Inference, Models and Simulation for Complex Systems — CSCI 7000-003
Lecture 8 — 23 September 2010
Prof. Aaron Clauset

2 Structural measures of networks (continued)

2.3 Shortest paths

A *path* in a network is a sequence of vertices $x \to y \to \cdots \to z$ such that each consecutive pair of vertices $i \to j$ is connected by an edge (i, j) in the network. A *shortest path*, which is also called a *geodesic path* (from geometry), is the shortest of all possible paths between two vertices. Shortest paths are examples of self-avoiding paths, meaning that they do not intersect themselves. The length of the longest of these geodesic paths is called the *diameter* of a network, and is meant to evoke the notion of a volume in a metric space.[1]

In most mathematical network models, as the network size grows, that is, as $n \to \infty$, the diameter also grows, albeit more slowly (for instance, see Section 2.3.2 below). For instance, in most random graph models, the diameter grows like $O(\log n)$ [or sometimes $O(\log \log n)$]. In a 2005 KDD paper, Jure Leskovec, Jon Kleinberg and Christos Faloutsos showed evidence that some empirical networks, however, exhibit the opposite behavior—their diameter *shrinks* as the network grows or as time progresses. They called this process *densification*, and suggested that, in networks like the scientific citation network, interdisciplinary papers, which cite articles from multiple disciplines, are the key instigator of the shrinkage process.

2.3.1 Measuring the diameter

Measuring the diameter of a network requires computing a $n \times n$ matrix containing each of the pairwise topological distances d_{ij} for all pairs of nodes i, j. This is equivalent to the *All Pairs Shortest Paths* problem in graph theory, which can be solved efficiently in a variety of ways.

Depending on whether the network is weighted or unweighted, directed or undirected, etc., we could use something like a Breadth-First Search (BFS) tree, the Bellman-Ford algorithm, Dijkstra's algorithm or even a minimum-spanning tree algorithm like Kruskal's or Prim's to solve the *Single Source Shortest Paths* problem. Each of these algorithms returns a $n \times 1$ vector of distances from a single input vertex i to each of the other n vertices. If we repeat this procedure for each vertex i, we can build up the $n \times n$ pairwise distance matrix. In the simplest case—an unweighted, undirected network—one BFS takes $O(n + m)$ time, so the total running time is $O(n^2 + mn)$. For

[1] *Eularian* and *Hamiltonian* paths, which traverse every edge and every node exactly once, respectively are examples of other special kinds of paths. These, however, appear relatively infrequently in the study of networks.

sparse graphs, this is a relatively fast $O(n^2)$ but becomes a slow $O(n^3)$ for dense graphs.

Directed or weighted networks must be analyzed using something other than BFS, for instance, the Floyd-Warshall dynamic programming algorithm or Johnson's algorithm. Floyd-Warshall takes $O(n^3)$ time in the worst case, which doesn't scale up to large networks ($n > 10^5$ or so).

Such large networks also present problem with memory usage: any algorithm that first builds up the all-pairs shortest path distance matrix requires $O(n^2)$ memory, which can be prohibitive for large n. However, if we are only concerned with estimating the diameter, we don't need to store the entire pairwise distance matrix. Instead, we can use a solution to the *Single Source Shortest Paths* and keep track of the largest distance. This kind of approach takes only $O(n+m)$ memory (mainly to store the graph structure; storing the distances themselves takes only $O(n)$ memory plus some work space, which may add a $\log n$ factor).

For truly enormous networks, even this approach can be problematic and instead we must simply estimate the diameter via a sampling approach. If we choose pairs of vertices uniformly at random and measure their geodesic distance, we can estimate the diameter as the largest geodesic distance for a large (but manageable) number of iid samples of pairs of vertices; the accuracy of our estimate now depends on how well we sample the network, and how pathological its structure.

In general, geodesic paths play a crucial role in computing many other measures of network structure, as we'll see below in Section 2.5. And, the distribution of pairwise distances $\Pr(d)$ can itself be an interesting way to characterize the structure of a network.[2]

2.3.2 Small worlds and network diameter

In mathematical models of networks, the diameter plays a special role and can often be shown to vary in a clean functional way with the size of the network. If the diameter grows extremely slowly as a function of the network, e.g., $O(\log n)$, a network can be said to exhibit the *small world* property. The idea originally comes from a seminal study in social networks by the American sociology Stanley Milgram (1933 –1984).[3] Milgram sent letters to "randomly selected" individuals in Omaha, Nebraska and Wichita, Kansas with instructions asking them to please pass the letter (and instructions) to a close friend of theirs who either knew or might be likely to know a particular doctor in Boston. Before the passed along the letter, they should write their name on a roster to

[2] At one point early in the history of modern network theory, it was boldly proclaimed that there were only a few summary statistics you ever need to know about the structure of any network in order to fully characterize it. These were the degree distribution, the clustering coefficient and the pairwise distance distribution. If this claim had been true, network theory would be a much less interesting field than it currently is!

[3] The term "six degree of separation" is not due to Milgram, but comes from a play written by John Guare in 1990. The play was subsequently made into a movie of the same name starring Will Smith in 1993, and, ironically, not starring Kevin Bacon.

record the chain of message passing. Of the 64 letters that eventually reached the doctor—a small fraction of those sent out—the average length was only 5.5, not hundreds, and a legend was born.

Duncan Watts and Steve Strogatz, in a 1998 *Science* paper, studied this phenomenon using a simple mathematical model now called the "small world model." In this model, vertices are arranged on a 1-dimentional circular lattice (i.e., a "ring" network) and connected with their k nearest neighbors. Then, with probability p, each edge is rewired to join a uniformly random pair of vertices. Thus, $p = 0$ is the fully ordered limit where the clustering coefficient is largest and the diameter is $O(n)$; when $p = 1$, the network is fully disordered, the clustering coefficient is close to zero and the diameter is $O(\log n)$. See Figure 2 below.

Watts and Strogatz found that the diameter of the network collapses from $O(n)$, a "big world" where the longest geodesic path between two nodes traverses almost the entire network, to $O(\log n)$, a "small world" where the longest geodesic traverses a vanishingly small fraction of the network, when only a small fraction of edges have been randomly rewired, well before the clustering coefficient reaches zero. This behavior reminded them of some properties of social networks, which have small diameter (as evidenced by Milgram's study) but a high density of triangles,[4] which most models of random graphs to not exhibit simultaneously.

This small-worlds result is interesting for several reasons. First, it exemplifies an early effort to understand, using a toy model, how empirical networks can be different from simple random graph models. (In Lecture 8, we'll study random graph models in more detail.) Second, it shows that some measures of network structure can be extremely sensitive to uncertainty in the network structure. That is, suppose we were studying an empirical network that happened to be generated by the small-world model, where p was in the intermediate rewiring regime. If we happened to unluckily miss those few "long range" connections in our sampling of the network topology, then we would dramatically overestimate the network's true diameter. In contrast, other measures of network structure, such as the degree distribution or the clustering coefficient, are much less sensitive to such sampling errors. On the other hand, if the sampling is *biased*, even these can be highly inaccurate.[5]

[4]The small worlds model was generalized to d-dimensions by Jon Kleinberg in 2000, who further showed that under a particular distribution of the long-range links, greedy routing is highly efficient and takes only $O(\log^2 n)$ steps in expectation. Some of my own early work showed that this result can be achieved dynamically and adaptively using a simple rewiring rule. There's a potential independent project here, if anyone is interested.

[5]This is the case for sampling the degree distribution by using a breadth-first spanning tree, which generates highly biased degree distributions, even under relatively benign circumstances. This kind of sampling is basically what has been done for estimating the Internet's topology at the Internet Protocol (IP) level (which is distinct from measuring it at other places in the network protocol stack and from measuring it at the Autonomous System (AS) level) by using traceroute to measure paths from a single source to many destination IP addresses.

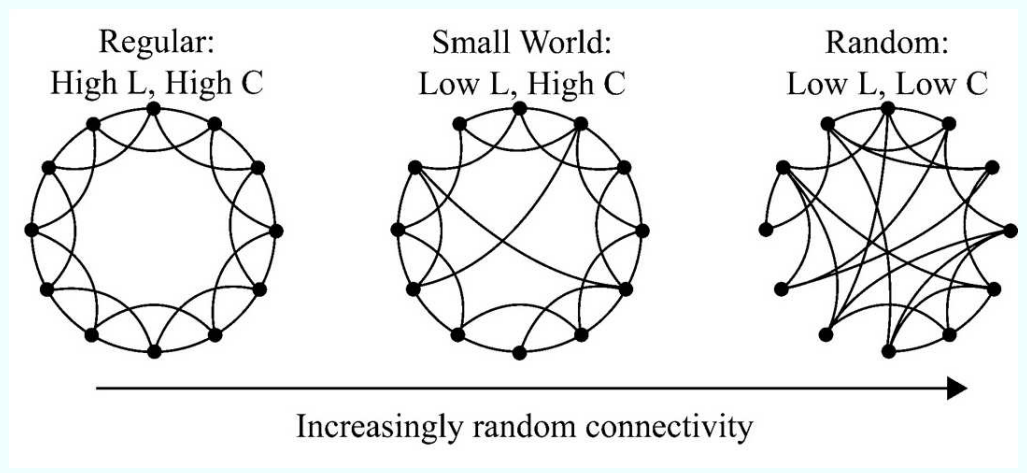

Figure 1: Watts and Strogatz's schematic illustrating the small worlds model.

2.4 Components

If there is no path between some pair of vertices in a network, the network is said to be *disconnected*; and, if every pair of vertices is connected by some path, then the network is *connected*.

Groups of vertices that are themselves connected, even if they are, as a group, disconnected from the rest of the network are called a *component*. In directed networks, if a group of nodes is connected only if we ignore the directionality of the edges, the group is called a *weakly connected component*. It's called a strongly connected component if we don't have to ignore directionality. In directed networks like the World Wide Web, an *out-component* of some vertex i is the set of vertices that can be reached by some path starting at i. Similarly, an *in-component* for some i is the set of vertices that can reach i by some path.

In mathematical models, an object called the *giant component* is often discussed; this is a component whose size is a finite fraction of the entire network, that is, a component with $O(n)$ vertices. (The term "giant component" is meaningless in most empirical situations since it's only defined asymptotically. For empirical networks with multiple components, it's typical to focus mainly on the largest component.) We'll revisit the notion of a giant component later, when we meet Erdös-Rényi random graphs.

2.4.1 Measuring components

The components of a network can be identified using standard graph algorithms. For undirected, unweighted networks, a BFS approach again works well (or rather, any algorithm that constructs a spanning forest will suffice), and techniques like Kruskal's algorithm should be used for other kinds of networks, e.g., weighted graphs.

The basic algorithm looks like this

```
for i = 1 to n
   label each vertex as unmarked
end
k = 0
for i = 1 to n
   if vertex i is unmarked
      k = k + 1
      grow a MST (or BFS tree) from vertex i
      mark every vertex in the MST as belonging to component k
   end
end
```

That is, we set up a $n \times 1$ vector that will store the component name of each vertex in the graph. Then, we loop through all the vertices, if a vertex is not already a member of some component, we grow a BFS or MST from that vertex in order to get the identities of all the vertices in the same component. The result is a $n \times 1$ vector in which the ith location gives the component name of vertex i, and a number k that tells us the number of components discovered. Because this algorithm takes no more time than a regular BFS, it runs in $O(n + m)$ time.

Identifying other kinds of components require a slightly different approach. For instance, strongly connected components can be identified using an application of Depth-First Search (DFS) trees in $O(n + m)$ time.

Once we have the components labeled and counted, we can tabulate their respective sizes s_ℓ, i.e., the number of vertices with component label ℓ. From there, we often compute some kind of summary statistic, like the mean size $\langle s \rangle$ or even the entire component size distribution $\Pr(s)$.

2.5 Centrality

The idea that some vertices are more important than others is ubiquitous but generally only meaningful in the context of a specific definition of *importance*. Often importance is related to some kind of dynamical process, e.g., social influence and opinion dynamics, but sometimes it is defined

purely structurally. Generally, it's best to measure importance directly, perhaps by simulating the dynamical process and measuring each vertex's contributions. A more common approach is to use a structural proxy, which is hoped to correlate strongly with the dynamical importance we actually care about.

There are a number of such topological measures of importance or *centrality*. We'll go through a few of the more common ones. Note that each of these measures has, at its root, a set of theoretical ideas that it's attempting to formalize and quantify and many are highly intuitive. As a warning, however, probably none are actually a direct measure of what we're really interested in and the assumptions may be completely ridiculous in some particular context.

2.5.1 Closeness

The *closeness centrality* draws an analogy with spatially embedded structures and measures of how topologically central a vertex is, with respect to the distance from a vertex i to all other vertices j. If we let d_{ij} denote is the length of the geodesic path from i to j (which we've measured by computing the all-pairs shortest path distance matrix), then the closeness centrality of a vertex i is defined as the harmonic mean of the distances from i

$$c_i = \frac{1}{n-1} \sum_{j(\neq i)} \frac{1}{d_{ij}} \quad , \tag{1}$$

where $d_{ij} = \infty$ if there is no path between i and j. (Note: we exclude the term $d_{ii} = 0$ since this would explode the sum.) This vertex-level quantity[6] thus measures how close is every other node in the network to vertex i, and larger values indicate more central vertices. This measure can be aggregated up to the network level by using the same harmonic mean trick as above on the closeness scores

$$\ell' = n \Big/ \sum_i c_i \quad . \tag{2}$$

Closeness can be generalized further by considering not just the geodesic path lengths, but a weighted sum of the length of all paths, with the shortest path getting the largest weight. This version, however, is rarely used.

2.5.2 Betweenness

The *betweenness centrality* is a measure of how many shortest paths run through a node or edge. The idea is that if we have two relatively disconnected components who are connected through a

[6]This definition of the closeness centrality is slightly different from others you might see in the literature; we use it here because it has nicer properties than the others.

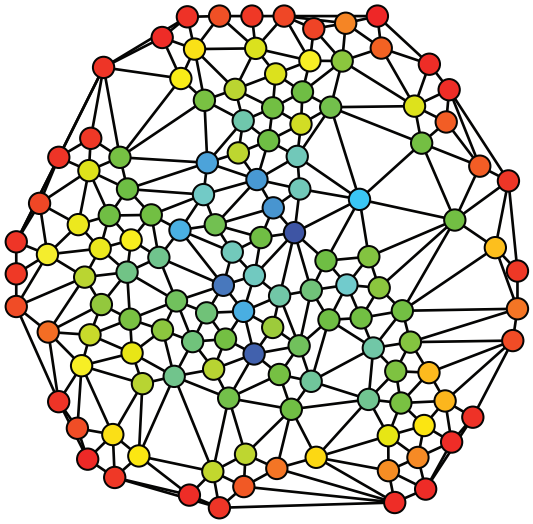

Figure 2: The hue of a vertex gives its betweenness centrality score, with red being low and blue being high (image from wikipedia).

small number of nodes or edges, then those are acting like "bridges" between the two groups. The betweenness centrality is defined mathematically as

$$b_i = \sum_{jk} \frac{\#\text{ geodesic paths } j \to \cdots \to i \to \cdots \to k}{\#\text{ geodesic paths } j \to \cdots \to k} = \sum_{jk} \frac{\sigma_{jk}(i)}{\sigma_{jk}} \ , \tag{3}$$

where σ_{jk} is the number of geodesic (shortest) paths between j and k, and $\sigma_{jk}(i)$ is the number of such paths that cross vertex i. Thus, if a vertex i has a high value of betweenness b_i, then many of the shortest paths cross it. Vertices that connect otherwise disconnected communities will thus score highly on this measure. In contrast, vertices in the middle of dense subgraphs will have a low score as very few shortest paths will pass through them. Figure 2 shows an illustrative example, from the wikipedia page on betweenness centrality.

Computing the closeness or betweenness centrality scores requires computing or estimating (via sampling) the all-pairs shortest paths distance matrix d_{ij}, and thus typically takes $O(n^3)$ for general networks, and $O(n^2)$ for sparse unweighted, undirected networks.

2.5.3 Degree centrality, Eigenvector centrality, Katz centrality and PageRank

Another family of centrality measures is based on measuring the degrees of nodes. The most basic measure is the degree centrality, which is equivalent to the vertex-level connectance measure we

saw in Lecture 7. Generalizations, which increase a node's centrality if they are themselves connected to other highly central nodes, include *eigenvector centrality* (the value in the ith entry of the eigenvector associated with the largest eigenvalue of the adjacency matrix), *Katz centrality* (a generalization of eigenvector centrality in which a free parameter α weights the relative contributions of connected central vertices), and *PageRank* (a variation of the Katz centrality that normalizes the contributions of highly central neighbors by their degree).

Computation here can often be done very quickly, because we are primarily interested only in the eigenvector of the adjacency matrix associated with the largest eigenvector. Diagonalization using a Fast Fourier Transform can be done in $O(n^2 \log n)$ time; we can also use the matrix power method; or we can simulate the process of random walkers moving on the graph structure as their stationary distribution is equivalent to the values in target eigenvector.

For a very good discussion of this family of measures, see Chapter 7.1–7.4 of M.E.J. Newman "Networks: An Introduction." Oxford Press (2010).

2.6 Homophily and assortative mixing

As a last measure of local network structure, consider the tendency for similar vertices to be connected. In sociology, this tendency is called *homophily* or *assortative mixing* and has a powerful effect on shaping the structure of social networks. Almost every nodal attribute you might care to measure in a social network exhibits homophily in the network structure: race, gender, age, political preferences, sports preferences, etc. all exhibit homophily. Its presence makes understanding causality of certain associations extremely difficult, particularly when the attributes are somewhat flexible. That is, are i and j connected because they have similar attributes, or are their attributes similar because they are connected? In general, it's difficult to distinguish these hypotheses, and thus we say that they are *confounded*.

We can quantify the degree of homophily in a network by tabulating an association matrix M, in which the M_{ij} value gives the fraction of connections from vertices with attribute i to vertices with attribute j. If attributes are simple, e.g., ethnicity or age in a social network, and the attributes are known for every vertex, this matrix can be tabulated quickly. Strongly assortative structure appears as a strong diagonal component in this matrix, and *disassortative* structure is a strong off-diagonal component. We can roughly quantify this dichotomy by computing a standard Pearson correlation coefficient on this matrix

$$r = \frac{\sum_{ij}(A_{ij} - k_i k_j/2m)x_i x_j}{\sum_{ij}(k_i \delta_{ij} - k_i k_j/2m)x_i x_j} \ , \tag{4}$$

where x_i is the attribute value of vertex i, A is the adjacency matrix, k denotes the degree, and $\delta_{ij} = 1$ if $x_i = x_j$ and 0 otherwise.

When nodal attributes are not available, or not of interest, vertex-level structural measures can be substituted. And, thus one of the more common kinds of assortativity studied is mixing by degree, which can be calculated from topological data alone.