CS 0368-4246: Combinatorial Methods in Algorithms (Spring 2025) April 21, 2025

Lecture 5: Property Testing and Sublinear-Time Algorithms

Instructor: *Or Zamir* Scribes: *Tomer Shinar*

1 Introduction

In previous lectures, we saw examples of using samples of subgraphs to solve a problem on a bigger graph. Today, we will study cases where we can query only a small part of the input to answer a question about it.

2 Problems Layout

We are given:
• A very big object (array/ graph)
• A property P that we want to determine whether the object satisfies.
We will ask when we can decide whether the property p holds for the object, without looking at all of it (using sampling from it). [1]

We have already seen the following examples of sublinear-time algorithms:
• Find P_k in a graph in $O(n)$ time instead of $O(m)$.
• Sampling Edges to see if a graph is dense.

3 Array Majority

Object: binary array
Property: 1-Maj: most of the elements in the array are 1's.
We consider a relaxed version of the 1-Majority problem:
• If the majority is 1 - we will want to return true
• If the array is $\varepsilon - far$ from being 1-maj - we will want to return false
• Otherwise - any result will be accepted.

Definition 1. *We will say an object $\varepsilon - far$ from a property if we need to change at least ε of its cells to hold the property. In our case, the array is $\varepsilon - far$ from 1-Maj if it has fewer than $(1/2 - \varepsilon)n$.*

We will want to solve this problem with high probability (say $> 90\%$).

Algorithm 2. *We will sample $\frac{100}{\varepsilon^2}$ independent indices and calculate their average. If the average is more than $\frac{1-\varepsilon}{2}$ we will say true, otherwise false.*

Claim 3. *The probability that the sampled average is far from the real average is small. (meaning that the algorithm returns the correct result with high probability)*

Proof. We will look at the average of the samples:
$X = \frac{1}{k}(X_1 + ... + X_k)$
$X_i \sim Ber(\alpha)$
$\mathbb{E}[X_i] = \alpha, V(X_i) \leq 1$
$\mathbb{E}[X] = \alpha, V(X) = \frac{1}{k^2} k V(X_i) \leq \frac{1}{k} 1$
$\sigma(X) = \sqrt{V(X)} \leq \sqrt{\frac{1}{k}} = \frac{\varepsilon}{10}$
From Chebyshev, the probability that X is far by more than $5\sigma(X)$ is small. □

Claim 4. *Every deterministic algorithm for the same problem requires $\Omega(n)$samples.*

Proof. We will apply the algorithm for the input $\vec{x} = 1^n$. Assume for contradiction the algorithm queries less than $(\frac{1}{2} - \varepsilon)n$. We will mark with I the indices sampled, and define a new input:

$$x_i = \begin{cases} 1, & \text{if } i \in I \\ 0, & \text{if } i \notin I \end{cases}$$

x is $\varepsilon - far$ from the property, but the algorithm does not distinguish between both inputs. □

Claim 5. *Every algorithm (can be randomized) that works for $\varepsilon = 0$ requires $\Omega(n)$ samples.*

Proof. For simplicity, we will assume that n is even, majority 1 means $> \frac{n}{2}$ ones. We will define the following distribution of inputs:
• Sample uniformly a random vector of length n with $\frac{n}{2}$ ones.
• Sample uniformly an index i
• Flip the value in the vector at index i
Proof idea: an algorithm that does $o(n)$ will encounter i with probability of $o(1)$, so when it gives its answer, we "haven't decided yet" the correct answer. □

4 Graph Property Testing [2]

There are 2 possible models for sampling on a graph:

1. Adjacency Matrix - each query access a cell.
   $\varepsilon - far$ - changing εm^2 cells.

2. - Adjacency List - each vertex has its degree and a pointer to the neighbors array. To query if an edge exists, there are some hybrid models (have both representations, a sorted neighbors list).

We will focus on the first model.
What properties will be interesting and relevant to test?

1. Connectivity

2. Contains subgraph H of constant size such that H is not bipartite (otherwise, every non-sparse graph will contain it). For example, triangles.
   True: no triangles, False: need to remove $> \varepsilon n^2$ to remove all triangles.

3. Is graph α-colorable/bipartite?
   True: bipartite, False: need to remove $> \varepsilon n^2$ edges to make it bipartite.

5 Tester for Bipartite

Algorithm 6. *We will sample a small group of vertices v', check if the induced subgraph $G[v']$ is bipartite, and answer the same for the original graph.*

If G is bipartite then $G[v']$ is also bipartite. We will want to show that if G is $\varepsilon - far$ from bipartite, then the induced subgraph, with good probability, will also not be bipartite.

Claim 7. *v' of size $\frac{10log(\frac{1}{\varepsilon})}{\varepsilon^2}$ is enough. The number of samples will be $\tilde{O}(\varepsilon^4)$.*

We will try to sample a small subgraph that will reduce the number of partitions in the graph that consist with the subgraph, and then sample more edges to cancel the remaining partitions.

Proof. We will think of the sample as 2 independent parts U, S with sizes $|U| = \frac{100\log\frac{1}{\varepsilon}}{\varepsilon}$, $|S| = \frac{10|U|}{\varepsilon}$.

Definition 8. *We will define U as "good" if there are at most $\frac{\varepsilon}{6}n$ vertices such that they are not neighbor of U, and have a degree of $\geq \frac{\varepsilon}{6}n$.*

Claim 9. *A randomly chosen U is "good" with probability ≥ 0.9.*

Proof. Let v be a vertex of degree $\geq \frac{\varepsilon}{6}n$. Every random vertex is a neighbor of v with probability $\geq \frac{\varepsilon}{6}$. So if we randomly choose $\frac{100\log\frac{1}{\varepsilon}}{\varepsilon}$ vertices, the probability that all of them not being neighbor of v is $(1-\frac{\varepsilon}{6})^{\frac{100\log\frac{1}{\varepsilon}}{\varepsilon}} \leq e^{\frac{\varepsilon}{6}\frac{100\log\frac{1}{\varepsilon}}{\varepsilon}} \leq \varepsilon^2$.
The expected amount of such vertices is $\leq \varepsilon^2 n$, hence the number of such vertices will be lower than $\frac{\varepsilon}{6}n$ with good (≥ 0.9) probability. □

Let us look at a possible partition of U: $U = U_1 \oplus U_2$. Such partition means that $N(U_1)$, $N(U_2)$ are on different sides. To contradict it, it will be enough to find an edge with both vertices on the same $N(U_i)$. We will say that such edge violates the partition (U_1, U_2).

Claim 10. *If U is "good" and $U = U_1 \oplus U_2$ than there are at least $\frac{\varepsilon n^2}{10}$ edges that violates (U_1, U_2).*

Proof. We will look at a possible partition of the V, $N(U_1) \oplus V \setminus N(U_1)$. Since the graph is $\varepsilon - far$ from bipartite, there are at least εn^2 edges inside $N(U_1)$ or inside $V \setminus N(U_1)$. Let's count the edges in the following groups, knowing that U is "good":
- Vertices of degree $\geq \frac{\varepsilon}{6}n$: $\geq \frac{\varepsilon}{6}n^2$
- $\geq \frac{\varepsilon}{100}n$ problematic vertices (not neighbors of U): $\geq \frac{\varepsilon}{100}n^2$
- vertices in U: constant

This sums to $\geq \frac{\varepsilon}{2}n^2$ edges, so there are at least $\frac{\varepsilon}{2}n^2$ violating edges for (U_1, U_2). □

Every partition (U_1, U_2) of a "good" U can be contradicted by a random edge with probability of $\geq \frac{\varepsilon}{10}$. If we sample $|S| = \frac{10|U|}{\varepsilon}$ edges, then a specific (U_1, U_2) will be contradicted with a probability $\geq 1 - e^{-|U|}$, because the probability that in all samples we missed an edge that violate U $\leq (1-\frac{\varepsilon}{10})^{|S|} = (1-\frac{\varepsilon}{10})^{\frac{10|U|}{\varepsilon}} \leq e^{-|U|}$.
The probability there is a partition of U we didn't contradicted $\leq 2^{|U|}e^{-|U|} = o(1)$.
Notice that we didn't sample all edges in S, but only —S— edges in S, and edges between S and U. So, the total amount of samples is $|U|^2 + |S| + |S||U|$ which is about $\frac{\log\frac{1}{\varepsilon}}{\varepsilon^3}$. □

We managed to solve this problem with $\tilde{O}(\frac{1}{\varepsilon^3})$. It was solved with $\tilde{O}(\frac{1}{\varepsilon^2}$ and it was proved that we need $\Omega(\varepsilon^{-1.5})$ samples to solve the problem, but the range between $\varepsilon^{-1.5}$ to ε^{-2} is still open.

6 Tester for "G don't have a subgraph H"

More specifically we will want to test if G have no triangles.

Algorithm 11. *We sample a large constant amount of triplets of vertices, for each we will check if they form a triangle.*

If G have not triangles we will always say "yes".
If G is $\varepsilon - far$ from being triangle free - ?
Side note: the number of samples must be very large, otherwise we could have found a good algorithm for k-clique when $\varepsilon = \frac{1}{n^2}$.

Definition 12. *Given a graph G and $A, B \subseteq V$ we will define the density*
$d(A,B) = \frac{|E(A,B|}{|A||B|}$

Definition 13. *Given a graph G, $A, B \subseteq V$ and $\gamma > 0$, we will call (A, B) $\gamma - regular$ if for all $A' \subseteq A : |A'| \geq \gamma|A|$, $B' \subseteq B : |'B| \geq \gamma|B|$ it holds that $|d(A',B') - d(A,B)| \leq \gamma$*

Theorem 14. *Szemerédi regularity lemma*
For every $\gamma > 0$ and every positive integer l, there exists an integer $T = T(\gamma, l)$ such that every graph $G = (V, E)$ with $|V| \geq T$ has a partition of its vertex set

$$V = V_0 \cup V_1 \cup \cdots \cup V_t$$

for some integer t satisfying $l \leq t \leq T$, and the following conditions hold:

- $\forall i : \lfloor \frac{|V|}{t} \rfloor \leq |V_i| \leq \lceil \frac{|V|}{t} \rceil$
- *all but at most $\gamma\binom{t}{2}$ of the pairs (V_i, V_j), $i \neq j$, are γ-regular.*

Note: T is a very large function of γ, know to be at least $Tower(Poly(\frac{1}{\gamma}$.

Lemma 15. *Triangle Removal Lemma*
For all $\varepsilon > 0$ exists $\delta > 0$ such that if G is $\varepsilon - far$ from being triangle-free, then G have at least δn^3 different triangles.

Proof. ($\delta \approx \frac{10000\varepsilon^3}{T(\frac{\varepsilon}{4}, \lceil \frac{4}{\varepsilon} \rceil)^3}$)
We can assume n is big, otherwise $n = O(1)$ and we can sample everything.
We take a partition from regularity lemma with $\gamma = \frac{\varepsilon}{4}$, $l = \lceil \frac{4}{\varepsilon} \rceil$. We will ignore every edge that matches on of the following conditions:

- edges inside a part V_i: $\leq t(\frac{n}{t})^2 = \frac{n^2}{t} \leq \frac{\varepsilon}{4}n^2$
- edges between a non-regular pair: $\leq \gamma n^2 \leq \frac{\varepsilon}{4}n^2$
- edges in a pair $d(V_1, V_2) < \frac{\varepsilon}{2}$: $\frac{t^2}{2}\frac{\varepsilon}{2}(\frac{n}{t})^2 < \frac{\varepsilon}{4}n^2$

Overall, we deleted less than εn^2 edges.After all the deletions, there are still triangles. So the graph contains parts V_1, V_2, V_3 such that every pair is regular with $d(V_i, V_j) < \frac{\varepsilon}{2}$. We will show that each such triplet have many triangles.

Definition 16. *We call $v \in V_1$ regular, if it has at least $\frac{\varepsilon}{8}\frac{n}{t}$ neighbors in V_2 and at least $\frac{\varepsilon}{8}\frac{n}{t}$ neighbors in V_3.*

Claim 17. *Most of the vertices in V_1 are regular.*

Let's take $V_1' \subset V_1$, the subset with all vertices that have less than $\frac{\varepsilon}{8}\frac{n}{t}$ neighbors in V_2. Notice that $d(V_1', V_2) \leq \frac{|V_1'|\frac{\varepsilon}{8}\frac{n}{t}}{|V_1'||V_2|} = \frac{\varepsilon}{8}$
So from regularity $|V_1'| < \gamma|V_1| = \frac{\varepsilon}{4}\frac{n}{t}$. The same is correct for V_3, so except for $\frac{\varepsilon}{2}\frac{n}{t}$, all other vertices of V_1 ($(1-\frac{\varepsilon}{2})\frac{n}{t}$ vertices) are regular.

If $v \in V_1$ is regular, we will look at V_2', V_3' his neighbors in V_2, V_3. Each edge between its neighbors will create a triangle, so it is in at least $|E(V_2', V_3')|$ triangles.
$|E(V_2', V_3')| \geq (d(V_2, V_3) - \gamma)|V_2'||V_3'| = \Omega(\varepsilon\varepsilon\frac{n}{t}\varepsilon\frac{n}{t}) = \Omega(\varepsilon^3\frac{n^2}{t^2})$
Sum for all regular v we will get $\Omega(\varepsilon^3\frac{n^3}{t^3})$ □

References

[1] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 73–83, 1990.

[2] Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.