6.896 Sublinear Time Algorithms September 21, 2001

Lecture 3: Estimating the Number of Connected Components

Lecturer: Ronitt Rubinfeld *Scribe: Kevin Matulef*

1 Testing for Connectedness

Reviewing from last class, we had the following tester for connectedness:

1. Choose $m = O(\frac{1}{\epsilon d})$ nodes.
2. For every chosen node s, perform a BFS until
   (a) $\geq \frac{8}{\epsilon d}$ new nodes are found.
   (b) s is found to be in a component of size $\leq \frac{8}{\epsilon d}$.
3. If (b) occurs, reject, otherwise accept.

Note that this algorithm requires $O(\frac{1}{\epsilon^2 d})$ total queries. For a BFS to discover up to $\frac{8}{\epsilon d}$ new nodes, it requires up to $O(\frac{8}{\epsilon d} \cdot d) = O(\frac{1}{\epsilon})$ queries. Since we do $O(\frac{1}{\epsilon d})$ BFSs, the total number of queries is $O(\frac{1}{\epsilon^2 d})$.

To analyze the validity of the algorithm, we had the following lemma:

Lemma 1 *G is ϵ-far from $P_n^{(d)} \Rightarrow G$ has $\geq \frac{\epsilon dn}{4}$ components*

From this, we derive the following corollary:

Corollary 2 *G is ϵ-far from $P_n^{(d)} \Rightarrow G$ has $\geq \frac{\epsilon dn}{8}$ "small" components of size $\leq \frac{8}{\epsilon d}$.*

Proof
Let $\ell =$ number of components of size $\leq \frac{8}{\epsilon d}$ and $\ell' =$ number of components of size $> \frac{8}{\epsilon d}$. From the lemma, we know that $\ell + \ell' \geq \frac{\epsilon dn}{4}$.

We can bound the number of large components ℓ' with the equation $\ell' \frac{8}{\epsilon d} \leq n$. This simply comes from the fact that there can't be more nodes in large components than the total number of nodes. We can rearrange this to say $\ell' \leq \frac{\epsilon dn}{8}$.

Combining our lower bound on the total number of components with our upper bound on the number of large components, we get the lower bound $\ell \geq \frac{\epsilon dn}{4} - \frac{\epsilon dn}{8} = \frac{\epsilon dn}{8}$ on the number of small components. ■

Now we have the tools to prove that the tester works.

Theorem 3 *Let G be a degree $\leq d$ graph with n nodes. Then*

1. *If G is connected, the tester passes.*
2. *If G is ϵ-far from $P_n^{(d)}$*, $\Pr[\textit{tester fails}] \geq 2/3$.

Proof The first part of the theorem is trivial. For the second part, assuming G is ϵ-far from being connected, by the corollary we have

$$\begin{aligned}
\Pr\left[\text{s is in component of size } \leq \frac{8}{\epsilon d}\right] &= \frac{\#\text{ nodes in small components}}{n} \\
&\geq \frac{\#\text{ small components}}{n} \\
&= \frac{\epsilon d}{8}
\end{aligned}$$

Therefore,

$$\begin{aligned}
\Pr\left[\text{no s is in a small component}\right] &\leq (1-\frac{\epsilon d}{8})^m \\
&= (1-\frac{\epsilon d}{8})^{\frac{c}{\epsilon d}} \text{ for some constant c} \\
&\leq e^{-c'} \text{ where c' is a constant dependent on c}
\end{aligned}$$

Assuming that c is chosen appropriately, this can be made less than 1/3. ■

2 Estimating the Number of Connected Components of a Graph

Model Let G be a graph with degree $\leq d$, represented in the adjacency list model.

Goal Let C be the number of connected components of a graph. We seek an *additive* estimate, accurate to within $\pm\epsilon n$ of C.

Plan For a given node $u \in G$, define n_u to be the number of nodes in u's component.

Now, note the following:

Fact 4 *For any connected component $A \subseteq V$, we have that*

$$\sum_{u \in A} \frac{1}{n_u} = \sum_{u \in A} \frac{1}{|A|} = 1$$

Moreover, this implies

$$\sum_{u \in V} \frac{1}{n_u} = C$$

Thus, one way to calculate the number of connected components C, is to calculate n_u for every u, and use the formula above. This sounds like a pretty stupid idea, since it takes a total of $O(dn^2)$ time ($0(dn)$ time to calculate each n_u). We could alternatively just use a BFS to calculate the number of connected components in $O(nd)$ time. However, if we have a quick way to *estimate* the n_u's on just a few nodes u, then we can use standard sampling bounds to give us an approximation of the sum. In particular, we define our estimates of n_u, denoted $\hat{n}_u$, as follows:

Definition 5 $\hat{n}_u = \min\{n_u, 2/\epsilon\}$

Similarly, we can also define the estimate of C that we get by using the $\hat{n}_u$'s instead of the n_u's:

Definition 6 $\hat{C} = \sum_{u \in V} \frac{1}{\hat{n}_u}$

The following simple fact bounds the amount that our estimates can be off:

Fact 7 $0 \leq \frac{1}{\hat{n}_u} - \frac{1}{n_u} \leq \epsilon/2$. *Further, this implies* $\sum_{u \in V}(\frac{1}{\hat{n}_u} - \frac{1}{n_u}) = \hat{C} - C \leq \epsilon n/2$

Proof By the definition of $\hat{n}_u$, we have $\frac{1}{\hat{n}_u} \geq \frac{1}{n_u} \geq 0$. Now, either we have that $\hat{n}_u = n_u$, in which case $\frac{1}{n_u} - \frac{1}{\hat{n}_u}$ is just 0, or we have that $\hat{n}_u = 2/\epsilon$. In this case, we know that $\frac{\epsilon}{2} = \frac{1}{\hat{n}_u} \geq \frac{1}{n_u} \geq 0$, showing $\frac{1}{\hat{n}_u} - \frac{1}{n_u} \leq \epsilon/2$
■

Fact 8 *We can compute $\hat{n}_u$ in $O(d/\epsilon)$ time.*

Proof This follows simply from the fact that $\hat{n}_u$ is upper bounded by $2/\epsilon$. To calculate $\hat{n}_u$, we simply use a BFS to count the number of vertices in $u's$ component. If our count ever reaches $2/\epsilon$, we stop short and output $\hat{n}_u = 2/\epsilon$. If our count ends before then, we know $\hat{n}_u = n_u$. Either way, we run for at most $O(d/\epsilon)$ steps.
■

Finally, we explicitly write out the algorithm for approximating the number of connected components:

Algorithm *Approx_num_cc*(G, e)

1. Choose $\Theta(1/\epsilon^3)$ random nodes, labeled $u_1, ..., u_r$.
2. $\forall u_i$, compute $\hat{n}_{u_i}$.
3. Output $\tilde{C} = n(\frac{1}{r}\sum_{i=1}^{r} \frac{1}{\hat{n}_{u_i}})$

Here we've labeled the output $\tilde{C}$, to differentiate it from $\hat{C}$, the value we defined earlier, and C, the true number of connected components. In the expression of $\tilde{C}$, the value in parenthesis is the average value of $\frac{1}{\hat{n}_{u_i}}$ observed in the sampled nodes. Shortly, we will use Chernoff Bounds to compare this quantity with $\frac{1}{n}\hat{C}$, the average value of all the $\hat{n}_u$'s.

The runtime of the algorithm is $O(d/\epsilon^4)$. This comes from the fact that we choose $O(1/\epsilon^3)$ nodes, and then spend $O(d/\epsilon)$ time computing $\hat{n}_u$ for each one.

In order to prove that this algorithm does indeed give a good approximation, we will show that the value it outputs, $\tilde{C}$, is probably close to the value $\hat{C}$. Because we showed earlier that $\hat{C}$ is close to C, by the triangle inequality this will prove that $\tilde{C}$ is probably close to C. More formally, we prove

Theorem 9 $\Pr\left[\left|\tilde{C} - \hat{C}\right| \leq \epsilon n/2\right] \geq 3/4$

Proof

First note that for all i, $\epsilon/2 \leq 1/\hat{n}_{u_i} \leq 1$ (the first of these inequalities comes from the definition of $\hat{n}_{u_i}$, the second comes from the fact that u_i's component must contain at least one node). Now, by using the Chernoff bound, we have:

$$\Pr\left[\left|\frac{1}{r}\sum_{i=1}^{r}\frac{1}{\hat{n}_{u_i}} - E[\frac{1}{\hat{n}_u}]\right| \geq \frac{\epsilon}{2}E[\frac{1}{\hat{n}_u}]\right] \leq e^{-\Omega(rE[\frac{1}{\hat{n}_u}](\epsilon/2)^2)}$$

Here we have substituted $E[\frac{1}{\hat{n}_u}]$ for $\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\hat{n}_{u_i}} = \frac{1}{n}\hat{C}$. Note that this quantity is $\geq \frac{n\epsilon}{2}$, since for each i, $1/\hat{n}_{u_i} \geq \epsilon/2$. Simplifying, we get:

$$\Pr\left[\left|\frac{n}{r}\sum_{i=1}^{r}\frac{1}{\hat{n}_{u_i}} - \hat{C}\right| \geq \frac{\epsilon}{2}\hat{C}\right] \leq e^{-\Omega(\frac{a}{\epsilon^3}\cdot\frac{\hat{C}}{n}\cdot\frac{\epsilon^2}{4})} \text{ (a is some constant depending on r)}$$

$$\Pr\left[\left|\tilde{C} - \hat{C}\right| \geq \frac{\epsilon}{2}\hat{C}\right] \leq e^{-\Omega(\frac{a}{\epsilon^3}\cdot\frac{\epsilon}{2}\cdot\frac{\epsilon^2}{4})} = e^{\Omega(a/8)}$$

With an appropriate choice of r (and thus of the constant a), the right hand side can be made less than 1/4. ■

Corollary 10 $\Pr\left[\left|\tilde{C} - C\right| \leq \epsilon n\right] \geq 3/4$

Proof Using the triangle inequality, together with one of the previously proven facts, we know that when $\left|\tilde{C} - \hat{C}\right| \leq \epsilon n/2$, then

$$\left|\tilde{C} - C\right| \leq \left|\tilde{C} - \hat{C}\right| + \left|\hat{C} - C\right| \leq \epsilon n/2 + \epsilon n/2 = \epsilon n$$

Therefore

$$\Pr\left[\left|\tilde{C} - C\right| \leq \epsilon n\right] \geq \Pr\left[\left|\tilde{C} - \hat{C}\right| \leq \epsilon n/2\right] \geq 3/4$$

■

This completes our argument that with high probability, the algorithm outputs an estimate of the number of connected components, accurate to within an additive factor of $\pm\epsilon n$. Next time, we will see how we can use this algorithm as a subroutine in a different algorithm which estimates the minimum spanning tree of a graph, to within a *multiplicative* factor of $(1 \pm \epsilon)$.