1 Network basics

A *network* is a collection of vertices (or nodes or sites or actors) joined by edges (or links or bonds or ties). In mathematical jargon, networks are also called *graphs* and have been studied as mathematical objects for hundreds of years. Until the second half of the 20th century, most representations of empirical networks were small and largely confined to maps of social ties, constructed painstakingly by social scientists. Modern computers have now made it much easier to measure, store, draw and analyze the structure of extremely large networks. Arguably, the largest network currently studied is the World Wide Web, which contains tens of billions of nodes and likely trillions of links. ("Arguably" and "likely" because the WWW is so large that its structure is not known exactly.)

Because a network is logically equivalent to a graph, anything that can be represented as a set of discrete entities with pairwise[1] interactions can put in a network representation. For instance:

network	vertex	edge
Internet	computer	network protocol interaction
World Wide Web	web page	hyperlink
power grid	generating station or substation	transmission line
friendship network	person	friendship
metabolic network	metabolite	metabolic reaction
gene regulatory network	gene	regulatory effect
neural network	neuron	synapse
food web	species	predation or resource transfer

In some cases, the network representation is a close approximation of the underlying system's structure. In others, however, it's a stretch. For instance, in molecular signaling networks, some signals are conglomerations of several proteins, each of which can have its own independent signaling role. A network representation here would be a poor model because proteins can interact with other proteins either individually or in groups, and it's difficult to represent these different behaviors within a simple network.[2] In general, it's important to think carefully about how well a network representation captures the important underlying structure of a particular system, and how we might be misled if that representation is not very good.

1.1 Types of networks

Networks come in several flavors, largely based on the type of information the represent or the types of structures allowed/disallowed. Here, we will cover some of the most common types.

[1] Higher-order interactions can also be defined, and networks of these are called *hypergraphs*. Examples include collaboration networks like actors appearing in a film, scientists coauthoring a paper, etc.

[2] Such a network could be represented using a mixed hypergraph, in which some edges are defined pairwise, while others are hyperedges of different orders, defined as interactions among sets of nodes.

Simple graphs, multigraphs and self-loops.
A network is called *simple* if (i) there can be at most one edge (i, j) between any pair of vertices, i.e., edges are binary, (ii) there are no edges connecting a vertex to itself (a feature we call a *self-loop*), and (iii) a connection (i, j) implies a connection (j, i), i.e., edges are undirected. Figure 1a shows an example of a simple graph.

A *multigraph* relaxes the constraint on multiple connections (and generally also the constraint on self-loops). For instance, if vertices represent cities, and edges represent driving paths between a pair of cities, then a multigraph will be a reasonable representation because there can be several distinct such paths between a pair of cities. Similarly, in a network of neuron cells, two neurons can have multiple synapses and we might wish to represent each such connection as a distinct edge.

Weighted networks.
There are two types of weight information commonly used to annotate a network: edge weights and vertex attributes. A weight for an edge (i, j) is often denoted by a weight function $w(i, j)$ or by a scalar value w_{ij}. Edge weights are useful for representing attributes associated with the connection, e.g., interaction strength, frequency of interaction, capacity of interaction, etc., and are generally either a non-negative real number or natural number. Vertex attributes are simply values attached to the vertices, e.g., size, capacity, memory, etc. In social networks, common attributes are demographic values, e.g., age, sex, location, etc.

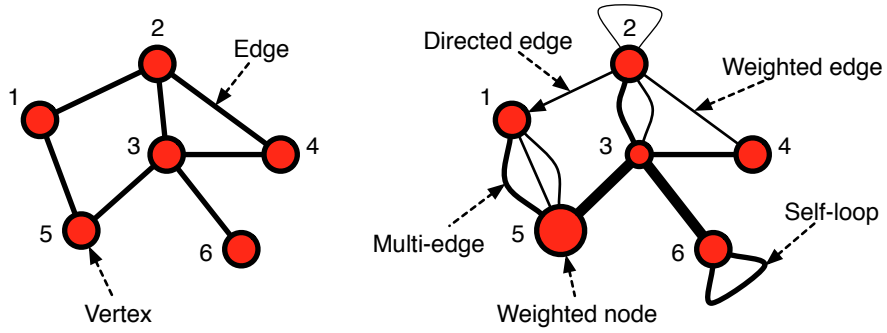


Figure 1: Examples of different types of edge and node structures. The left-hand network is an unweighted, undirected simple graph. The right-hand network is more exotic.

Directed networks.
A *directed network* allows asymmetric connections, i.e., connection (i, j) may exist independently of whether the connection (j, i) exists. We sometimes use the word "arc" to identify such a directed connection, rather than the more ambiguous term "edge." The World Wide Web is an example of a directed network, in which webpages are vertices, and hyperlinks are arcs or directed edges.

An *acyclic network* or graph is a special kind of directed network that contains no cycles, i.e., for all choices of i, j, if there exists a path $i \to \cdots \to j$ then there does not exist a path in the reverse direction $j \to \cdots \to i$. All trees are acyclic undirected networks. When we allow directionality in the edges, non-trees can be acyclic. For instance, a citation network, in which published papers are vertices and paper i connects to paper j if i cites j in its bibliography, is a kind of acyclic directed network (at least in theory; in practice, some cycles exist).

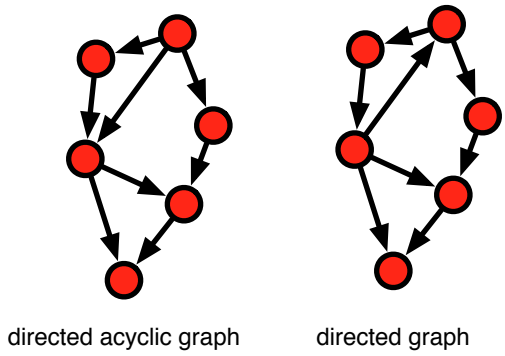


Figure 2: Examples of directed graphs.

Bipartite networks and one-mode projections.
If vertices can represent distinct classes of objects and only objects of different classes interact, i.e., edges cross between classes but not within classes, then we have a k-partite graph, where k is the number of classes. The simplest and most common form of such graph is the *bipartite* graph, with $k = 2$. One popular type of bipartite graph is the actor-film network, in which actors and films represent the two classes, and actors connect to the films in which they play a part.

Often, we wish to convert such a heterogeneous graph into a simple graph in which every vertex is of the same class, i.e., we want a *one-mode projection* of the bipartite graph. There are k one-mode projections for every k-partite graph. In a one-mode projection, two vertices are connected if they share a neighbor in the original graph. For instance, in the actor-film network, two actors would be connected if they ever acted in the same film together; or, two films would be connected if they have an actor in common.

One consequence of a one-mode projection is the construction of cliques, i.e., a subgraph of size ℓ in which every pair of nodes is connected. For instance, all actors in a particular film will be joined in a clique in the one-mode actor projection.

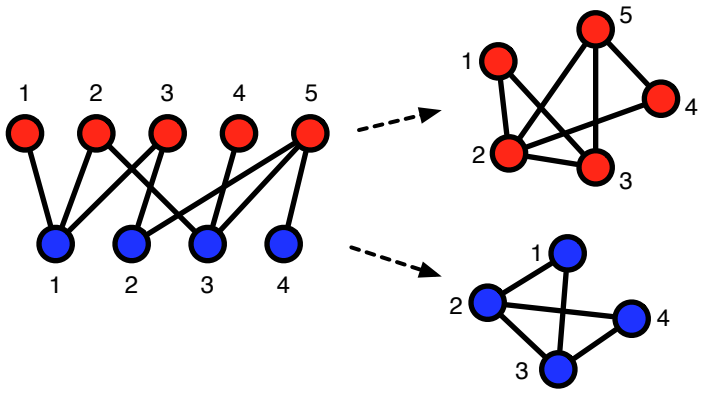


Figure 3: Example of a bipartite graph and its two one-mode projections.

Temporal, dynamic and evolving networks.
The networks described so far are static, meaning that the vertices and edges do not change. Graphs that change over time are an important class of networks. For example, citation networks are dynamic networks, because new vertices join the network continuously and each time a new vertex joins, it creates new edges representing citations to papers in its bibliography.

There are two types of temporal networks, although these are not as different as they might sound. In the first case, we convert some kind of underlying time-stamped interaction data into a sequence of network "snapshots" $A^{(t_1)}, A^{(t_2)}, \ldots$, where the sequence $t_1, t_2, \ldots$ represents a set of periods of time, e.g., hours in the day or days in a week. Within a particular snapshot, two vertices are connected if they ever interacted within the corresponding time period. An alternative representation of time-stamped interactions is to annotate each edge with the particular moment or span of time in which it occurred, e.g., (i, j, t_1, t_2) for an interaction starting at time t_1 and ending at time t_2.

Other types of networks.
There are, of course, many other types of networks. *Planar graphs* are those capable of being embedded in a 2d plane such that no edges cross. *Spatial networks* are empirical networks whose vertices have some position in space, commonly a position on the surface of the planet, e.g., road and city networks, airport transportation networks, oil and gas distribution networks, shipping networks, etc. These networks are often very close to, but not exactly, planar.

Hypergraphs are another form of network, in which edges denote the interaction of more than two vertices. *Multiplex networks* are a form of network in which multiple "layers," each of which may have a distinct edge set itself, represent different types of interactions between a common set of

vertices. Crucially, there may be complicated dynamics on each vertex that govern which layer some kind of interaction occurs on, so multiplex networks are not merely a special kind of graph in which edges are annotated by different colors or layer numbers.

There are, of course, even more types of networks to be found in the literature, but these represent the most common types.

1.2 Representations of networks

There are three main ways to represent a network.

The first is an *adjacency matrix*, often denoted A, where

$$A_{ij} = \begin{cases} w_{ij} & \text{if } i \text{ and } j \text{ are connected} \\ 0 & \text{otherwise} \end{cases}$$

If A represents an *unweighted network*, then $w_{ij} = 1$ for all i, j.

An adjacency matrix representation of Fig. 1a is

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} .$$

Notice that the diagonal is all zeros and the non-zero entries are 0 or 1. This indicates that the corresponding network is a simple network. Also notice that the matrix is symmetric across the diagonal. Undirected networks have this structure because the upper triangle represents connections i, j while the lower triangle represents j, i.

Adjacency matrices are commonly used in mathematical expressions, e.g., when describing what a network algorithm does, but they are also sometimes used in algorithm's actual operation. The disadvantage of doing so, however, is a matrix always takes $O(n^2)$ memory to store, where n is the number of nodes in the network. Most empirical networks are *sparse*, meaning that the number of non-zero entries in the adjacency matrix is $O(n)$, and an adjacency matrix stores this small number of non-zero elements inefficiently. (Sparse-matrix data structures can circumvent this problem; these are essentially equivalent to our next representation.)

The second is an *adjacency list*, which stores only the non-zero elements of the adjacency matrix in a full list of all vertices. From a data-structures point of view, an adjacency list is like a hash

table, in which each of the table entries corresponds to a vertex, and the elements we store in that table entry are a list of vertices reachable directly from that vertex, i.e., its out-going edges. Here is the adjacency list representation of Fig. 1a:

1	2	5		
2	3	1	4	
3	2	5	4	6
4	2	3		
5	1	3		
6	3			

Because our network is undirected, every edge (i, j) appears twice in the adjacency list, once as j in i's list, and once as i in j's list. The adjacencies for a given vertex can be stored as a linked list, or in a more efficient data structure, like a self-balancing binary tree (e.g., a red-black or a splay tree). This form of representation is essentially equivalent to a "sparse matrix," and is thus a common internal format for network data structures.

The third representation is an *edge list*, which stores only the non-zero elements of the adjacency matrix. Here's the undirected edge list representation of Fig. 1a:

$$\{(1,2),(1,5),(2,3),(2,4),(3,5),(3,6)\},$$

where it is assumed that all edges have unit weight and that the presence of an edge (i, j) implies an undirected tie between i, j.

This kind of structure is more typically used to store a network in a file on disk, possibly with additional information representing annotations for weighted edges [e.g., (i, j, w_{ij})], node annotations (like weights or attributes; usually stored in a file header), etc.

2 The goals of network analysis and modeling

Note: this may be the most important section of the entire class.
There are two principal modes of network analysis and modeling: exploratory and hypothesis-driven. Loosely speaking, these two modes correspond to the situation in which one does not have a formal idea of cause and effect (exploratory, or unsupervised analysis) versus the situation in which one has a clear idea of mechanism (hypothesis driven, or supervised analysis).

2.1 Exploratory analysis

Much of network science is, in fact, exploratory, where the goal is to identify the structural patterns and correlations within a network, or across a set of networks, that are interesting.[3] In this mode, the network and associated auxiliary information (edge weights, vertex attributes, etc., if any) are analyzed in an unsupervised fashion. That is, we would like to know what *shape* the network takes, and what *unusual* patterns it exhibits, if any. Connecting these observations with hypotheses or mechanisms is then done in a *post hoc* inductive fashion. Even if our ultimate interest is in understanding what *degrees of freedom* underlie and explain these patterns, the first step is always to identify the patterns that need to be explained. As such, we often employ random-graph models in exploratory analysis as a benchmark against which to measure unusualness. We also often use network models to find effective *coarse grainings*, providing a more compressed description of a network's overall structure.

Many exploratory network analysis tasks can be reduced to the following kind of model. We imagine that an edge (i, j) exists with probability

$$p_{ij} = \Pr(i \to j \,|\, x_i, x_j, \gamma_i, \gamma_j, \theta_{ij}) \ , \tag{1}$$

where each x represent a set of vertex-level observed attributes, each γ represents a set of vertex-level latent (unobserved) attributes, and θ is some latent attributes of the pairing of i and j. For instance, consider a pair of individuals i and j on Facebook. Each person's x contains the attributes they disclose about themselves on Facebook (age, sex, location, etc.). Their γ represents all attributes not disclosed on Facebook (including attributes that Facebook does not ask about), and θ represents latent attributes of the pair (family relationship, work relationship, etc.).

Facebook has many reasons for wanting to know p_{ij}, but they may not care about why it takes that value or how that value changes over time. But, if they knew a functional representation of Eq. (1) for their network, they could do many powerful things, including inferring missing attributes and predicting missing links. The goal of exploratory analysis is, to a large extent, estimating a low-dimensional form for Eq. (1), i.e., a form that depends on many fewer variables than the number of vertices or edges in the network. The more compact the form, the simpler the shape of the network.

Good exploratory analysis requires *creativity* (to imagine what shape the network might have, and why), *mathematical intuition* (to know what kinds of shapes are possible, and even plausible), *algorithmic tools* (to know how to see that shape and to extract it from the data), and *statistical rigor* (to show that the shape is real and not a clever illusion). Good exploratory analysis finds new and interesting patterns within empirical data, and generates questions to be addressed through a more hypothesis-driven approach.

[3] It is an interesting question as to why so much of network science is in this mode, rather than the other. Notably, there are many fields in which the balance is reversed.

2.2 Hypothesis-driven analysis

Applications of network science (e.g., in social or biological or other systems) are often more hypothesis driven, where the goal is to understand what role the network's structure plays in a particular question of cause and effect. In this mode, the network is analyzed in a more supervised fashion, in which the pattern or behavior of interest is identified beforehand, and the question is deciding how the network relates to, constrains or drives that pattern.

The details of hypothesis-driven analysis are often very domain specific, in the sense that the system is grounded in a particular domain whose underlying dynamics or rules are crucial for investigating the hypothesis. This is not always true, of course, and there are many hypothesis-driven analyses about networks themselves. In these cases, the hypothesis is often a "structural" one, in which we seek to understand some particular connectivity pattern.

For instance, suppose Facebook has already identified some particular pattern in a network, e.g., that vertices with certain types of attributes tend to be connected (a pattern called *homophily* or *assortativity*), which we might express as a particular form for $\Pr(i \to j \,|\, x_i, x_j, \gamma_i, \gamma_j, \theta_{ij})$. There are at least two reasons[4] why we may see such a correlation across edges: edges may form for some reason independent of the attributes' values and subsequently the values align with the pattern as a result of the edge existing, or, the attributes' values are aligned for some reason independent of the existence or not of an edge and subsequently the probability of an edge existing (or persisting) increases as a result of the attributes' values. These are two very difference mechanisms—in one, edges drive attribute patterns, while in the other, attribute patterns drive edges. Thus, we have two hypotheses that both produce the same observed pattern, and our job is to decide the degree to which each is correct or incorrect.

Much of hypothesis-driven analysis has this kind of flavor, in which we attempt to *explain* why a certain pattern exists, by articulating a *mechanism* by which the observed pattern is caused. A "structural" mechanism is one that depends only upon processes or features related to the network itself. Ideally, mechanisms are formalized mathematically, so that one can derive the effect directly from the cause, and then compare the prediction with data.

Good hypothesis-driven analysis requires *creativity* (to imagine how a network's shape could lead to the behavior of interest), *mathematical intuition and rigor* (to know what kinds of mechanisms are possible and to show their consequences), *numerical tools* (to simulate the mechanism and analyze its results), and *statistical rigor* (to show that the hypothesis is supported or not). Good hypothesis-driven analysis identifies and demonstrates believable causes for real effects, and shows that these explanations are better than simple alternatives.

[4] A third reason is that some external process drives everything, e.g., non-stationary cultural dynamics.

3 Simple structural measures of networks

Regardless of which mode we operate in, simple measures of network structure are the basis on which nearly all network analysis and modeling rests. In this section, we will go over the basic language of networks and build some intuition for thinking about their shape.

Given network G, there are many quantities we could potentially measure as a way to characterize its structure. For instance, we could measure localized structures and then compute the average value or its distribution over the entire graph. Alternatively, we could define more global measures of structure, and anything in between. Here, we will cover some of both.

3.1 Degrees

The most fundamental of all network measures is the *degree* of its vertices. By convention, we let k_i denote the degree of vertex i, which is a count the number of connections terminating (equivalently: originating) at that node. Using the adjacency matrix, the degree of vertex i is defined as

$$k_i = \sum_{j=1}^{n} A_{ij} \ , \tag{2}$$

which is equivalent to summing the ith column (or row) of the adjacency matrix A. If A represents a weighted network, this sum is called the node *strength*; the term "degree" is reserved for unweighted counts.

Every edge in an undirected network contributes twice to some degree (once for each endpoint or "stub"), and so the sum of all degrees in a network must be equal to twice the total number of edges in a network m:

$$m = \frac{1}{2}\sum_{i=1}^{n} k_i = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij} = \sum_{i=1}^{n}\sum_{j=i}^{n} A_{ij} \ . \tag{3}$$

And, the mean degree of a node $\langle k \rangle$ in the network is

$$\langle k \rangle = \frac{1}{n}\sum_{i=1}^{n} k_i = \frac{2m}{n} \ . \tag{4}$$

If we divide the mean degree by its maximum value

$$\rho = \frac{2m}{\binom{n}{2}} = \frac{2m}{n(n-1)} = \frac{\langle k \rangle}{n-1} \ , \tag{5}$$

we have a quantity that is sometimes called the network's *connectance* or *density*.[5]

One of the more common uses of the degree is to tabulate a network's *degree distribution*, denoted $\Pr(k)$, which gives the probability that a vertex selected uniformly at random will have k neighbors. (This is distinct but related to the *degree sequence*, which is simply a list of the degrees of every node in a network.) The degree distribution for Fig. 1a is

k	$\Pr(k)$
1	1/6
2	3/6
3	1/6
4	1/6

where $\Pr(k) = 0$ for all other values of k.

In studies of empirical networks, the degree distribution is often used as a clue to determine what kinds of generative models to consider as explanations of the observed structural patterns. Generally, empirical social, biological and technological networks all exhibit right-skewed degree distributions, with a few nodes having very large degrees, many nodes having intermediate degrees, and a large number having small degrees.

3.2 Geodesic 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 *geodesic* or *shortest path* is the shortest of all possible paths between two vertices, and these serve as the basis for a number of important measures of network structure.

The first and simplest such measure is the network *diameter*, which is the length of the longest of these geodesics and is meant to evoke the notion of a volume in a metric space.[6] Like many measures based on paths, measuring the diameter is done by running an algorithm that solves either the *All Pairs Shortest Paths* (APSP) problem or the *Single Source Shortest Paths* (SSSP) problem (n times, in the case of the diameter).

If the network is unweighted and undirected, a simple Breadth-First Search (BFS) tree will solve the SSSP problem,[7] providing us with the length of the path from a given source vertex i to all other

[5] Sometimes, connectance is defined as c/n, which is asymptotically equivalent to $c/(n-1)$.

[6] *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.

[7] Or, run Bellman-Ford or Dijkstra's algorithm. All SSSP algorithms return a $n \times 1$ vector of distances from a single input vertex i to each of the other n vertices. By repeating this procedure for each vertex, we may construct

vertices reachable from i. For directed or weighted networks, we would instead use an algorithm like Floyd-Warshall or Johnson's algorithm.[8] Note that *reachability* is a crucial detail: if some vertex j is unreachable from i, then there is no geodesic path between i and j and the distance between them is either infinite or undefined. The diameter of a network is thus the diameter of its largest component (see Section 3.3 below).

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 very slowly as a function of network size, e.g., $O(\log n)$, a network is said to exhibit the "small world" property.

The basic idea of "small world" networks comes from a seminal study in social networks by the American sociologist Stanley Milgram (1933–1984).[9] Milgram mailed 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 doing so, they should also write their name on a roster to 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 toy model, now called the "small world model," in which vertices are arranged on a 1-dimensional circular lattice (a "ring" network) and connected with their k nearest neighbors. Each edge is then rewired, with probability p, to connect a uniformly random pair of vertices. At $p = 0$, the network is fully ordered, where the density of local connections is largest and the diameter is $O(n)$. At $p = 1$, the network is fully disordered, local connections are absent and the diameter is $O(\log n)$. See Figure 2 below.

The interesting behavior emerges as we dial p between these two extremes: when only a small number of edges have been randomly rewired (small p), the diameter of the network collapses from $O(n)$ to $O(\log n)$ while the local structure is still largely preserved. That is, a highly-ordered "big world" can be transformed, by rewiring only a small number of connections, into a still mostly ordered small world, in which geodesic paths traverse a vanishing fraction of the network. This behavior reminded them of some properties of social networks, which have small diameter (as evidenced by

the $n \times n$ pairwise distance matrix, one column at a time. BFS on an unweighted, undirected network takes $O(n+m)$ time, so the total running time is $O(n^2 + mn)$. For sparse graphs, this is a relatively fast $O(n^2)$ but becomes a slow $O(n^3)$ for dense graphs.

[8] Floyd-Warshall takes $O(n^3)$ time in the worst case, which doesn't scale up to large networks ($n > 10^5$ or so)

[9] 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.

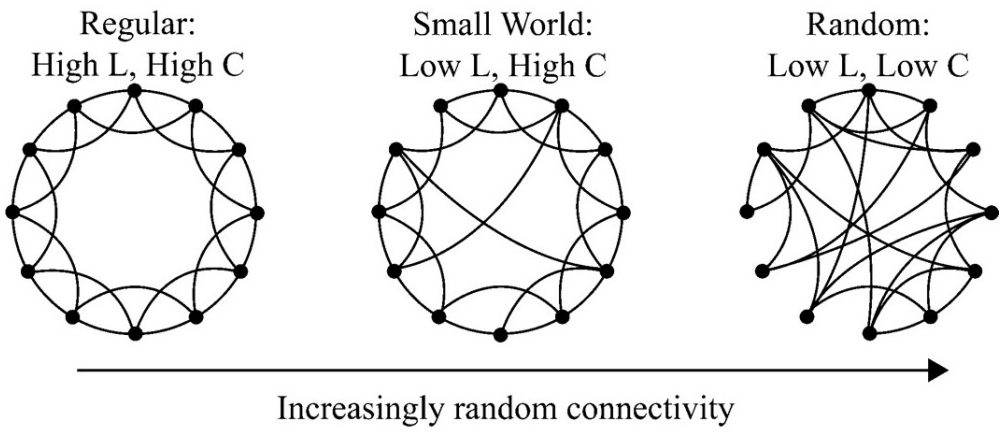


Figure 4: Watts and Strogatz's schematic illustrating the small worlds model.

Milgram's study) but also many triangles.[10]

The small-world result is interesting for several reasons. First, it exemplifies an early effort to understand, using a toy model, how social networks can have both substantial local structure (mainly triangles) but also a small diameter. Second, it shows that some measures of network structure can be extremely sensitive to uncertainty in the network structure. Suppose we obtained data generated by our toy model with an intermediate value of p, such that only a small number of edges had been rewired. If we observe each real edge with probability p, then it is possible that we will miss several of the long-range connections that cross the ring. The result would be a substantial overestimate of the diameter of the network, potentially confusing $O(n)$ with $O(\log n)$. Fortunately, not all measures of network structure are this sensitive to sampling errors, but it is well worth our time to consider the impact of the data generation and data observation processes when analyzing and modeling networks.

3.3 Components

If every pair of vertices is connected by some path, then the network is *connected*. If there is some pair of vertices between which no path exists, the network is said to be *disconnected*, i.e., it is

[10]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.

composed of more than one *component*.

In an undirected graph, the set of vertices reachable from one vertex is called a *component* (and, for every vertex j reachable from i, i is also reachable from j). In a directed graph, reachability in one direction does not imply reachability in the other, and the notion of "connected" becomes more nuanced. A group of nodes that is pairwise reachable only if we ignore the direction of the edges is a *weakly connected component*, while a group of notes that is pairwise reachable if we obey the directions is a *strongly connected component*. Similarly, an *out component* is the set of vertices that can be reached from i, while an *in component* is the set of vertices that can reach i.

Many empirical networks are composed of multiple components, and among them, there is always a *largest component*, which is usually the component of greatest interest. In mathematical models, the *giant component* is also the largest component, but we add an additional property, which is to say that the size of the giant component must be $O(n)$.[11] We will revisit the notion of a giant component later in the semester, when we study models of random graphs.

Counting components.
To count and measure the size of the components within a network, we use any standard SSSP or APSP algorithm. For undirected networks, a breadth-first or depth-first search forest suffices, while for weighed or directed networks, Dijkstra's algorithm works well.

While the algorithm runs, we need only label all vertices that are pairwise reachable with the same vertex label, in an auxiliary array. When the algorithm terminates, we may make a single pass through the array to count the number of unique labels (the number of components) and count the number of times each label occurs (the sizes of each component). In a weighted graph, identifying strongly connected components is most easily done via a depth-first search forest, in $O(n + m)$ time.

3.4 Reciprocity (directed networks)

In directed networks, not all edges are bidirectional, and the fraction of those edges that are bidirectional can tell us interesting things about the network, depending on what kind of network it is. For instance, the figures at the top of the next page show one reciprocated and one unreciprocated edge.

In social networks, bidirectional or reciprocated edges can indicate whether a friendship is perceived as being equal or whether one party views it as stronger than the other. That is, reciprocated edges can tell us something about social status. In transportation networks, they represent reachability,

[11] The term "giant component" is meaningless in most empirical situations since it is only defined asymptotically. For empirical networks with multiple components, we instead focus on the largest component and reserve the term "giant component" for mathematical models.

while in communication networks, they may represent influence.[12] Reciprocity can be calculated in any directed network, e.g., in food webs, predation and parasitism are directed relationships, as is genetic regulation in a gene-regulatory network. However, the meaning of a reciprocated link depends on the context and the underlying processes in that domain. For example, a reciprocated link in a food web means something different than a reciprocated link in a social network.

The reciprocity of a network is given by the fraction of reciprocated links. To compute this value, we simply count the relative frequency of cliques of size 2 in a directed network. When edges have unit weight, a network's reciprocity is defined as

$$r = \frac{1}{m} \sum_{ij} A_{ij} A_{ji} \ . \tag{6}$$

In this way, if both the forward and backward direction of a particular edge appear in the network, the reciprocity score is incremented twice. The normalization factor counts all edges that could be reciprocated.

Reciprocity may also be defined as a vertex-level measure, in which we count only the fraction of 2-cliques attached to some vertex i:

$$r_i = \frac{1}{k_i} \sum_{j} A_{ij} A_{ji} \ , \tag{7}$$

which may be a useful way of estimating a vertex-level covariate for additional analysis (e.g., to compare with the vertex degree, or centrality score). This measure is sometimes called the *local reciprocity*.

In empirical social networks, a value of $r \simeq 0.25$ is not unusual. While this may like a surprisingly small value, this value is observed even in very large networks where the probability of two directed edges forming a bidirectional loop is roughly $O(1/n)$. In some cases, a small value is found in part because the survey technique limits the number of responses, or because different people interpret

[12] But, you should be wary of anyone claiming to either measure or test for social influence. In general, most studies of "influence" are actually studies of correlation, and as with homophily, correlation does not imply causation. Keep your wits and skepticism about you.

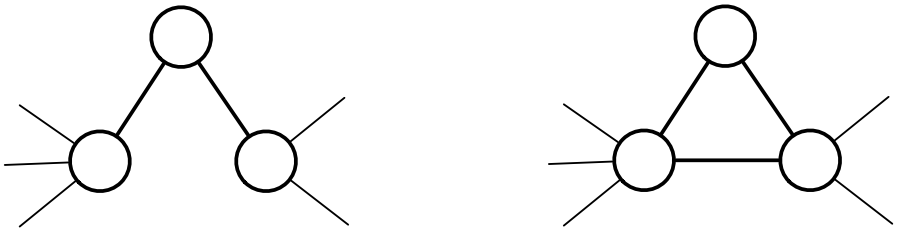
the word "friend" to mean different things. In other cases, a small value may reflect the presence of status-driven relationships. In undirected networks, reciprocity is always 1.

3.5 Clustering coefficient

Structurally, reciprocity measures the fraction of 2-cliques that appear in a directed network. Another commonly used measure counts the fraction of 3-cliques or triangle density in an undirected network (directed versions also exist, but their calculation is slightly more tricky, as there are many more ways three vertices could be connected in a directed network), and is called the *clustering coefficient*.[13] This measure is defined mathematically as

$$C = \frac{\text{(number of closed paths of length 2)}}{\text{(number of paths of length 2)}} \quad . \tag{8}$$

A path of length two has the natural definition, which is a sequence of vertices i, j, k for which the edges (i, j) and (j, k) are in the network. A "closed path" of length two is simply a path of length two plus the edge (k, i).



Thus, C is a number between 0 and 1, and measures the density of triangles in the network. It takes the value $C = 1$ only for a fully connected component, i.e., a clique of size n. The opposite extreme, a value of $C = 0$ can occur on a number of different networks, the most obvious being any bipartite network (including trees).

In the above formulation, as in our discussion of paths for betweenness, we count distinct orderings of the vertices as being different. If we collapse these counts, we may simplify Eq. (8) in the case of undirected graphs by counting only closed and "open" triangles. Let us define a "connected triple" or "open triad" as any trio of nodes i, j, k in which at least two pairs are connected. A "closed triad" is then any such trio in which we have added the final, missing connection (hence the word

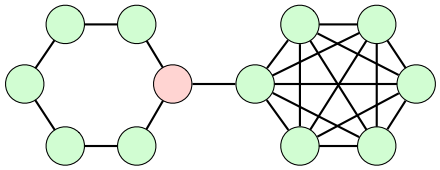
[13] This name is not a particularly good one, as the term "clustering" is used in several other contexts in the study of networks and vector data sets. Readers should be wary of these alternative usages when reading the literature.

"closed"). The clustering coefficient is then

$$C = \frac{\text{(number of triangles)} \times 3}{\text{(number of connected triples)}} \ , \tag{9}$$

where the factor of 3 comes from the symmetry of the triangle.

To illustrate this measure, consider again our simple cycle-and-clique network. The clique has 6



vertices and thus both 20 triangles and 60 connected triples. The cycle contains 6 connected triples and no triangles. Finally, there are 2 connected triples starting from the cycle and 5 connected triples starting from the clique that use the edge joining the cycle to the clique. Thus, the clustering coefficient is $C = 60/73 = 0.82$. This value is relatively large for social networks, which generally have clustering coefficients of $C \simeq 0.20$ or so. A non-trivial clustering coefficient is generally a distinguishing feature of social networks, with most biological and technological networks containing far fewer triangles, and which exhibit clustering coefficients close to 0.

The clustering coefficient can also be defined as a vertex-level measure:[14]

$$C_i = \frac{\text{(number of pairs of neighbors of } i \text{ that are connected)}}{\text{(number of pairs of neighbors of } i)} \ , \tag{10}$$

which is the natural definition of triangle density where we require that vertex i be the middle vertex of every connected triple. In our cycle-plus-clique example above, the highlighted vertex has a local clustering coefficient of $C_i = 0$ because it participates in no triangles. Its immediate neighbor in the clique is a more interesting case, with $C_i = 10/15 = 0.67$.

4 The shape of (some) networks

With these tools in hand, we can now begin to analyze the shape of networks. The big table on the next page lists a number of different empirical networks, drawn from social, informational,

[14] Two ideas from the sociological literature that are closely related to the local clustering coefficient are *structural holes* and *redundancy*, which we won't cover here.

	network	type	n	m	z	ℓ
social	film actors	undirected	449 913	25 516 482	113.43	3.48
	company directors	undirected	7 673	55 392	14.44	4.60
	math coauthorship	undirected	253 339	496 489	3.92	7.57
	physics coauthorship	undirected	52 909	245 300	9.27	6.19
	biology coauthorship	undirected	1 520 251	11 803 064	15.53	4.92
	telephone call graph	undirected	47 000 000	80 000 000	3.16	
	email messages	directed	59 912	86 300	1.44	4.95
	email address books	directed	16 881	57 029	3.38	5.22
	student relationships	undirected	573	477	1.66	16.01
	sexual contacts	undirected	2 810			
information	WWW `nd.edu`	directed	269 504	1 497 135	5.55	11.27
	WWW Altavista	directed	203 549 046	2 130 000 000	10.46	16.18
	citation network	directed	783 339	6 716 198	8.57	
	Roget's Thesaurus	directed	1 022	5 103	4.99	4.87
	word co-occurrence	undirected	460 902	17 000 000	70.13	
technological	Internet	undirected	10 697	31 992	5.98	3.31
	power grid	undirected	4 941	6 594	2.67	18.99
	train routes	undirected	587	19 603	66.79	2.16
	software packages	directed	1 439	1 723	1.20	2.42
	software classes	directed	1 377	2 213	1.61	1.51
	electronic circuits	undirected	24 097	53 248	4.34	11.05
	peer-to-peer network	undirected	880	1 296	1.47	4.28
biological	metabolic network	undirected	765	3 686	9.64	2.56
	protein interactions	undirected	2 115	2 240	2.12	6.80
	marine food web	directed	135	598	4.43	2.05
	freshwater food web	directed	92	997	10.84	1.90
	neural network	directed	307	2 359	7.68	3.97

Figure 5: Basic statistics for a number of published networks. The properties are: type of graph, directed or undirected, total number of vertices n, total number of edges m, mean degree z and mean geodesic distance ℓ. Table adapted from M.E.J. Newman, "The structure and function of complex networks." *SIAM Review* **45**, 167–256 (2003).

technological, and biological systems, and their values along a few of these measures. It is worth noting that the field has progressed greatly since this table was published in 2003, but there are relatively few other or newer "big tables" showing a specific set of network statistics for a large and diverse set of real-world networks. As such, it remains an instructive list with many lessons to teach.

What should be immediately clear from this table is that network sizes vary enormously. Even with the social network category, the smallest is $n = 573$ (student relationships) while the largest is $n = 4.7 \times 10^7$ (telephone call graph), more than 80,000 times larger. And yet, despite this enormous

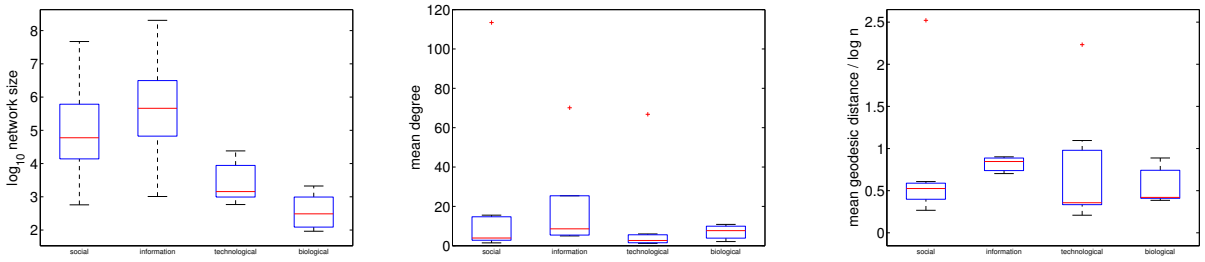


Figure 6: Boxplots showing the within-class distributions of network size ($\log n$), mean degree, and mean geodesic distance (normalized by network size).

difference in size, the mean degree of these two networks differs by less than a factor of 2. Similarly, the largest mean geodesic distance in a social network (16.01; student relationships) is nearly as large as the mean geodesic distance for the Altavista WWW network (16.18).

Figure 4 shows the within-class distributions of these simple measures (as box plots). These reveal that, indeed, social and information networks tend to be larger than technological networks, and that biological networks tend to be the smallest. The mean degrees are all similar, being generally less than 20, although social and information networks have the broadest distributions here. Interestingly, the mean geodesic distance (normalized by network size) shows the opposite pattern, with technological and biological networks showing the greatest variance. These patterns suggest that there are some broad patterns across these types of networks.

Another simple way to investigate the hypothesis that there is little overall variation in these measures across network class is to plot the mean degree and the mean geodesic distance versus network size. Figure 4 (next page) shows the results, with each class of network (social, information, technological, biological) given different colors and shapes. What is notable about these scatter plots is the lack of any clear pattern, either across the classes or within each class. That is, there appears to be little or no systematic correlation between mean degree and network size or between mean geodesic distance and network size.

5 At home

1. Peruse Chapters 1–5 (pages 15–104) in *Networks*
2. Read Chapter 6.1–6.12 (pages 109–149) in *Networks*

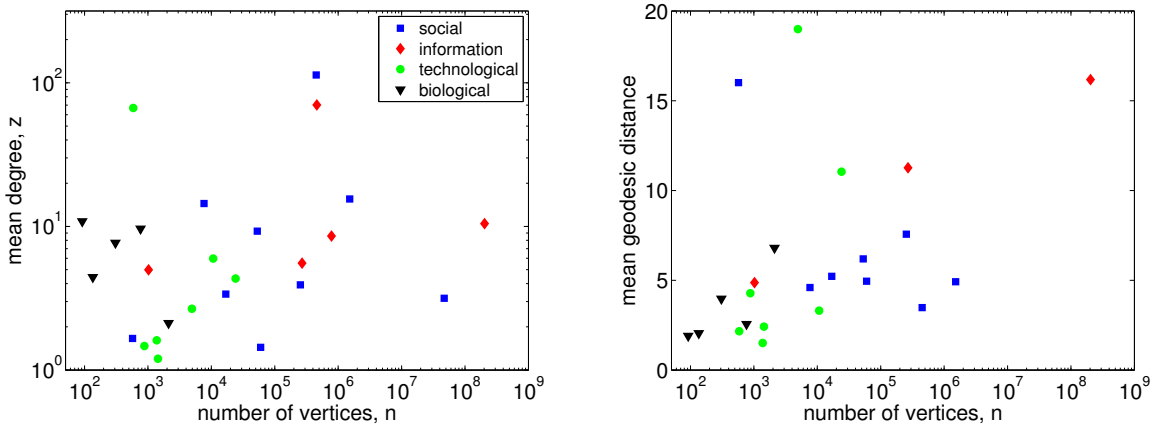


Figure 7: Scatter plots for mean degree and mean geodesic distance versus number of vertices.