

Math 443/543 Graph Theory Notes 10: Small world phenomenon and decentralized search

David Glickenstein

November 10, 2008

1 Small world phenomenon

The small world phenomenon is the principle that all people are linked by short chains of acquaintances. For instance, “six degrees of separation” is the idea that everyone is linked by chains of people at most 6 persons long. That is, given any persons A and B , there are a chain of people such that A knows C who knows D who knows E who knows F who knows G who knows H who knows B . Other examples are the Kevin Bacon game (that every actor can be linked to Kevin Bacon via actors they have acted with) and the Erdos number (related to collaborations on mathematics papers).

The first real empirical evidence is from an experiment by the social psychologist Stanley Milgram during the 1960s. He tested it in the following way. He chose a target person, as stockbroker living near Boston. And he took a random person in, say, Nebraska and asked them to get a letter to the target. Each person was only allowed to forward the letter to a single acquaintance they know on a first name basis. The only things the person knew about the target was his name, occupation, address, and some personal information. One can think of the letter as tracing a path in the graph of all people who know each other (the acquaintance graph). Milgram found that sometimes the letters made it to the target, and among those that did make it to the target, the median length of the path was around 6, hence the idea of 6 degrees of separation.

For our purposes, we want to understand why a graph like the acquaintance graph might have short paths, and then think about how to find short paths.

Simplistic idea: Suppose each person knows 100 others on a first name basis. Then in 2 steps, one should be able to reach 10,000 people, in three steps, $100 \times 100 \times 100 = 1,000,000$ people, and then by 5 steps 10 billion people. The population of the US is estimated just above 300 million and the population of the world is estimated to be less than 7 billion.

This idea is a bit too simplistic, since many people who you know on a first name basis also know each other. This means that, in reality, 5 steps will not reach nearly this many people. The implicit assumption made above is that everyone knows 100 random people of all possible people.

Which brings us to the Watts-Strogatz model. The idea is that there is a balance between the random association of people and the notion that people know others based on proximity. Watts and Strogatz considered a model where each vertex is placed around a ring, and proximity is measured by attaching each vertex to all other vertices within some proximity (such as 2). Then, with some probability p , each edge is rewired to a random edge, not allowing multiple edges. Thus, for $p = 0$, there is no randomness, for $p = 1$, it is entirely random, and for p in between there is some amount of randomness. For each such graph, They measured two quantities:

1. The characteristic path length L . This is the typical separation between two vertices, computed as the length of shortest paths averaged over all pairs of vertices.
2. The clustering coefficient C . For each vertex v , one can compute C_v as the number of edges between neighbors of v , divided by the total number possible (which is $k_v(k_v - 1)/2$ if v has k_v neighbors). C is the average of these over all vertices.

One can look at Figure 2 in WS to see the relationship for different values of p . The interesting phenomenon is that the characteristic path length decreases well before the clustering coefficient does. Thus Watts and Strogatz suggest the following definition of small world phenomenon.

Definition 1 *A graph exhibits small world behavior if $L \gtrsim L_r$ and $C \gg C_r$, where L_r and C_r are the characteristic path length and clustering coefficient for a random graph and $A \gtrsim B$ means A is larger than B but close to B and $A \gg B$ means that A is much larger than B .*

In other words, small world phenomenon has a lot of clustering, but still short paths. See WS for some estimates of L and C for several large graphs which seem to exhibit small world behavior.

We can look at another type of graph which can exhibit small-world behavior. Consider an $n \times n$ grid, and suppose there are edges between every other grid point within radius r , that is, any other edge which can be reached by r steps by moving along the grid. This gives a nice notion of geometric proximity. Now suppose that every vertex is connected to k vertices which are randomly distributed throughout the entire network. Then each vertex has roughly $r^2 + k$ edges. Notice that by using only random edges, one can reach k^2 vertices in two steps, and k^j vertices in j steps. Thus, on the average, one could reach all n^2 vertices in $2 \log_k n = c \log n$ steps. Note that by using nonrandom edges only, it would take on average a multiple of n steps. Thus for large n , the shortest paths act like random edges. However, the clustering is quite high. Considering only the grid vertices, clustering is like

$$C^{-1} = \frac{\frac{1}{2}(r^2 + k)(r^2 + k - 1)}{\frac{1}{2}r^2(r^2 - 1)} = 1 + \frac{r^2(2k - 1) + k(k - 1)}{r^2(r^2 - 1)}.$$

Notice that this is quite close to 1 if k is very small compared with r^2 .

In fact, one could consider a network where k is a number between 0 and 1, representing a fact that we are actually only allowing random edges for some vertices (not every vertex). Conceptually, we can group parts of the $n \times n$ grid into “towns,” each of which has some number of random edges. We then produce a different graph which is similar to our original formulation. One can get from one person to another in the town quite easily using the edges based on proximity.

2 Decentralized search

How was it that participants in the Milgram experiment were able to get the letters to their targets without any centralized help? They did not find paths and choose the shortest, but were able to find a short path knowing only minimal information. Our next task is to try to find models for this.

The model is that we have a starting node s and a target node t . Suppose s only knows where t is, but not how to reach t quickly (because s does not know about the random edges). This is also true for each intermediate node along the path. Of course, an obvious strategy is to just use the edges based on proximity and move closer and closer to where t is located. This will produce a delivery time on the order of the distance from s to t . (Delivery time is the number of edges in the path from s to t .) We wish to find an algorithm that has, on the average, a faster delivery time than this naive algorithm. Since we already know that the shortest path is, on the average, proportional to $\log n$, we would hope to find an algorithm which achieves this.

It turns out that our grid model does not admit such an algorithm. Kleinberg was able to show that it takes at least $n^{2/3}$. The main idea is that it is quite hard to choose which random edges to pick.

We use a new model which is slightly more general than the network model. Instead of adding k random edges to the nodes within r grid steps, for each pair of edges v and w , we add an edge between them with probability proportional to $d(v, w)^{-q}$, for some value q . We call q the *clustering exponent*. If $q = 0$, then we have equal probabilities and so we have the previous grid model. If q is very large, then we have very few random edges at all. So $q = 0$ has the most randomness and the randomness decreases as q increases.

The problem we will find is that if there is too much randomness, it is hard to exploit it in an algorithm. But if there is not enough randomness, there aren't enough short paths.

Theorem 2 (Kleinberg) *If $q = 2$ then there is an algorithm for a decentralized search with delivery time proportional to $(\log n)^2$. If $q \neq 2$, then any algorithm for a decentralized search has a delivery time proportional to n^c for some c (which depends on q).*

We will give an idea of why $q = 2$ allows for a good decentralized search algorithm.

Consider a vertex v and a distance d . we can look at the ring of all grid points at least d grid steps from v but less than $2d$ grid steps from v . For an arbitrary edge from v , the probability that it links to a particular point in this ring is roughly d^{-q} . There are approximately d^2 vertices in this ring, and so the probability that an edge from v links to any point in the ring is proportional to d^{2-q} . If $q = 2$, this does not depend on d !

Henceforth, let's assume that $r = 1$, and so each vertex is attached to 4 other vertices (unless the vertex is on the boundary). So for a fixed vertex v , there are 4 vertices a distance 1 away from v , 8 vertices a distance 2 away, 12 vertices a distance 3 away, and, in general, $4n$ vertices a distance n away from v . Also, let's assume that each vertex has one random edge. The algorithm is quite simple, choose the adjacent vertex which is closest to the target. The key fact we will show is that this does find the target on average in time $(\log n)^2$.

First, we need to be a bit more formal about how we choose the random edges. Given each vertex v , we will assign an edge to another vertex u with probability proportional to $d(u, v)^{-2}$, where d is the distance on the grid. In order to make this a probability, that means that the probability that the edge is uv is equal to

$$P_{v,u} = \frac{d(u, v)^{-2}}{\sum_{w \neq v} d(w, v)^{-2}}.$$

We can estimate the denominator by

$$\begin{aligned} \sum_{w \neq v} d(w, v)^{-2} &\leq \sum_{j=1}^{2n} (4j) j^{-2} \\ &= 4 \sum_{j=1}^{2n} \frac{1}{j} \\ &\leq 4 \left(1 + \int_1^{2n} \frac{1}{x} dx \right) \\ &= 4 + 4 \ln(2n) \\ &\leq 4 \ln(2en) \\ &\leq 4 \ln(6n) \end{aligned}$$

where we have used the fact that there are at most $4j$ vertices of distance j from any vertex, and that the largest distance is $2n$. This means that the probability $P_{v,u}$ is at least

$$\frac{d(u, v)^{-2}}{4 \ln(6n)}.$$

Now, we will measure progress by looking at the domain in phases.

Definition 3 *The search is in phase j if the distance of the current vertex to t is between 2^j and 2^{j+1} .*

Note that the largest value of j is equal to the least integer greater than $\log_2 n - 1$.

Definition 4 Let X_j be the total number of steps spent in phase j . The expected number of steps spent in phase j is equal to

$$E_j = \sum_{i=1}^{\infty} i \Pr[X_j = i].$$

We see that we can calculate this as follows. The probability that $X_j \geq i$ is equal to

$$\Pr[X_j \geq i] = \sum_{k=0}^{\infty} \Pr[X_j = i + k].$$

We see that we can rearrange this definition of expected steps to be

$$\begin{aligned} E_j &= \sum_{i=1}^{\infty} i \Pr[X_j = i] \\ &= \sum_{i=1}^{\infty} \sum_{k=0}^{\infty} \Pr[X_j = i + k] \\ &= \sum_{i=1}^{\infty} \Pr[X_j \geq i]. \end{aligned}$$

We can now estimate

$$\Pr[X_j \geq i] = \left(1 - \sum_{u \in B_j} P_{u,v} \right)^{i-1}$$

since $1 - \sum_{u \in B_j} P_{u,v}$ is the probability that we stay in phase j for the next step. So we need a lower bound on

$$\begin{aligned} \sum_{u \in B_j} P_{u,v} &\geq \sum_{u \in B_j} \frac{d(u,v)^{-2}}{4 \ln(6n)} \\ &= \frac{1}{4 \ln(6n)} \sum_{i=1}^{2^j} (4i) i^{-2} \\ &\leq \frac{1}{\ln(6n)} \left(1 + \int_1^{2^j} \frac{1}{x} dx \right) \end{aligned}$$

Let B_j be the collection of vertices which are a distance of 2^j or less from t .

Proposition 5 There are at least 2^{2j-1} vertices in B_j .

Proof. The worst case scenario is if t is in the corner, in which case there are $i + 1$ vertices a distance i away. Thus there are at least

$$\begin{aligned} \sum_{i=0}^{2^j} (i + 1) &\geq \sum_{i=1}^{2^j} i = \frac{1}{2} (2^j + 1) 2^j \\ &\geq 2^{2j-1} \end{aligned}$$

vertices in B_j . ■

Proposition 6 *Suppose a message is at stage j . The probability a message enters B_j after one step is at least*

$$\frac{1}{128 \ln(6n)}.$$

Proof. Let u be the current vertex where the message is. We need to estimate

$$\sum_{v \in B_j} P_{u,v}$$

from below. Note that every vertex $v \in B_j$ is a distance less than or equal to $2^j + 2^{j+1} = 3(2^j) < 2^{j+2}$ from the current vertex u since we are in stage j . Thus

$$P_{u,v} \geq \frac{d(u,v)^{-2}}{4 \ln(6n)} > \frac{1}{4(2^{2j+4}) \ln(6n)}.$$

Since there are at least 2^{2j-1} vertices in B_j , we see that the probability of entering B_j is thus at least

$$\frac{2^{2j-1}}{4(2^{2j+4}) \ln(6n)} = \frac{1}{128 \ln(6n)}.$$

■

Proposition 7 *The search stays in phase j for, on the average, fewer than $128 \ln(6n)$ steps.*

Note, that when we say,

Proof. Since the probability that we stay in phase j for more than one step is equal to the probability that we do not enter B_j after one step,

$$\Pr[X_j \geq 1] \leq 1 - \frac{1}{128 \ln(6n)}.$$

We can compute

$$\Pr[X_j \geq i] = \left(1 - \frac{1}{128 \ln(6n)}\right)^{i-1}$$

and hence the expected value number of steps in page j is

$$E = \sum_{i=1}^{\infty} \Pr[X_j \geq i] \leq \sum_{i=1}^{\infty} \left(1 - \frac{1}{128 \ln(6n)}\right)^{i-1} = 128 \ln(6n).$$

■

Theorem 8 *The decentralized search will, on the average, find the target in fewer than $\alpha (\log n)^2$, for some α which does not depend on n .*

Proof. The expected time is equal to the sum of the expected lengths from one step to the next. Since there are $\log_2 n$ steps, we see that the expected time is less than

$$128 \ln(6n) \log_2 n = \alpha (\log n)^2,$$

since

$$\begin{aligned} 128 \ln(6n) \log_2 n &= \frac{128}{\log e \log 2} (\log 6n) (\log n) \\ &= \frac{128}{\log e \log 2} [(\log 6) (\log n) + (\log n)^2] \\ &\leq \frac{256}{\log e \log 2} (\log n)^2 \end{aligned}$$

for large n . ■