CS 683 Advanced Design and Analysis of Algorithms February 22, 2008

Lecture 15: More on Generating Functions

Instructor: John Hopcroft *Scribe: Cristian Danescu Niculescu-Mizil, Yao Yue*

Growth Model with Preferential Attachment

We talked about growth models in Lecture 12 (Feb 15), now we discuss adding edges by preferential attachment. To achieve this, we create the graph as follows: at each unit of time t_i,

1 add a vertex, and label it with i

2 with probability δ add an edge from the new vertex to vertex picked with probability proportional to its degree.

Let $d_i(t)$ be expected degree of vertex i at time t.

$$\frac{d}{dt}d_i(t) = \delta \cdot \frac{d_i(t)}{2\delta t} = \frac{d_i(t)}{2t}$$
$$d_i(t) = at^{1/2}$$

To determine the value or a, we use the fact that at time t_i, the expected degree of vertex i is δ. Therefore,

$$d_i(t_i) = at_i^{1/2} = \delta$$
$$a = \frac{\delta}{sqrtt_i}$$
$$\Rightarrow d_i(t) = \delta\sqrt{\frac{t}{t_i}}$$

We are interested in the degree distribution of vertices. To obtain this, we first look at the cumulative distribution function (CDF) of degree of vertices. The expression of d_i above shows that it grows monotonically with time. For a given degree g and current time t, there exists a time t_g such that all vertices added before t_g are expected to have degree greater than g at time t, as shown in Figure 1.

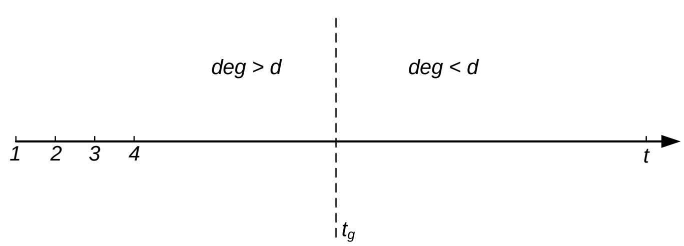

Figure 1: Expected degree of vertices added at different time

Using the monotonicity of d_i, the value of t_g can be calculated by solving the following equation:

$$\delta t_g \sqrt{t} = g$$
$$\frac{t}{t_i} = (\frac{g}{\delta})^2$$
$$t_i = (\frac{\delta}{g})^2 t$$

Consequently, the fraction of vertices having degree less than g at time t is $F(g) = 1 - (\frac{\delta}{g})^2$ (as shown in Figure 2), which is the CDF of degree of vertices. By taking the derivative the $F(g)$ we obtain the probability density function of degree of vertices $f(g) = 2\delta^2 d^{-3}$. This is a power law distribution of order 3. Interestingly, study reveals that the World Wide Wide also exhibits a power law distribution of order approximately 3.

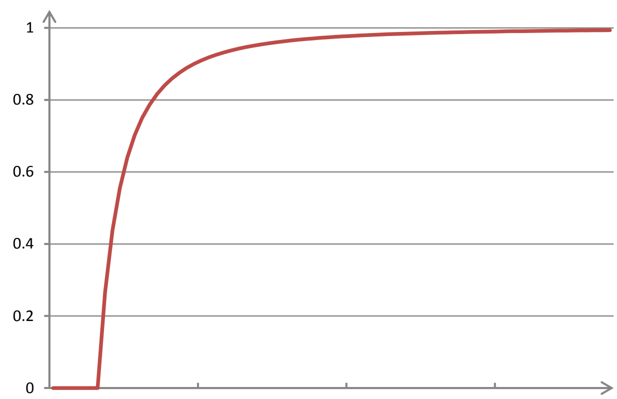

Figure 2: Fraction of vertices having degree $\leq d$

Small World Graphs

The concept of "Six Degrees of Separation" is known since Stanley Milgram's famous experiment in the 1960's. We are now analyzing this phenomenon from an algorithmic point of view, conforming with Jon Kleinberg's work presented in [1].

The model employed is a two dimensional grid network (see Figure 3) with n nodes, in which additional edges (called long range edges) are added as follows: for each vertex u add an edge from u to a random vertex v with probability $P(u \to v) \propto d^{-r}(u, v)$ where d is the lattice distance.

The question is wether a short path exists between any pair of nodes and if there is an efficient local algorithm for finding such a path. Short path means a path of length $\propto \log(n)$, local algorithm means that from each node can inspect only its neighbors. For example, in Figure 3, s is an arbitrarily chosen source and t is a destination, there must be a path from s to t, but not necessarily short path. A local algorithm can inspect the red nodes (connected through the lattice edges) and the blue node (connected through a long range edge). If each such inspection costs one unit of time then an efficient algorithm would be poly-logarithmical.

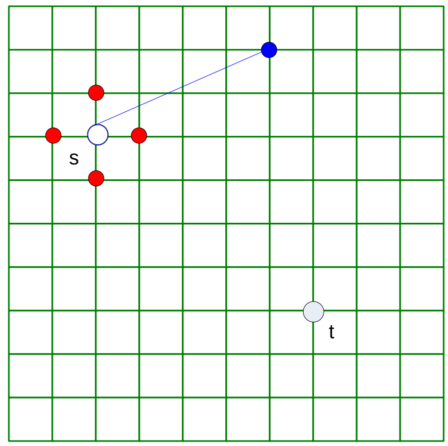

Figure 3: Two dimensional grid network with addition edges

The major results can be summarized as follows:

$r < 2$ a short path exists but there is no local algorithm for finding this path

$r = 2$ there exists an algorithm to find a short path

$r > 2$ a short path may not exist; and even if it does exist, there is no local algorithm to find it

In the following lecture a proof will be presented; for now we restrict ourselves to discuss the results and the intuition behind them.

For $r < 2$, consider the extreme case where $r = 0$. This means all vertices are equally likely for end points. With high probability there exists a short path [1], but we do not have an efficient algorithm to find it. Actually, we show here that it is impossible to find such a path in $n^{1/4}$ time.

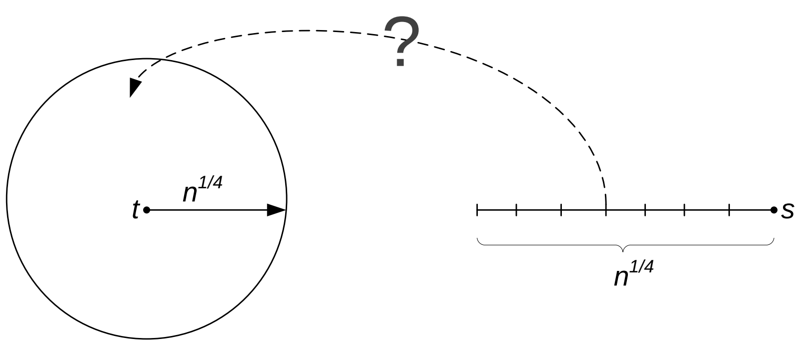

Figure 4: Can we find a long edge in $n^{1/4}$ steps?

As shown in Figure 4, consider the destination t and its neighborhood $N(t)$ of radius $n^{1/4}$, which contains $n^{1/2}$ vertices. Starting from the source s, we look into no more than $n^{1/4}$ edges to find a path that leads to some node in $N(t)$. These $n^{1/4}$ edges can start from as many as $n^{1/4}$ vertices. For each of these edges, the probability that it reaches some endpoint in $N(t)$ is $\frac{n^{1/2}}{n}$. The probability that it does not hit $N(t)$ is then $1 - \frac{n^{1/2}}{n}$. Therefore, for all $n^{1/4}$ edges, the probability that none reaches $N(t)$ is $(1 - \frac{n^{1/2}}{n})^{n^{1/4}}$. Since

$$\lim_{n\to\infty} (1 - \frac{n^{1/2}}{n})^{n^{1/4}} = \lim_{n\to\infty} (1 - \frac{n^{1/2}}{n})^{\frac{n^{1/2}}{n^{1/4}}} = \lim_{n\to\infty} (e^{-1})^{n^{1/4}} = 1$$

the probability that we cannot find such a path converges to 1 when n is large.
For $r = 2$, there exists a very simple algorithm to find a short path.

Algorithm: At each step, select an edge that gets you closest to the destination.

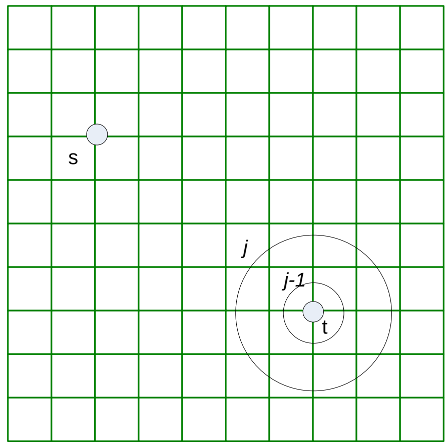

Figure 5: Phase definition based on distances to destination

As shown in Figure 5,if the distance of the last vertex on the path is in $\left(2^j, 2^{j-1}\right]$ of t, we say that the algorithm is in phase j.[1]

References

[1] Jon Kleinberg, *The Small-World Phenomenon: An Algorithmic Perspective*, SIGKDD'00: Proceedings of the 32nd ACM Symposium on Theory of Computing, 2000.

[1]This definition will be used in the following lecture for the analysis of the algorithm.