1/\epsilon)})$ *random vertices to find a triangle with probability* $2/3$.

Suppose a graph G had t triangles. In q random samples, the probability of finding a given triangle is $\binom{q}{3}\Theta(n^{-3}) = \Theta(q^3 n^{-3})$, so the expected number of triangles found is $\Theta(tq^3n^{-3})$. By Markov's inequality, if $tq^3n^{-3} = o(1)$, meaning $q = o(n^3/t)$, then the probability of finding a triangle is $o(1)$. Thus, a nonadaptive tester must have $q = \Omega(n^3/t)$, so to prove Theorem 2, it suffices to have $t = O(\epsilon^{\Omega(\log 1/\epsilon)} n^3)$.

2. A Bad Graph with Few Triangles

We've reduced Theorem 1 to finding arbitrarily large bad graphs with $O(\epsilon^{\Omega(\log 1/\epsilon)}n^3)$ triangles. In this section, we will find one bad graph with few triangles, which we will later be able to extend to arbitrarily large graphs in Section 3.

2.1. Construction.

Define $[m] := \{1, 2, \ldots, m\}$. Let X be a *happy* subset of $[m]$, meaning there are no distinct $a, b, c \in X$ such that $b = \dfrac{a+c}{2}$.

Then, we define a tripartite graph, with parts J, K, and L. We set $|J| = m, |K| = 2m, |L| = 3m$, and use the positive integers to number the vertices in each part. Then, for each $j \in [m]$ and $x \in X$ we have an edge between vertex j in J and vertex $j + x$ in K and an edge between vertex j in J and vertex $j + 2x$ in L, and for each $k \in [2m]$ and $x \in X$ we have an edge between vertex k in K and $k + x$ in L. The number of vertices in this graph is $6m$ and the number of edges is $4m|X|$.

Now, for each triangle, we can determine which $x \in X$ generated each of its edges. Let $a \in X$ have generated the edge from J to K, b have generated the edge from J to L, and c have generated the edge from K to L. We get that $b = \dfrac{a+c}{2}$, so since X is happy, we know $a = b = c$. This yields that each triangle is completely determined by its starting point j and the x used to generate its edges. The graph therefore contains $m|X|$ disjoint triangles, which means we need to remove $m|X|$ edges to make the graph triangle-free. To make this graph ϵ-far from being triangle-free, we should try to find a large X.

2.2. Finding a Large Happy X.

The following lemma guarantees us a large happy $X \subset [m]$.

Lemma 3. *There exists a happy $X \subset [m]$ such that* $|X| \geq \dfrac{m}{e^{10\sqrt{\log m}}}$.

Proof. Let $d \geq 2$ and $k := \lfloor \log_d m \rfloor$. Call *little* those integers whose base-d representations have at most k digits, all of which are at most $d/2$. Note that every little integer is less than $d^k \leq d^{\log_d m} = m$. Also, there are exactly $\lfloor d/2 \rfloor^k$ little integers.

Now, take any three distinct little integers a, b, c such that $b = \dfrac{a+c}{2}$. Since $b + b = a + c$ and there are no carries between digits in any of these sums, we recover that each digit of b is the average of the corresponding digits of a and c. By Jensen's inequality, if we apply any strictly convex function to the digits, the function will not have the same value on a, b, and c. This means that if we partition the little integers by the sum of the squares of their digits, a, b, and c cannot all be in the same partition, so each partition is happy! The number of partitions is at most $k\lfloor d/2 \rfloor^2$, so we can find a partition of size at least $\frac{\lfloor d/2 \rfloor^k}{k \lfloor d/2 \rfloor^2}$. By setting d appropriately, we can achieve the desired bound. (The details of setting parameters were omitted in lecture because they are tedious and unimportant.) $\square$

2.3. **Analyzing the Construction.** Let $f(m) := e^{-10\sqrt{\log m}}$, so that $|X| = f(m)m$. Since the number of edges we must remove to make the graph triangle free is $m|X| = f(m)m^2$ and $n = 6m$, our graph is bad only if $\epsilon \leq \frac{f(m)}{36}$. We also have $4m|X| = 4f(m)m^2$ edges.

This is problematic - a fixed ϵ gives us a lower bound on $f(m)$ for a bad graph, but $f(m)$ decreases to 0 as m gets large, so our construction only works up to some fixed m. Also, even if we were to make large graphs bad, the density of edges would go to 0, which means we shouldn't be using the dense graph model.

We need some way to increase the density of edges, as well as the number of edges needed to make the graph triangle-free. Luckily, we have some wiggle room: the number of triangles in our construction is $m|X| = o(n^2)$, but we know it is okay to have up to $O(\epsilon^{\Omega(\log 1/\epsilon)} n^3)$ triangles.

3. Blowing Up G

Let's take stock: for some fixed ϵ, our goal is to build a graph with at least n vertices and $O(\epsilon^{O(\log 1/\epsilon)} n^3)$ triangles. From Section 2, if we pick the largest m such that our construction is bad, then $f(m) \geq 36\epsilon$. This gives us a graph G with $6m$ vertices, at least $144\epsilon m^2$ edges, and at least $36\epsilon m^2$ triangles.

To get a larger graph, consider *blowing up* G into $G^{(S)}$ in the following way: for each vertex v in G, add S vertices $v_1, v_2, \ldots, v_S$ to $G^{(S)}$, for some $S > 1$. For each edge (u, v) in G, connect each u_i to each v_j in $G^{(S)}$, creating S^2 edges.

Since we want at least n vertices, we'll set $S = \left\lceil \frac{n}{6m} \right\rceil$. We have three things to check: the graph must be dense, bad, and have few triangles. The details are somewhat tedious:

- The number of edges in $G^{(S)}$ is at least $144\epsilon m^2 S^2 = \Theta(\epsilon n^2)$, which means that using the dense model is okay.
- Each triangle in G turned into S^3 triangles in $G^{(S)}$ (since we have S choices for each of the three vertices), so there are $36\epsilon m^2 S^3$ triangles. Each edge in $G^{(S)}$ is used in exactly S triangles (since there are S choices for the last vertex), and there are no other triangles in $G^{(S)}$. Therefore, we must delete at least $\frac{36\epsilon m^2 S^3}{S} = \epsilon(6mS)^2 \geq \epsilon n^2$ edges to make G triangle-free, so the graph is bad.

- There are $\Theta(\epsilon n^3/m)$ triangles. Since $f(m) = \Theta(\epsilon)$, we get that $m = \Omega(1/\epsilon^{\Omega(\log 1/\epsilon)})$, so the number of triangles is $O(\epsilon^{\Omega(\log 1/\epsilon)} n^3)$.

Having achieved all our goals, we have shown the lower bound from Theorem 1.