6.889 Sublinear Time Algorithms October 6, 2020

Lecture 11

Lecturer: Ronitt Rubinfeld *Scribe: Alice A. Zhang*

1 Overview

In this lecture we continue to discuss testing H-freeness in graphs. To this end, we covered the following topics:

- an additive number theory lemma
- characterization of best algorithms for property testing
- a lower bound on the complexity of a tester for triangle-freeness

2 A More General Theorem

Previously, we talked about testing dense graph properties, particularly triangle-freeness. Using the Szemerédi regularity lemma, we showed that this can be done in constant time. In fact, the proof can be extended to show that, for any small constant-sized subgraph H, we can test whether a graph is H-free in constant time in terms of the size of the graph.

The dependence on ε is much worse—in terms of ε, we get a complexity of $2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}$, where the height of the "exponent tower" is $1/\varepsilon^c$. It is possible to do better, but in this class we will see that a superpolynomial dependence on ε is required. This result is half of the following theorem for graphs in the adjacency matrix, due to Noga Alon.

Theorem 1 *If a subgraph H is bipartite, then testing a graph for H-freeness can be done with sample complexity* $\mathrm{poly}(1/\varepsilon)$. *If H is not bipartite, then no tester with complexity* $\mathrm{poly}(1/\varepsilon)$ *suffices.*

3 Testing for Triangle-Freeness

We will prove a special case of Theorem 1, where H is a triangle. Since a triangle is not bipartite, we are going to show that no polynomial in $1/\varepsilon$ suffices. There are two main tools we will use, the first of which is the additive number theory lemma.

3.1 Additive Number Theory Lemma

Let m be a positive integer and let M denote the set $\{1, 2, 3, ..., m\}$.

Lemma 2 *For all m, there exists a subset $X \subseteq M$ such that X has size greater than or equal to $m/e^{10\sqrt{\log m}}$ and contains no nontrivial solution to $x + y = 2x$, where $x, y, z \in X$.*

We will refer to the property of having no nontrivial solution to $x + y = 2x$ as *sum-freeness.* Another way of thinking about sum-freeness this is to say there are no three equally spaced points x, y, z, or that no element of the set is the average of two other elements. We give a proof by construction.

Proof of Lemma 2 Let d be an integer, set to $e^{10\sqrt{\log m}}$, and let $k = \lfloor \log_d m \rfloor - 1$. Define a set X_B as

$$X_B = \left\{ \sum_{i=0}^{k} x_i \cdot d^i \mid x_i < \frac{d}{2} \text{ for } 0 \le i \le k \text{ and } \sum_{i=0}^{k} x_i^2 = B \right\},$$

where $B > 0$, and $\sum_{i=0}^{k} x_i \cdot d^i$ is equal to some element $x \in M$, so the x_i's give a base d representation of x, i.e. $x = (x_k \ldots x_0)$.

First, we claim that X_B must be a subset of M.

Claim 3 *For any value of B, $X_B \subseteq M$.*

Why is Claim 3 true? It is clear that all elements of X_B will be positive integers. So we just need to show that the largest element in the set will not be greater than m. Since we have a condition $x_i < \frac{d}{2}$, where x_i is a digit in the base d representation of an element of X, it must be true for any element $\ell \in X_B$ that $\ell \leq d^{k+1}$. Then, by the sequence of inequalities

$$\ell \leq d^{k+1} \leq d^{\lfloor \log_d m \rfloor - 1 + 1} \leq d^{\log_d m} = m,$$

we see that every element of X_B is at most m. So $X_B \subseteq M$.

It follows that from this claim that if we consider all possible values of B, the set of X_B's partition the elements of M whose base d representations satisfy $x_i < \frac{d}{2}$ for $0 \leq i \leq k$, since it will be true for any such element that $\sum_{i=0}^{k} x_i^2$ equals a unique value of B.

We want to pick B such that X_B is maximized. Using a simple counting argument, we can show that the following claim about the possible sizes for X_B holds.

Claim 4 *There exists some value of B such that $|X_B| \geq \frac{m}{e^{10\sqrt{\log m}}}$.*

To show Claim 4, we will first figure out how big B can be. If we set each value of x_i to its maximum possible value, $d/2$, we get $B = (k+1)(d/2)^2$. So

$$B \leq (k+1)\left(\frac{d}{2}\right)^2 < k \cdot d^2.$$

Next, we find a lower bound for $\sum |X_B|$. This sum should be equal to the number of elements of M with base d representation satisfying $x_i < \frac{d}{2}$ for $0 \leq i \leq k$, since the X_B's partition this set. There are $k+1$ digits in the base d representation, and each digit can be anything from 0 up to $d/2$, non-inclusive. This gives us

$$\sum |X_B| \geq \left(\frac{d}{2}\right)^{k+1} > \left(\frac{d}{2}\right)^{k}.$$

In summary, there are at least $\left(\frac{d}{2}\right)^k$ elements partitioned into at most $k \cdot d^2$ sets. By the Pigeonhole Principle, this implies that there exists some value of B such that

$$|X_B| \geq \frac{(d/2)^k}{k \cdot d^2}.$$

Applying our settings from the beginning, $d = e^{10\sqrt{\log m}}$ and $k = \lfloor \log_d m \rfloor - 1$, gives us

$$|X_B| \geq \frac{m}{e^{10\sqrt{\log m}}},$$

completing the proof of Claim 4.

If we are able to show that some set X_B with size greater than $m/e^{10\sqrt{\log m}}$ is sum-free, we will have proven Lemma 2. This is going to work. In fact, we claim that *any* X_B is sum-free.

Claim 5 *For any value of B, there is no nontrivial solution to the equation $x+y = 2z$ where $x, y, z \in X_B$.*

This follows from our constraints in the definition of X_B. First, note that the constraint $x_i < {}^{d}\!/_{2}$ for each digit in the base d representation of an element of X_B implies that if we add two elements together, each digit in the sum will be less than d. Thus, summing a pair of elements in X_B doesn't generate carries.

Let's express the sum-freeness property using the same base d representation we used to define X_B. So, for $x, y, z \in X_B$,

$$x + y = 2z \iff \sum_{i=0}^{k} x_i \cdot d^i + \sum_{i=0}^{k} y_i \cdot d^i = 2\sum_{i=0}^{k} z_i \cdot d^i.$$

Additionally, the fact that there are no carries allows us to split this into $k+1$ equations, one for each possible value of i:

$$\begin{aligned} x_0 + y_0 &= 2x_0, \\ x_1 + y_1 &= 2z_1, \\ &\vdots \\ x_k + y_k &= 2x_k. \end{aligned}$$

It is acceptable for some of the equations in this set to have the trivial solution; however, if we want $x + y = 2z$ to have a nontrivial solution, there needs to be some i such that $x_i + y_i = 2z_i$ has a nontrivial solution.

We will also use the fact that

$$x_i + y_i = 2z_i \Rightarrow x_i^2 + y_i^2 \geq 2z_i^2 \quad \text{for all } 0 \leq i \leq k,$$

with equality only if $x_i = y_i = z_i$. This holds because Jensen's inequality tells us that for convex functions, $\frac{1}{2}(f(a_1) + f(a_2)) \geq f(\frac{a_1+a_2}{2})$, with equality only if $a_1 = a_2$. The above equation results from using the function $f(a) = a^2$, which is convex, and plugging in x_i for a_1 and y_i for a_2; then it follows that $z_i = \frac{a_1+a_2}{2}$.

So suppose there is a solution to $x + y = 2z$ and it is *not* the case that $x = y = z$. Then for some i, $x_i + y_i = 2_i$, and it is not the case that $x_i = y_i = z_i$. We get a strict inequality $x_i^2 + y_i^2 > 2z_i^2$ for this particular value of i, and for all other $0 \leq j \leq k$ with $j \neq i$, $x_j^2 + y_j^2 \geq 2z_j^2$. For each of x, y, z, the sums of the squares of the digits satisfy the strict inequality

$$\sum_{i=0}^{k} x_i^2 + \sum_{i=0}^{k} y_i^2 > \sum_{i=0}^{k} 2z_i^2.$$

Since x, y, z are each elements of X_B, they satisfy $\sum_{i=0}^{k} x_i^2 = \sum_{i=0}^{k} y_i^2 = \sum_{i=0}^{k} z_i^2 = B$. But then the above inequality evaluates to $2B > 2B$, a contradiction. Thus, our supposition was false; X_B has no nontrivial solution. This finishes the proof of Claim 5.

Together, Claims 3, 4, and 5 tell us that there is some sum-free subset X_B of M with at least ${}^{m}\!/_{e^{10\sqrt{\log m}}}$ many elements.

■

3.2 Characterization of Best Algorithms for Property Testing

Our second main tool is a characterization of "best" algorithms for property testing. We showed in Homework 2 that given:

- a graph G in the adjacency matrix model
- a nontrivial graph property P that is closed under isomorphism

- a tester T which uses $q(n, \varepsilon)$ queries

there exists some tester T', called the "natural tester," which picks $q(n, \varepsilon)$ nodes randomly and queries the submatrix to decide whether G has property P, and achieves the same behavior as T using $\Theta(q^2)$ queries. A consequence of this is that if the natural tester has complexity $\Omega(q)$, then any tester must have complexity $\Omega(\sqrt{q})$. The reduction between T and T' also preserves one-sidedness: if the graph has the property, the tester will definitely output "PASS," but on graphs without the property, there is some probability that the tester will still output "PASS."

3.3 A First Attempt at Establishing the Lower Bound

Using a sum-free subset $X \subseteq M$, we will construct graphs such that the natural tester needs lots of queries. First, we will use three sets of nodes:

$$V_1 = \{1, 2, \ldots, m\},\ V_2 = \{1, 2, \ldots, 2m\},\ \text{and } V_3 = \{1, 2, \ldots, 3m\}.$$

For each $x \in X$, we will construct three kinds of edges between the three node sets:

- an edge from every node $j \in V_1$ to node $j + x \in V_2$,
- an edge from every node $k \in V_2$ to node $k + x \in V_3$, and
- an edge from every node $\ell \in V_1$ to node $\ell + 2x \in V_3$.

There are $6m$ nodes in this construction, so $m = \Theta(n)$. The number of edges is $\Theta(m \cdot |X|) = \Theta(n^2/e^{10\sqrt{logn}})$. This is our first attempt at a construction. The graph is not dense; we will see that it is not quite dense enough to show our desired result, and we will need to try something a bit different.

This graph construction forms "intended" triangles of the form j, $j + x$, $j + 2x$, where these points lie in V_1, V_2, and V_3, respectively. There is an intended triangle corresponding to each pair of $j \in V_1$ and $x \in X$, for a total of $m \cdot |X| = \Theta(n^2/e^{10\sqrt{\log n}})$ intended triangles.

Claim 6 *All triangles in G are intended.*

Proof In our construction of G, V_1, V_2, and V_3 have no internal edges, so a triangle must have $v_1 \in V_1$, $v_2 \in V_2$, and $v_3 \in V_3$. If there were an unintended triangle, it would have an edge between $j \in V_1$ and $j + x_1 \in V_2$, an edge between $j + x_1 \in V_2$ and $j + x_1 + x_2 \in V_3$, and an edge between $j \in V_1$ and $j + x_1 + x_2 \in V_3$. In order for the third edge to exist, there needs to be some x_3 such that $j + x_1 + x_2 = j + 2x_3$. The three elements of X need to satisfy $x_1 + x_2 = 2x_3$. Since X is sum-free, there is no nontrivial solution. So there are no unintended triangles. We only have intended triangles, and we can count exactly how many there are. ■

If all the triangles in G are disjoint, then to make G triangle-free, we need to delete an edge from each triangle. So in the disjoint case, G's distance to triangle-free is just the number of triangles.

Claim 7 *All intended triangles in G are disjoint.*

Proof We identify a triangle with nodes j, $j + x$, and $j + 2x$ lying in V_1, V_2, and V_3 respectively, and suppose there is some node $k \in V_1$ forming a triangle with $j + x$ and $j + 2x$ as well, so the two triangles share the edge $(j + x, j + 2x)$. Then there is an element $y \in X$ such that $k + y = j + x$ and $k + 2y = j + 2x$. But subtracting one equation from the other gives us that $x = y$ and $j = k$. So these two triangles are not distinct after all; if two triangles share one edge, they must share all edges.

The above argument assumes that the shared edge goes between V_1 and V_2. But we can use an argument of the same form to show the same result if we assume that the shared edge lies in any two of V_1, V_2, and V_3. It follows that there is no pair of distinct intended triangles that share an edge. ■

Claims 6 and 7 together allow us to deduce that the number of edges we need to remove to make G triangle-free is

$$\Theta(\#\text{ of triangles}) = \Theta\left(\frac{n^2}{e^{10\sqrt{\log n}}}\right) = \Theta(m \cdot |X|).$$

Unfortunately, this isn't exactly what we want. We wanted to get a graph that was ε-far from triangle-free, but the graph we have constructed is $\Theta(1/e^{10\sqrt{\log n}})$-far from triangle-free.

3.4 Improving Upon First Attempt

Without going into every detail, here is an idea for "blowing up" our graph such that it is ε-far from triangle-free. We will transform G into a graph $G^{(s)}$, called the "s-blow up" of G, constructed as follows:

1. For every node in G, we create an independent set of size s in $G^{(s)}$.
2. For every edge (u, v) in G, we create a complete bipartite graph on the sets of nodes in $G^{(s)}$ corresponding to u and v.

Given this construction, we can approximately count the number of nodes, edges, and triangles in $G^{(s)}$. There are going

- $\Theta(m \cdot |s|)$ nodes,
- $\Theta(m \cdot |X| \cdot |s|^2)$ edges, and
- $\Theta(m \cdot |X| \cdot |s|^3)$ triangles.

The first two are relatively straightforward. The number of triangles is proven using Hall's Theorem; we omit this proof.

We also have a lemma relating the distance of $G^{(s)}$ from triangle-free to the number of edge-disjoint triangles.

Lemma 8 *The distance of $G^{(s)}$ from triangle-free is at least the number of edge-disjoint triangles, which is $m|X| \cdot s^2$.*

We basically saw why the first part of the lemma is true during our first attempt at a graph construction, when we reasoned that to make a graph containing some number of edge-disjoint triangles triangle-free, it is necessary to remove one edge from each triangle. We won't prove why the number of edge-disjoint triangles is $m|X| \cdot s^2$, but it also follows from Hall's Theorem.

With everything we have set up so far, we finally are able to sketch out a proof of the lower bound mentioned at the very beginning.

Proposition 9 *The lower bound on the query complexity of a tester for triangle-freeness is $\Omega(1/\varepsilon^c)$.*

Sketch of Proof If we use Lemma 8, we get that the distance of $G^{(s)}$ from triangle-free will be something like the number of disjoint triangles divided by the number of entries in the adjacency matrix, which is

$$\frac{m|X| \cdot s^2}{m^2 s^2} = \frac{|X|}{m} \geq \frac{1}{e^{10\sqrt{\log m}}} \geq \varepsilon.$$

The last inequality holds by our choice of m, which has so far gone unspecified.

We also said that the number of triangles is

$$m \cdot |X| \cdot s^3 \approx \left(\frac{\varepsilon}{c'}\right)^{c' \log c'/\varepsilon} \cdot n^3.$$

If we take a sample of size q and run the natural tester, the expected number of triangles in the sample will be

$$\mathrm{E}[\text{\# of triangles in sample}] < \binom{q}{3}\left(\frac{\varepsilon}{c'}\right)^{c' \log c'/\varepsilon}$$

The $\binom{q}{3}$ comes from choosing three nodes to check whether they form a triangle, and the $\left(\frac{\varepsilon}{c'}\right)^{c' \log c'/\varepsilon}$ is the fraction of triples that form a triangle. We set the right hand side to be much smaller than 1, unless $q > \left(\frac{c''}{\varepsilon}\right)^{c'' \log c''/\varepsilon}$. Then by Markov's Inequality, the probability that there is any triangle in the sample is

$$\Pr[\text{sample contains triangle}] << 1.$$

Since our tester has one-sided error, we must see a triangle to output "FAIL." So if we don't take a quantity of samples that is superpolynomial in $1/\varepsilon$, we are not going to see a triangle. Thus the query complexity of the tester is $\Omega(1/\varepsilon)$.

■

4 Extra Details on Parameter Settings

In the previous section, we did not get into the specifics of how to set the parameters, so we will present the parameters settings here.

Given ε, we will set m first. We choose m to be the largest integer such that

$$\varepsilon \leq e^{\frac{1}{10}\sqrt{\log m}}.$$

Then it will be true that

$$m \geq \left(\frac{c}{\varepsilon}\right)^{c \log c/\varepsilon}.$$

Now that we have m, we will set s.

$$s \approx \frac{n}{6m} \approx n\left(\frac{\varepsilon}{c}\right)^{c \log \varepsilon/c}.$$

The details of figuring out how these parameters work out to our desired result are left as an exercise.