6.5240 Sublinear Time Algorithms September 9, 2024

Lecture 2

Lecturer: Ronitt Rubinfeld *Scribe: Dylan Dalida*

Today was the second lecture of the course. We estimated the average degree of a graph in the general case, in sublinear time

1 Problem statement

Consider a simple graph (V, E) with nodes $V = \{1, 2, \cdots, \}$ and undirected edges $|E| = m$. Let $\deg(v)$ be the *degree* of node v; i.e. the number of edges connected to node v. The *average degree* is defined as

$$\frac{\sum_{v \in V} \deg(v)}{n} = \frac{2m}{n}$$

Assumptions:

- We can access information about the graph through the following queries:
  - Degree: given v vertex, return $\deg(v)$.
  - Neighbor: given (v, j) - a vertex and an index, return the jth edge connected to v.
    Note: the neighbors are not listed in any particular order.

  Both can be queried in constant time.
- The number of edges is $\Omega(n)$.
  Reason: Sparse graphs such as the 0-edge graph vs the 1-edge graph are hard to distinguish, might need $\Theta(n)$ queries on average to get a nonzero degree if it even exists.

We want to find a good multiplicative approximation to the average degree of G in sublinear time. Formally, we're given two parameters: ϵ, an approximation parameter, and δ, a success probability parameter. If $\bar{d} \geq 1$ is the actual average degree, our goal is to provide a $\tilde{d}$ such that

$$\Pr[|\tilde{d} - \bar{d}| \leq \epsilon \bar{d}] \geq 1 - \delta.$$

Taking some k samples (with the number possibly dependent on δ and/or ϵ) might work in this case, but there are counterexamples for why this might fail even given the above assumptions. In particular, there exist graphs for which we know that the number of samples must be $\Omega(\sqrt{n})$ - see last week's notes for more details.

In the previous lecture, we made the additional assumption that the ratio between the maximum and the minimum degree is bounded (i.e. for some Δ, $\deg(v) \in [\Delta, 10\Delta]$), and this allowed us to get the average by sampling. Here we won't make that assumption, but surprisingly, there exists an algorithm that handles this case as well, and achieves the bound of $\Theta(\sqrt{n})$.

2 Algorithm description

Definition. Let $\prec$ be an operation that compares any two vertices u and v that satisfies $u \prec v$ iff either $\deg(u) < \deg(v)$, or both $\deg(u) = \deg(v)$ and $u < v$.

In particular, $\prec$ is both symmetric and transitive; i.e. for any two nodes u and v, either $u \prec v$ or $v \prec u$, and if for any three nodes u, v, and w, we have $u \prec v$ and $v \prec w$, then we have $u \prec w$. In other words, $\prec$ is a *total order* on the nodes, allowing us to order them in increasing order according to $\prec$.

Definition. Let $\deg^+(v)$ be the number of neighbors u of v such that $v \prec u$, i.e. the number of neighbors of v that go after it in the total order.

In particular, we have $\frac{\sum_{u \in V} \deg^+(u)}{n} = \frac{m}{n} = \frac{\bar{d}}{2}$. This is true because each edge (u_i, v_i) is counted at most once in the sum, because either $u_i \prec v_i$ or vice-versa.

This gives rise to our algorithm:

Algorithm. Let k be a positive integer. From $i = 1$ to k:

1. Pick a random vertex v_i.
2. Pick a random neighbor of v_i (which we can do in O(1) because query reasons). Call it u_i.
3. If $v_i \prec u_i$, let $X_i = 2\deg(v_i)$. Otherwise, let $X_i = 0$.

Finally, return $\frac{\sum_{i=1}^{k} X_i}{k}$.

3 Proof of runtime and correctness

First, let's make sure that this is sublinear: comparison, degree queries, and selecting random numbers are all doable in constant time. And since these each of these are done at most twice per iteration, each cycle of steps 1-3 runs in constant time, so in total this is $O(k)$. Our goal is to show that we can choose $k \in O(\sqrt{n})$ that solves our problem. Individually, the X_is are a good estimator for $\bar{d}$:

Claim. $E[X_i] = \bar{d}$.

Proof.

$$\begin{aligned} E[X_i] &= \sum_{v \in V} Pr[v \text{ chosen in step 1}] \cdot E[X_i | v \text{ chosen in step 1}] \\ &= \frac{1}{n} \sum_{v \in V} \sum_{u \in V} Pr[v \text{ chosen in step 1}] \cdot E[X_i | (v,u) \text{ chosen in step 1}] \\ &= \frac{2}{n} \sum_{v \in V} \deg^+(v) \\ &= \frac{2m}{n} \qquad \square \end{aligned}$$

This gives us an estimate, but we're not sure if it's good enough. If we could somehow find a bound on the probability that $v_i < u_i$, then we could make sure that with high enough probability, the average of the X_is don't go too far from $\bar{d}$. In a way, this can be done:

Lemma. For all $v \in V$, $\deg^+(v) \leq \sqrt{2m}$ (!).

Proof. Suppose there were t nodes that are greater than v using $\prec$ as comparison, i.e. $t = |\{u \mid v \prec u\}|$. (Note that this is different from $\deg^+(v)$ in the sense that we consider *all* nodes, not just neighbors.)

- If $t < \sqrt{2m}$, then because all nodes in $\deg^+(v)$ are also counted in t, we have $\deg^+(v) \leq t < \sqrt{2m}$.
- If $t \geq \sqrt{2m}$, then if $\deg^+(v) \geq \sqrt{2m}$, then $\deg(v) \geq \sqrt{2m}$ as well. But because the total order is defined as increasing in order of degrees, each of the t vertices that come after v in the total order also have degree at least $\sqrt{2m}$. Thus the total sum of degrees is at least $\sqrt{2m}(\sqrt{2m}+1) > 2m$, which is a contradiction (as the sum of all degrees must be exactly $2m$). $\square$

Recall the previous lecture, where we obtained a good multiplicative bound for each of the X_i, and combined them all using Chernoff. Because we don't have this kind of bound for the general case, and we can't really use the previous fact to directly bound the values of X_i, we're going to bound the variance. This is motivated by the following inequality, which will allow us to bound the probability of failure; i.e. that the additive error is outside the bounds we want:

Proposition. *(Chebyshev)* Let X be a random variable. Then $Pr[|X - E[X]| \geq b] \leq \frac{\text{Var}[x]}{b^2}$.

Let's just do it: it's variance bounding time.

Claim. $\text{Var}[X_i] \leq 4\bar{d}\sqrt{2m}$.
Proof. Following the same proof above, we know that

$$\begin{aligned}
\text{Var}[X_i] &= E[X_i^2] - E[X_i]^2 \\
&\leq E[X_i^2] \\
&= \sum_{v \in V} \sum_{u \in V} \Pr[v \text{ chosen in step 1}] \cdot \Pr[u \text{ chosen in step 2}] \cdot (X_i \text{ given } (u, v))^2 \\
&= \sum_{v \in V} \sum_{u \in V} \left(\frac{1}{n}\right) \left(\frac{1}{\deg(v)}\right) (0 \text{ if } v \prec u \text{ else } 2\deg(v))^2 \\
&= \frac{1}{n} \sum_{v \in V} \sum_{\substack{u \in V \\ v \prec u}} \left(\frac{1}{\deg(v)}\right) (2\deg(v))^2 \\
&= \frac{4}{n} \sum_{v \in V} \sum_{\substack{u \in V \\ v \prec u}} \deg(v) \\
&= \frac{4}{n} \sum_{v \in V} \deg^+(v) \deg(v) \\
&\leq \frac{4}{n} \sum_{v \in V} \sqrt{2m} \deg(v) = 4\bar{d}\sqrt{2m}. \qquad \square
\end{aligned}$$

where in the last line we used the claim on the bound of $\deg^+(v) \leq \sqrt{2m}$. Now this might seem like too big of an upper bound to lead to a useful result, but note that we're averaging multiple X_i together. We can then use the following fact to find the variance of the averages (which indeed is smaller):

Proposition: Let $Y = \frac{1}{k} \sum_{i=1}^{k} X_i$ be an average of pairwise independent variables X_i. Then Var[Y] $= \frac{1}{k} Var[X_i]$.

Now we're ready for the home stretch. Choose $k = \frac{16}{\epsilon^2}\sqrt{n}$.

Claim: $Pr[|\tilde{d} - \bar{d}| \leq \epsilon \bar{d}] \geq \frac{3}{4}$.
Proof: We know that

$$E[\tilde{d}] = \bar{d}$$

and by the previous proposition we also know that

$$\text{Var}[\tilde{d}] = \frac{1}{k}\text{Var}[X_i] = \frac{4\bar{d}\sqrt{2m}}{k}$$

So by Chebyshev we have

$$\begin{aligned}
\Pr[|\tilde{d}-\bar{d}| \geq \epsilon\bar{d}] &= \Pr[|\tilde{d}-E[\tilde{d}]| \geq \epsilon\bar{d}] \\
&\leq \frac{\frac{4\bar{d}\sqrt{2m}}{k}}{(\epsilon\bar{d})^2} \\
&= \frac{4\sqrt{2m}}{k\epsilon^2\bar{d}} \\
&= \frac{2\sqrt{2}n}{k\epsilon^2\sqrt{m}} \qquad \left(\bar{d} = \frac{2m}{n}\right) \\
&\leq \frac{4\sqrt{n}}{k\epsilon^2} \qquad \left(\text{multiply } \bar{d} \geq 1 \implies \frac{\sqrt{2m}}{\sqrt{n}} \geq 1\right) \\
&= \frac{1}{4}. \qquad \square
\end{aligned}$$

Finally, by "amplifying" (running multiple runs of that algorithm and taking the average, cf Homework 0), we could guarantee that the probability is as close as possible to 1 as we want. In particular, running the algorithm $O(\log \frac{1}{\delta})$ times can give us a success probability of at least $1-\delta$. Yay! `:DD`