CS 683 Advanced Design and Analysis of Algorithms Feb 4, 2008

Lecture 7: More on Graph Connectivity

Instructor: John Hopcroft *Scribes: Anna Blasiak, Vasumathi Raman*

1 Review

Disappearance of isolated vertices (Lecture 3)
$\rightarrow$ sharp threshold at $\frac{\log n}{n}$

Graph Connectivity(continued from last lecture)
1 of 3 things is true:

1. Graph Connected
2. There exist isolated vertices
3. No isolated vertices, graph not connected

If (3) disappears before (2) does, then the threshold for disappearance of isolated vertices is the threshoold for graph connectivity.

Disappearance of isolated vertices:
Let $x =$ number of isolated vertices.
$E[x] = n(1-p)^{(n-1)}$
Consider $p = \frac{c \ln n}{n}$: $E[x] = n\left(1 - \frac{c \ln n}{n}\right)^{n-1} = ne^{-c \ln n} = nn^{-c}$
if $c > 1$ $\lim_{n\to\infty} E[x] = 0$ $\Rightarrow$ probability of isolated vertex is zero
if $c < 1$ $\lim_{n\to\infty} E[x] = \infty$ second moment argument implies there exists an isolated vertex

The following argument from last lecture is wrong:
P(specific vertex isolated) $= (1-p)^{n-1}$ correct.
P(no vertex isolated) $= [1-(1-p)^{n-1}]^n$ wrong, not independent events.

Disappearance of components of size $k \geq 2, k = o(n)$:
Let x be the number of components of size k.

$E[x] = \binom{n}{k}$ Pr(connected)Pr(no edges out) $\leq \binom{n}{k}(1-p)^{k(n-k)}$

Let $p = \frac{c \ln n}{n}$
$E[x] \leq \frac{n^k}{k!}\left(1 - \frac{c \ln n}{n}\right)^{kn-k^2} = \frac{n^k}{k!} e^{-ck \ln n}\left(1 - \frac{c \ln n}{n}\right)^{-k^2}$

If k is constant then $= \frac{n^k}{k!} n^{-ck} \quad \to 0$ as $n \to \infty$.
For $k = o(\sqrt{n})$, possibly $\left(1 - \frac{c \ln n}{n}\right)^{-k^2} \to 1$.

For k not constant, we will prove this in a later lecture

2 The Gap

Why is the GCC so much bigger than the other components (and why are there no components of intermediate size)?
For $p = \frac{d}{n}, d \geq 1$,

$$\begin{array}{ccc} |\text{GCC}| & & |\text{other components}| \\ cn & \overset{gap}{\longleftrightarrow} & \log n \end{array}$$

Simple Algorithm for Searching the GCC
Thus we start from a single vertex and expand the "frontier" of discovered vertices by exploring vertices we haven't explored before.
Let $S =$ the expected number of discovered vertices, $F =$ the expected number of discovered but unexplored vertices, and $t =$ time (with one vertex explored at every time step). Then

$$\begin{aligned} \frac{dS}{dt} &= \text{expected number of edges explored} \times \text{probability of discovering a new vertex} \\ &= d\left(1 - \frac{s}{n}\right) \end{aligned}$$

Solution: $S(t) = n(1 - e^{-\frac{d}{n}t})$
Check: $\frac{dS}{dt} = de^{-\frac{d}{n}t} = d - d + de^{-\frac{d}{n}t} = d - \frac{dn}{n}(1 - e^{-\frac{d}{n}t}) = d\left(1 - \frac{s}{n}\right)$

Frontier F

$$\begin{aligned} &= S - t \text{ (exploring one vertex at each unit of time } t) \\ &= \left(n - ne^{-\frac{d}{n}t}\right) - t \end{aligned}$$

Normalized Frontier: $1 - e^{-\frac{d}{n}t} - \frac{t}{n}$ [1]

t can be at most n, since we cannot explore more than n nodes. If we take $x = \frac{t}{n}$, the (normalized) expected size of the frontier is $f(x) = 1 - e^{-dx} - x$. Fig 1.

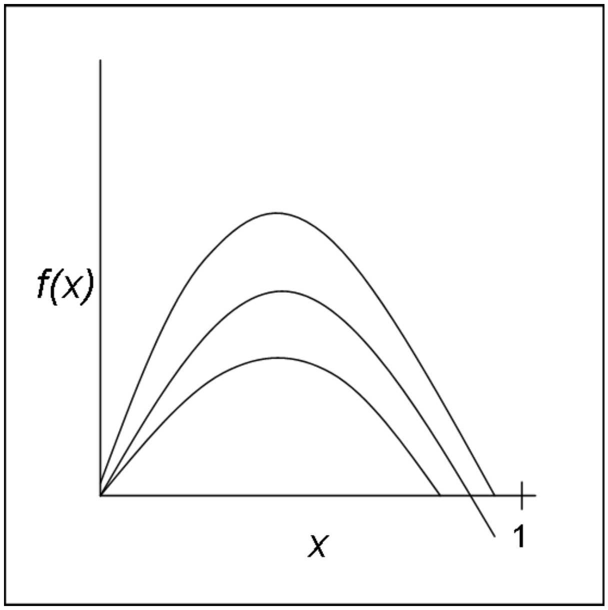

Figure 1: Normalized Expected Size of the Frontier

But when we proved that $\frac{dS}{dt} == d\left(1 - \frac{s}{n}\right)$, we were using $d =$ the expected value of the number of edges explored at time t. In reality, this is a random variable with probability distribution

$$\Pr[\text{degree} = d] = \binom{n}{d} p^d (1-p)^{n-d}$$

So we should really calculate the probability distribution of S, then take the expected value. But in this particular case (with this probability distribution and equation), the math works out. The proof of this would make a good portfolio exercise.

What are the actual sizes of the components?

If we consider the size of the frontier F to be a random variable, the above algorithm can only terminate when F hits 0 ($f(x) = 0$), and no more vertices are discovered. So we want non-zero root Θ of $f(x)$. This gives us $\frac{t}{n}$, where one vertex is explored at each time step. So $t = \Theta n$. The actual size of components is $\Theta n \pm \sqrt{n}$, and follows the distribution shown in Fig 2.

Note: The average size of a non-GCC component is $O(1)$, even though the maximum size is $\frac{logn}{n}$.

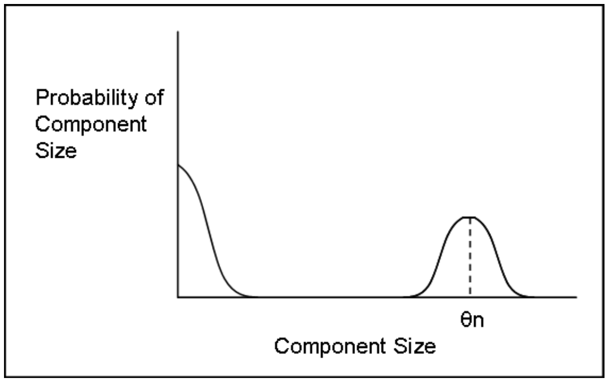

Figure 2: Distribution of Component Sizes

3 Directed Random Graphs

We have already studied undirected random graphs $G(n, p)$. Now we look at two kinds of directed graphs:

1. $\vec{G}(n, p)$: This is the same as $G(n, p)$, but for every undirected edge that is present in G, add 2 directed edges (one in either direction) $\vec{G}$. So if edge (u, v) exists in G, both (u, v) and (v, u) exist in G.

2. $D(n, p)$: Here, every directed edge $((u, v)$, $(v, u))$ is present with probability p.

Claim: The probability that the component containing vertex v in $G(n, p)$ is of size s equals the probability that the set of vertices reachable form v in $\vec{G}$ or D in s. **Proof:** It is clear for $G, \vec{G}$ since they are essentially the same graph. We need to show this for G, D. To do this, do a depth-first search (DFS), only generating edges (via a $\frac{1}{p}$ coin flip) when they are needed. Thus, for any pair of vertices, we only flip once for an edge between them (i.e. only in one direction). To be continued...

References

[1] Richard M. Karp. The transitive closure of a random digraph. *Random Struct. Algorithms*, 1(1):73–94, 1990.