Inference, Models and Simulation for Complex Systems CSCI 7000-001
Lecture 12 11 October 2011
Prof. Aaron Clauset

1 Networks

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] 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.

1.1 Types of networks

There are many types of networks, for example, multigraphs, simple graphs, hypergraphs, networks with self-loops, bipartite networks, acyclic networks, weighted networks, etc. We'll go through most of these and point out their differences. Figure 1 below shows some of them schematically.

A network in which a pair of nodes i, j can have multiple, independent connections is called a *multigraph*, e.g., two cities can be joined by multiple roads and two neurons can interact through multiple synapses. Sometimes, it is more convenient to collapse the different multi-edges between two nodes into a single edge annotated by a *weight* w_{ij} equal to the count of those multi-edges. Weights can also be used to represent the strength or capacity or frequency of the interaction, and are generally real-valued. A network with weighted edges is called a *weighted network*. Nodes can also be annotated, e.g., with a vector of personal attributes (as on Facebook).

If nodes are allowed to connect to themselves, such an edge is called a *self-loop*. A network with no self-loops and no multi-edges is called a *simple network* or graph. The most common type of empirical network data is a simple network, since tabulating it only requires knowing which vertices interact and not anything about the types of interactions or nodes. A *directed network* is one in which connections can be asymmetric, i.e., node i can connect to node j without the reverse being true. The World Wide Web is an example of a directed network. An *acyclic network* 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 $j \to \cdots \to i$. A tree is a kind of acyclic undirected network, and most phylogenies are examples of acyclic directed networks. Citation networks should be examples of acyclic directed networks, but are often not in practice. Citation networks are also examples of dynamic or *temporal networks*, where the set of edges changes over time. *Spatial networks* have vertices that are embedded in some kind of metric space. Networks can also be simultaneously spatial, temporal, weighted, etc.

Sometimes, we wish to represent the interactions between different types of things within a network structure, e.g., actors and the films they're in, scientists and the papers the coauthor, etc. These are called k-partite networks, where nodes of type μ only connect to nodes of type $\nu \neq \mu$. When $k = 2$, as in actors and films, they're called bipartite networks. We can always convert a k-partite network into a simple graph, where every node is of the same type, by computing the *one-mode projection*. In this construction, all the nodes of a single type are retained and pairs of these i, j are connected in the projection if and only if they had a common neighbor in the original network. For example, we link actors together if they appeared in the same film; in the original network, each actor connected to a vertex representing the movie they appeared in together. 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 the actors in a given movie will be joined in a clique in the one-mode actor projection.

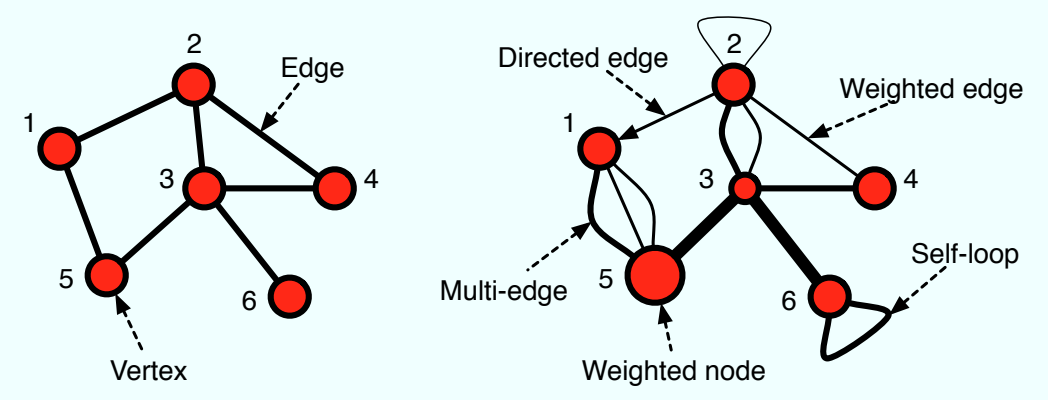

Figure 1: Examples of different types of edge and node structures. The left-hand network is an unweighted, undirected simple graph. The right-hand network is more exotic.

1.2 Representations of networks

There are two 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.

Adjacency matrices are most typically used in mathematical expressions, e.g., when describing what a network algorithm does, but it's also sometimes used explicitly in network algorithms. The disadvantage of doing so, however, is that they take $O(n^2)$ 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 a simple adjacency matrix representation is a very inefficient approach to storage. Sparse-matrix data structures can circumvent this problem and are often used in practice.

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.

The second 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 Structural measures of networks

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. We'll briefly cover some of the conventional measures of structure, starting with the more local measures.

2.1 Degrees

The *degree* of a node k_i is simply a count of the number of connections terminating (equivalently: originating) at that node. Using the adjacency matrix, the degree of vertex i is defined as

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

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:

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

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

$$\langle k \rangle = \frac{1}{n}\sum_{i=1}^{n} k_i = \frac{2m}{n} \ . \tag{3}$$

If we divide the mean degree by its maximum value

$$\rho = \frac{2m}{\binom{n}{2}} = \frac{2m}{n(n-1)} = \frac{\langle k \rangle}{n-1} \ , \tag{4}$$

we have a quantity that is sometimes called the network's *connectance* or *density*.[3]

One of the most common uses of the degree measure is in tabulating the *degree distribution* for a network $\Pr(k)$, which gives the distribution of a vertex selected uniformly at random. (This is distinct but related to the *degree sequence*, which is simply a list of the degrees of every node in a graph.) 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]Sometimes, connectance is defined as c/n, which is asymptotically equivalent to $c/(n-1)$.

2.2 Reciprocity, Transitivity and Similarity

In sociology, the aspects of network structure that are primarily emphasized are dyadic and triadic—i.e., pairs or trios of nodes and the connections between them—and more complex ideas like *structural holes*, *structural equivalence* and *social balance* are primarily defined in terms of them.

The simplest dyadic measure is *reciprocity*, which counts the frequency of loops of length 2 in a directed network. As an average over the entire network, when edges have unit weight, reciprocity is defined mathematically as

$$r = \frac{1}{m} \sum_{ij} A_{ij} A_{ji} \ , \tag{5}$$

but can also be defined as a vertex-level measure

$$r_i = \frac{1}{k_i} \sum_{j} A_{ij} A_{ji} \ . \tag{6}$$

Many online social networks, like Facebook, only allow undirected ties between people—if i lists j as a friend then j must also list i as a friend—so reciprocity is automatic and $r = 1$. Others, like Twitter, allow directional ties and thus more closely resemble the kind of data derived from social survey methods, where a researcher asks each person to name their friends. If i names j, but not vice versa, then the friendship is not reciprocated. Some studies of the WWW and email networks report high degrees of reciprocity, with $r \simeq 0.57$ and $r \simeq 0.23$ respectively. Notably, the idea of reciprocity can be applied to any directed network, for example, in food webs, predation and parasitism are directed relationships, as is genetic regulation in a gene-regulatory network. How reciprocity is interpreted in these contexts, however, varies by domain.

Structurally, reciprocity measures the fraction of 2-cliques that appear in a directed network. A more general measure would count the fraction of ℓ-cliques in the network, and this measure is called *transitivity*. The simplest measure of transitivity sets $\ell = 3$ and thus measures the density of triangles in the network. It can be applied to either directed or undirected networks.[4] A global measure of transitivity in a network is the "clustering coefficient",[5] which can be defined in several ways. One of the simplest is

$$C = \frac{(\text{number of triangles}) \times 3}{(\text{number of connected triples})} \ , \tag{7}$$

[4] In the sociological literature, the motivation for measuring transitivity comes from the importance of cliques, which represent groups of tightly interconnected friendships. Structurally, a 3-clique is also a loop of length 3, and a different generalization of this measure counts the frequency of loops of longer lengths.

[5] This name is not a particularly good one, as the term "clustering" is used in several other contexts in the study of networks.

where a "connected triple" is any trio of nodes i, j, k connected like (i, j) and (j, k). Thus, a connected triple is an "open" triangle, and the clustering coefficient measures the fraction of these open triangles that are closed. The extra factor of 3 appears because each closed triangle is composed of three open triangles, one centered on each of the three corners. The clustering coefficient can also be defined as a vertex-level measure:[6]

$$C_i = \frac{\text{(number of pairs of neighbors of } i \text{ that are connected)}}{\text{(number of pairs of neighbors of } i\text{)}} \ . \tag{8}$$

Some studies of networks, both empirical and model-based, have shown a correlation between the vertex-level clustering coefficient and other vertex-level measures, like the degree. For instance, high-degree vertices often exhibit lower local clustering coefficients.

Finally, given two vertices i and j, we may wish to measure how similar they are. Naturally, similarity can be quantified in many different ways. Two basic approaches from the sociological literature are called *structural equivalence* and *regular equivalence*.[7] Two vertices are structurally equivalent if they share many (or all) of the same neighbors. This kind of similarity is often quantified using the Jaccard coefficient (or some variation on it):

$$J_{ij} = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|} \ , \tag{9}$$

where $N(i)$ is the set of vertices with connections to vertex i.

Regular equivalence is more subtle: two vertices are regularly equivalent if their neighbors are similar, rather than themselves. This tries to get at the notion of the structural *role* of a vertex and ranks to vertices as similar if their structural position or role in the network is similar. This idea generalizes that of structural equivalence by making the definition recursive, and mathematically, it's defined as

$$\sigma_{ij} = \alpha \sum_{k\ell} A_{ik} A_{j\ell} \sigma_{k\ell} + \delta_{ij} \tag{10}$$

$$\sigma = (\mathbf{I} - \alpha \mathbf{A})^{-\mathbf{1}} \tag{11}$$

where Eq. (11) is the matrix version of Eq. (10), and is a kind of eigenvector calculation. Basically, σ represents a weighted sum of all paths between vertices i and j, where longer paths are given smaller weights. Vertices with similar weighted sums are more regularly equivalent. (Note: there are variations on this definition of regular equivalence, e.g., ones that normalize by the degree.)

[6] 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.

[7] There are a number of other similarity measures in the literature, for example, cosine similarity, a simple Pearson correlation coefficient or even the topological distance, and it's easy to come up with your own. The difficulty is showing that any particular similarity measure is actually useful.