6.842 Randomness and Computation	October 25, 2017
Lecture 14	
Lecturer: Ronitt Rubinfeld	*Scribe: Aviv Adler*

This lecture covers an application of Szemeredi's Regularity Lemma to testing whether a graph is triangle-free in constant time (though the constant involved is pretty nuts).

From Last Time

We recall some things we did in Lecture 13 which will be useful. As a notational convention, whenever we write something like $k := k(\epsilon)$, this just means "k is (only) dependent on ϵ".

Definition 1 (γ-Regularity) *If $G = (V, E)$ is a graph and $A, B \subseteq V$ (and disjoint), then the pair (A, B) is* γ-regular *if, for every $A' \subseteq A$, $B' \subseteq B$ such that $|A'| \geq \gamma|A|$ and $|B'| \geq \gamma|B|$,*

$$\left|d(A,B) - d(A',B')\right| \leq \gamma$$

where $d(X, Y)$ is the density *of edges from X to Y, i.e. $d(X,Y) = \frac{|E(X,Y)|}{|X||Y|}$.*

Theorem 2 (Lots of Triangles) *For any $\eta > 0$, there exists $\delta := \delta^{\triangle}(\eta)$ and $\gamma := \gamma^{\triangle}(\eta)$ such that for any $G = (V, E)$ with partition A, B, C of V where*

- *each pair in A, B, C has density $> \eta$*
- *A, B, C are pairwise γ-regular*

there are at least $\delta \cdot |A| \cdot |B| \cdot |C|$ distinct triangles (u, v, w) in G where $u \in A, v \in B, w \in C$.

Additionally, this holds for $\gamma = \eta/2$ and $\delta = \eta^3/16$

Theorem 3 (Szemeredi Regularity Lemma) *For all m and $\epsilon > 0$, there exists $T := T(m, \epsilon)$ such that, given $G = (V, E)$ where $|V| > T$ (and an equipartition $\mathcal{A}$ of V into m sets) then there exists equipartition $\mathcal{B}$ of V into k subsets $V_1, \ldots, V_k$ such that:*

- *$m \leq k \leq T$;*
- *at most $\epsilon\binom{k}{2}$ of the pairs V_i, V_j of subsets are* not *ϵ-regular;*
- *(and $\mathcal{B}$ refines $\mathcal{A}$).*

Note that k is upper-bounded by T, which has no dependence on $n := |V|$. This means for a sufficiently large graph, we can represent it as a collection of k subsets between which the edges "look" random (except for an ϵ-fraction of the pairs).

1 The $\triangle$-Free Problem

We now want to find an algorithm which can take a graph $G = (V, E)$ – represented as an adjacency matrix (so input size n^2) – and determine whether it is $\triangle$-free. Of course, to solve this problem exactly means reading more-or-less the entire input and so will take $\Omega(n^2)$ time (in fact, we can prove that at least $2\binom{n/2}{2} = \Theta(n^2)$ entries must be checked, because we can partition V into two equal sets V_1, V_2 and have an edge (u, v) for all $u \in V_1, v \in V_2$ – and then every edge within a part would result in a triangle being completed).

Instead, we want to test $\triangle$-freeness with some margin of error: if G is really $\triangle$-free we should know; if G is *far* (in some sense – see Definition 4) from being $\triangle$-free, we should know that it's not $\triangle$-free; and if it is neither then either output is acceptable (if we find it's not $\triangle$-free, this is okay because it isn't, but if we find it is $\triangle$-free that is also okay because it almost is).

Definition 4 (ϵ-Far) *Let G be a graph given in adjacency-matrix format (so input size is n^2). Then G is* ϵ-far from $\triangle$-free *if there is no way to add or delete $\leq \epsilon n^2$ (an ϵ fraction of the input size) to get a $\triangle$-free graph G'.*

Since $\triangle$-free is a *monotonic* property (adding new edges cannot make G triangle-free), we are only concerned with deleting edges.

We also note that if we want to deterministically solve the approximation described above, we would still need $\Omega(n^2)$ queries; we instead want a *randomized* algorithm. Specifically:

Problem 1 ($\triangle$-Free, Approximate) *Given graph G in adjacency-matrix form (which can be queried), we want an algorithm A which does the following:*

- *if G is $\triangle$-free, $A(G)$ (always) returns* `pass`*;*
- *if G is ϵ-far from $\triangle$-free, $A(G)$ returns* `fail` *with constant probability (specifically, not dependent on n);*
- *if G is neither of the above, $A(G)$ is allowed to return either* `pass` *or* `fail`*.*

The most obvious algorithm is to start querying random triplets of vertices; if we find a $\triangle$ we output `fail`, and if we go for a while without finding a $\triangle$, we output `pass`.

2 The $\triangle$-Free Tester

Our testing algorithm does the following where $\delta := \delta(\epsilon)$ (the exact dependence to be described later):

Given graph $G = (V, E)$ (as adjacency matrix) and $\epsilon > 0$, **do δ^{-1} times:**

- pick $u, v, w \in V$ uniformly at random;
- if $\triangle_{u,v,w}$ (i.e. u, v, w are vertices of a triangle in G), return `fail`

If above loop ends without `fail`, then return `pass`

If G is $\triangle$-free, then of course it will never find a $\triangle$ and so it is guaranteed to return `pass`. Therefore we are only concerned with what happens if G is ϵ-far from $\triangle$-free. So we want to show:

Theorem 5 (ϵ-Far = Lots of $\triangle$s) *For any $\epsilon > 0$, there exists $\delta := \delta(\epsilon) > 0$ such that if G is ϵ-far from $\triangle$-free, then G has at least $\delta\binom{n}{3}$ distinct $\triangle$s.*

It's easy to see that this implies that the algorithm has a constant probability of returning `fail` if G is ϵ-far from $\triangle$-free. This is because each sampling of vertices has at least a δ chance of finding a $\triangle$, and we apply it δ^{-1} times. Therefore, the probability that we *do not* find a $\triangle$ is at most $(1-\delta)^{\delta^{-1}} \geq 1/4$ (really, for small δ this is basically e^{-1}). Therefore, we have the corollary:

Corollary 6 (Algorithm Works!) *The testing algorithm given solves Problem 1.*

We now prove Theorem 5.

Proof We apply Szemeredi's Regularity Lemma with the following parameters:

- $m = 5/\epsilon$, and
- $\epsilon' = \min(\epsilon/5, \gamma^{\triangle}(\epsilon/5)) = \epsilon/10$ (where $\gamma^{\triangle}$ is as defined in Theorem 2).

This gives us an equipartition of V into k subsets $V_1, \ldots, V_k$ (so each part has n/k vertices) such that:

- $5/\epsilon \leq k \leq T(5/\epsilon, \epsilon')$, which implies that
$$\frac{\epsilon n}{5} \geq \frac{n}{k} \geq \frac{n}{T(\epsilon/5, \epsilon')} \tag{1}$$
- at most $\epsilon' \binom{k}{2}$ pairs of subsets are not ϵ'-regular.

Since our goal is to wind up in a position to apply Theorem 2, what we need are three parts V_a, V_b, V_c such that (i) they are pairwise regular and (ii) the edges between any pair are dense (i.e. the η is sufficiently large). To show the existence of such V_a, V_b, V_c, we "clean up" G by removing edges (which obviously can only reduce the number of triangles) to obtain G' where edges only exist between pairs of parts which are dense and regular. In particular, we make the following removals:

1. *remove all edges within any part V_i:*

   This removes at most $\binom{n/k}{2} \leq \frac{1}{2}\left(\frac{n}{k}\right)^2$ edges from any given part V_i, and there are k parts, so at most $\frac{n^2}{2k} \leq \frac{\epsilon n^2}{10}$ edges in total are removed.

2. *remove all edges between any pair of parts V_i, V_j which are not ϵ'-regular:*

   At most $\epsilon' \binom{k}{2} \leq \frac{\epsilon' k^2}{2}$ pairs of parts are not ϵ'-regular; and between each pair there are at most $\left(\frac{n}{k}\right)^2$ edges. So since $\epsilon' \leq \epsilon/5$ (we can plug in the $\epsilon/10$ here but no need), at most $\frac{\epsilon' n^2}{2} \leq \frac{\epsilon n^2}{10}$ edges in total are removed.

3. *remove all edges between any pair of parts V_i, V_j whose density $d(V_i, V_j) \leq \epsilon/5$:*

   Between any pair of parts there are $\left(\frac{n}{k}\right)^2$ possible edges; having density at most $\epsilon/5$ means that at most $\frac{\epsilon}{5}\left(\frac{n}{k}\right)^2$ of them are actually edges. And there are at most $\binom{k}{2} \leq \frac{k^2}{2}$ pairs of parts, so the number of edges removed is at most $\frac{\epsilon}{5}\left(\frac{n}{k}\right)^2 \frac{k^2}{2} \leq \frac{\epsilon n^2}{10}$.

Therefore, the total number of edges removed by this "cleanup" is at most $\frac{3\epsilon}{10} n^2 < \epsilon n^2$. But since G is ϵ-far from $\triangle$-free, this means that the new graph G' still has a triangle somewhere.

Because we removed all edges within parts, the vertices of this triangle must be in three distinct parts V_a, V_b, V_c; because we removed all edges between non-regular pairs, V_a, V_b, V_c must be pairwise ϵ'-regular; and, finally, since we removed all edges between pairs of parts which had density $\leq \epsilon/5$, each pair in V_a, V_b, V_c must have density more than $\epsilon/5$.

Therefore, we can apply Theorem 2 with G'' being graph induced by $V_a \cup V_b \cup V_c$ on G' (i.e. we just retain those vertices and all edges within them) and with the parameter $\eta = \epsilon/5$, so that $\epsilon' = \gamma^{\triangle}(\eta)$. Note that η depends only on ϵ, so δ' depends only on ϵ. The theorem then gives that the number of triangles in G'' is at least

$$\delta^{\triangle}(\epsilon/5) \cdot |V_a| \cdot |V_b| \cdot |V_c| = \delta^{\triangle}(\epsilon/5)(n/k)^3 \geq \left(\frac{\delta^{\triangle}(\epsilon/5)}{T(\epsilon/5, \epsilon')^3}\right) n^3$$

Since G'' is made by deleting vertices from G', which is the original graph G minus some edges, we know that G has at least this many triangles as well. So, setting $\delta = \frac{\delta^{\triangle}(\epsilon/5)}{T(\epsilon/5, \epsilon')^3}$, which depends only on ϵ, we have completed our proof. ■

Addenda

A few minor additional things.

Small G (or "g")

The proof of Theorem 5 does depend on G having at least $T(\epsilon/5, \epsilon')$ vertices, which is am unrealistically gigantic number. What happens if it is smaller?

But if it is smaller, then its size is upper-bounded by a constant so we can just run an algorithm which checks all triplets of vertices for triangles. This is technically also constant-time, even if the constant is ridiculous. So we have a constant-time algorithm for both "small" and "large" G.

Can we make the run-time of $\triangle$-free testing sane?

While the result shows the $\triangle$-free testing algorithm runs in constant time regarding the size of the graph, the constant is extremely large and not really practical. It turns out that $T(\epsilon/5, \epsilon')$ (where $\epsilon' = \epsilon/10$ as we saw) is basically a $(1/\epsilon)$-height tower of twos, e.g. $2^{2^{2^{\cdots}}}$; and then our algorithm basically tells us to take this many samples (cubed).

It is known that no constant-time (with regards to the size n of the graph G) algorithm for Problem 1 exists which is only polynomially dependent on $1/\epsilon$. But it's not known whether something as awful as a tower of twos is really needed; perhaps there's some algorithm which only requires $O(2^{1/\epsilon})$ samples or something. If so, the algorithm (or at the very least the proof that the works algorithm) would probably have to be not dependent on Szemeredi partitions, as the number of Szemeredi partitions needed to make this argument work is known to be basically this tower-of-2's.

Either finding a better constant-time algorithm or proving stronger lower bounds would be a good open problem.