CHAPTER 7

MEASURES AND METRICS

An introduction to some standard measures and metrics for quantifying network structure, many of which were introduced first in the study of social networks, although they are now in wide use in many other areas

IF WE KNOW the structure of a network we can calculate from it a variety of useful quantities or measures that capture particular features of the network topology. In this chapter we look at some of these measures. Many of the most important ideas in this area come from the social sciences, from the discipline of *social network analysis*, which was developed to aid our understanding of social network data such as those described in Chapter 3, and much of the language used to describe these ideas reflects their sociological origin. Nonetheless, the methods described are now widely used in areas outside the social sciences, including computer science, physics, and biology, and form an important part of the basic network toolbox.[77]

In the chapter following this one we will apply some of the measures developed here to the analysis of network data from a variety of fields and in the process reveal some intriguing features and patterns that will play an important role in later developments.

7.1 DEGREE CENTRALITY

A large volume of research on networks has been devoted to the concept of *centrality*. This research addresses the question, “Which are the most important or central vertices in a network?” There are of course many possible definitions of importance, and correspondingly many centrality measures for networks. In this and the following several sections we describe some of the most widely used such measures.

Perhaps the simplest centrality measure in a network is just the degree of a vertex, the number of edges connected to it (see Section 6.9). Degree is sometimes called *degree centrality* in the social networks literature, to emphasize its use as a centrality measure. In directed networks, vertices have both an in-degree and an out-degree, and both may be useful as measures of centrality in the appropriate circumstances.

Although degree centrality is a simple centrality measure, it can be very illuminating. In a social network, for instance, it seems reasonable to suppose that individuals who have connections to many others might have more influence, more access to information, or more prestige than those who have fewer connections. A non-social network example is the use of citation counts in the evaluation of scientific papers. The number of citations a paper receives from other papers, which is simply its in-degree in the citation network, gives a crude measure of whether the paper has been influential or not and is widely used as a metric for judging the impact of scientific research.

7.2 EIGENVECTOR CENTRALITY

A natural extension of the simple degree centrality is *eigenvector centrality*. We can think of degree centrality as awarding one "centrality point" for every network neighbor a vertex has. But not all neighbors are equivalent. In many circumstances a vertex's importance in a network is increased by having connections to other vertices that are *themselves important*. This is the concept behind eigenvector centrality. Instead of awarding vertices just one point for each neighbor, eigenvector centrality gives each vertex a score proportional to the sum of the scores of its neighbors. Here's how it works.

Let us make some initial guess about the centrality x_i of each vertex *i*. For instance, we could start off by setting $x_i = 1$ for all *i*. Obviously this is not a useful measure of centrality, but we can use it to calculate a better one x'_i, which we define to be the sum of the centralities of *i*'s neighbors thus:

$$x'_i = \sum_j A_{ij} x_j,$$

(7.1)

where A_{ij} is an element of the adjacency matrix. We can also write this expression in matrix notation as $\mathbf{x}' = \mathbf{A}\mathbf{x}$, where $\mathbf{x}$ is the vector with elements x_i. Repeating this process to make better estimates, we have after *t* steps a vector of centralities $\mathbf{x}(t)$ given by

$$\mathbf{x}(t) = \mathbf{A}^t \mathbf{x}(0).$$

(7.2)

Now let us write $\mathbf{x}(0)$ as a linear combination of the eigenvectors $\mathbf{v}_i$ of the adjacency matrix thus:

$$\mathbf{x}(0) = \sum_i c_i \mathbf{v}_i\,,$$

(7.3)

for some appropriate choice of constants c_i. Then

$$\mathbf{x}(t) = \mathbf{A}^t \sum_i c_i \mathbf{v}_i = \sum_i c_i \kappa_i^t \mathbf{v}_i = \kappa_1^t \sum_i c_i \left[\frac{\kappa_i}{\kappa_1}\right]^t \mathbf{v}_i,$$

(7.4)

where the κ_i are the eigenvalues of **A**, and κ_1 is the largest of them. Since $\kappa_i/\kappa_1 < 1$ for all $i \neq 1$, all terms in the sum other than the first decay exponentially as t becomes large, and hence in the limit $t \to \infty$ we get $\mathbf{x}(t) \to c_1 \kappa_1^t \mathbf{v}_1$. In other words, the limiting vector of centralities is simply proportional to the leading eigenvector of the adjacency matrix. Equivalently we could say that the centrality **x** satisfies

$$\mathbf{A}\mathbf{x} = \kappa_1 \mathbf{x}.$$

(7.5)

This then is the eigenvector centrality, first proposed by Bonacich [49] in 1987. As promised the centrality x_i of vertex i is proportional to the sum of the centralities of i's neighbors:

$$x_i = \kappa_1^{-1} \sum_j A_{ij} x_j,$$

(7.6)

which gives the eigenvector centrality the nice property that it can be large either because a vertex has many neighbors or because it has important neighbors (or both). An individual in a social network, for instance, can be important, by this measure, because he or she knows lots of people (even though those people may not be important themselves) or knows a few people in high places.

Note also that the eigenvector centralities of all vertices are non-negative. To see this, consider what happens if the initial vector **x**(0) happens to have only non-negative elements. Since all elements of the adjacency matrix are also non-negative, multiplication by **A** can never introduce any negative elements to the vector and **x**(t) in Eq. (7.2) must have all elements non-negative.[78]

Equation (7.5) does not fix the normalization of the eigenvector centrality, although typically this doesn’t matter because we care only about which vertices have high or low centrality and not about absolute values. If we wish, however, we can normalize the centralities by, for instance, requiring that they sum to n (which insures that average centrality stays constant as the network gets larger).

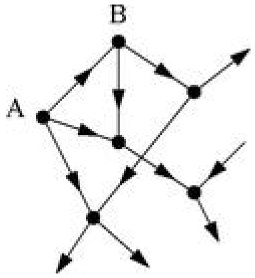


Figure 7.1: A portion of a directed network. Vertex A in this network has only outgoing edges and hence will have eigenvector centrality zero. Vertex B has outgoing edges and one ingoing edge, but the ingoing one originates at A, and hence vertex B will also have centrality zero.

In theory eigenvector centrality can be calculated for either undirected or directed networks. It works best however for the undirected case. In the directed case other complications arise. First of all, a directed network has an adjacency matrix that is, in general, asymmetric (see Section 6.4). This means that it has two sets of eigenvectors, the left eigenvectors and the right eigenvectors, and hence two leading eigenvectors. So which of the two should we use to define the centrality? In most cases the correct answer is to use the right eigenvector. The reason is that centrality in directed networks is usually bestowed by other vertices pointing towards you, rather than by you pointing to others. On the World Wide Web, for instance, the number and stature of web pages that point to your page can give a reasonable indication of how important or useful your page is. On the other hand, the fact that your page might point to other important pages is neither here nor there. Anyone can set up a page that points to a thousand others, but that does not make the page important.[79] Similar considerations apply also to citation networks and other directed networks. Thus the correct definition of eigenvector centrality for a vertex *i* in a directed network makes it proportional to the centralities of the vertices that point to *i* thus:

$$x_i = \kappa_1^{-1} \sum_j A_{ij} x_j,$$

(7.7)

which gives $\mathbf{Ax} = \kappa_1 \mathbf{x}$ in matrix notation, where $\mathbf{x}$ is the right leading eigenvector.

However, there are still problems with eigenvector centrality on directed networks. Consider Fig. 7.1. Vertex A in this figure is connected to the rest of the network, but has only outgoing edges and no incoming ones. Such a vertex will always have centrality zero because there are no terms in the sum in Eq. (7.7). This might not seem to be a problem: perhaps a vertex that no one points to *should* have centrality zero. But then consider vertex B, which has one ingoing edge, but that edge originates at vertex A, and hence B also has centrality zero, because the one term in its sum in Eq. (7.7) is zero. Taking this argument further, we see that a vertex may be pointed to by others that themselves are pointed to by many more, and so on through many generations, but if the progression ends up at a vertex or vertices that have in-degree zero, it is all for nothing—the final value of the centrality will still be zero.

In mathematical terms, only vertices that are in a strongly connected component of two or more vertices, or the out-component of such a component, can have non-zero eigenvector centrality.[80] In many cases, however, it is appropriate for vertices with high in-degree to have high centrality even if they are not in a strongly-connected component or its out-component. Web pages with many links, for instance, can reasonably be considered important even if they are not in a strongly connected component. Recall also that acyclic networks, such as citation networks, have no strongly connected components of more than one vertex (see Section 6.11.1), so all vertices will have centrality zero. Clearly this make the standard eigenvector centrality completely useless for acyclic networks.

A variation on eigenvector centrality that addresses these problems is the Katz centrality, which is the subject of the next section.

7.3 KATZ CENTRALITY

One solution to the issues of the previous section is the following: we simply give each vertex a small amount of centrality "for free," regardless of its position in the network or the centrality of its neighbors. In other words, we define

$$x_i = \alpha \sum_j A_{ij} x_j + \beta,$$

(7.8)

where α and β are positive constants. The first term is the normal eigenvector centrality term in which the centralities of the vertices linking to i are summed, and the second term is the "free" part, the constant extra term that all vertices receive. By adding this second term, even vertices with zero in-degree still get centrality β, and once they have a non-zero centrality, then the vertices they point to derive some advantage from being pointed to. This means that any vertex that is pointed to by many others will have a high centrality, although those that are pointed to by others with high centrality themselves will still do better.

In matrix terms, Eq. (7.8) can be written

$$\mathbf{x} = \alpha \mathbf{A}\mathbf{x} + \beta \mathbf{1},$$

(7.9)

where **1** is the vector (1, 1, 1 ...). Rearranging for **x**, we find that $\mathbf{x} = \beta(\mathbf{I} - \alpha\mathbf{A})^{-1} \cdot \mathbf{1}$. As we have said, we normally don't care about the absolute magnitude of the centrality, only about which vertices have high or low centrality values, so the overall multiplier β is unimportant. For convenience we usually set $\beta = 1$, giving

$$\mathbf{x} = (\mathbf{I} - \alpha\mathbf{A})^{-1} \cdot \mathbf{1}.$$

(7.10)

This centrality measure was first proposed by Katz in 1953 [169] and we will refer to it as the *Katz centrality*.

The Katz centrality differs from ordinary eigenvector centrality in the important respect of having a free parameter α, which governs the balance between the eigenvector term and the constant term in Eq. (7.8). If we wish to make use of the Katz centrality we must first choose a value for this constant. In doing so it is important to understand that α cannot be arbitrarily large. If we let $\alpha \rightarrow 0$, then only the constant term survives in Eq. (7.8) and all vertices have the same centrality β (which we have set to 1). As we increase α from zero the centralities increase and eventually there comes a point at which they diverge. This happens at the point where $(\mathrm{I} - \alpha\mathbf{A})^{-1}$ diverges in Eq. (7.10), i.e., when det $(\mathbf{I} - \alpha\mathbf{A})$ passes through zero. Rewriting this condition as

$$\det(\mathbf{A} - \alpha^{-1}\mathbf{I}) = 0,$$

(7.11)

we see that it is simply the characteristic equation whose roots α^{-1} are equal to the eigenvalues of the adjacency matrix.[81] As α increases, the determinant first crosses zero when $\alpha^{-1} = \kappa_1$, the largest eigenvalue of $\mathbf{A}$, or alternatively when $\alpha = 1/\kappa_1$. Thus, we should choose a value of α less than this if we wish the expression for the centrality to converge.[82]

Beyond this, however, there is little guidance to be had as to the value that α should take. Most researchers have employed values close to the maximum of $1/\kappa_1$, which places the maximum amount of weight on the eigenvector term and the smallest amount on the constant term. This returns a centrality that is numerically quite close to the ordinary eigenvector centrality, but gives small non-zero values to vertices that are not in the strongly connected components or their out-components.

The Katz centrality can be calculated directly from Eq. (7.10) by inverting the matrix on the right-hand side, but often this isn't the best way to do it. Inverting a matrix on a computer takes an amount of time proportional to n^3, where n is the number of vertices. This makes direct calculation of the Katz centrality prohibitively slow for large networks. Networks of more than a thousand vertices or so present serious problems.

A better approach in many cases is to evaluate the centrality directly from Eq. (7.8) (or equivalently, Eq. (7.9)). One makes an initial estimate of $\mathbf{x}$—probably a bad one, such as $\mathbf{x} = 0$—and uses that to calculate a better estimate

$$\mathbf{x}' = \alpha\mathbf{A}\mathbf{x} + \beta\mathbf{1}.$$

(7.12)

Repeating the process many times, $\mathbf{x}$ converges to a value close to the correct centrality. Since $\mathbf{A}$ has m non-zero elements, each iteration requires m multiplication operations and the total time for the calculation is proportional to rm, where r is the number of iterations necessary for the calculation to converge. Unfortunately, r depends on the details of the network and on the choice of α, so we cannot give a general guide to how many iterations will be necessary. Instead one must watch the values of x_i to observe when they converge to constant values. Nonetheless, for large networks it is almost always worthwhile to evaluate the centrality this way rather than by inverting the matrix.

We have presented the Katz centrality as a solution to the problems encountered with ordinary eigenvector centrality in directed networks. However, there is no reason in principle why one cannot use Katz centrality in undirected networks as well, and there are times when this might be useful. The idea of adding a constant term to the centrality so that each vertex gets some weight just by virtue of existing is a natural one. It allows a vertex that has many neighbors to have high centrality regardless of whether those neighbors themselves have high centrality, and this could be desirable in some applications.

A possible extension of the Katz centrality is to consider cases in which the additive constant term in Eq. (7.8) is not the same for all vertices. One could define a generalized centrality measure by

$$x_i = \alpha \sum_j A_{ij} x_j + \beta_i,$$

(7.13)

where β_i is some intrinsic, non-network contribution to the centrality for each vertex. For example, in a social network the importance of an individual might depend on non-network factors such as their age or income and if we had information about these factors we could incorporate it into the values of the β_i. Then the vector **x** of centralities is given by

$$\mathbf{x} = (\mathbf{I} - \alpha\mathbf{A})^{-1}\boldsymbol{\beta},$$

(7.14)

where **β** is the vector whose elements are the β_i. One nice feature of this approach is that the difficult part of the calculation—the inversion of the matrix—only has to be done once for a given network and choice of α. For difference choices of the β_i we need not recalculate the inverse, but simply multiply the inverse into different vectors **β**.

7.4 PAGERANK

The Katz centrality of the previous section has one feature that can be undesirable. If a vertex with high Katz centrality points to many others then those others also get high centrality. A high-centrality vertex pointing to one million others gives all one million of them high centrality. One could argue—and many have—that this is not always appropriate. In many cases it means less if a vertex is only one among many that are pointed to. The centrality gained by virtue of receiving an edge from a prestigious vertex is diluted by being shared with so many others. For instance, the famous *Yahoo!* web directory might contain a link to my web page, but it also has links to millions of other pages. *Yahoo!* is an important website, and would have high centrality by any sensible measure, but should I therefore be considered very important by association? Most people would say not: the high centrality of *Yahoo!* will get diluted and its contribution to the centrality of my page should be small because my page is only one of millions.

We can allow for this by defining a variation on the Katz centrality in which the centrality I derive from my network neighbors is proportional to their centrality *divided by their out-degree*. Then vertices that point to many others pass only a small amount of centrality on to each of those others, even if their own centrality is high.

In mathematical terms this centrality is defined by

$$x_i = \alpha \sum_j A_{ij} \frac{x_j}{k_j^{\text{out}}} + \beta.$$

(7.15)

This gives problems however if there are vertices in the network with out-degree $k_i^{\text{out}} = 0$. If there are any such vertices then the first term in Eq. (7.15) is indeterminate—it is equal to zero divided by zero (because A_{ij} = 0 for all i). This problem is easily fixed however. It is clear that vertices with no out-going edges should contribute zero to the centrality of any other vertex, which we can contrive by artificially setting $k_i^{\text{out}} = 1$ for all such vertices. (In fact, we could set k_i^{out} to any non-zero value and the calculation would give the same answer.)

In matrix terms, Eq. (7.15), is then

$$\mathbf{x} = \alpha \mathbf{A}\mathbf{D}^{-1}\mathbf{x} + \beta\mathbf{1},$$

(7.16)

with **1** being again the vector (1, 1, 1, ...) and **D** being the diagonal matrix with elements $D_{ii} = \max(k_i^{\text{out}}, 1)$. Rearranging, we find that $\mathbf{x} = \beta(\mathbf{I} - \alpha\mathbf{A}\mathbf{D}^{-1})^{-1} \cdot 1$, and thus, as before, β plays the role only of an unimportant overall multiplier for the centrality. Conventionally we set $\beta = 1$, giving

$$\mathbf{x} = (\mathbf{I} - \alpha\mathbf{A}\mathbf{D}^{-1})^{-1}\mathbf{1} = \mathbf{D}(\mathbf{D} - \alpha\mathbf{A})^{-1}\mathbf{1}.$$

(7.17)

Web search is discussed in more detail in Section 19.1.

This centrality measure is commonly known as *PageRank*, which is the trade name given it by the Google web search corporation, which uses it as a central part of their web ranking technology [55]. The aim of the Google web search engine is to generate lists of useful web pages from a preassembled index of pages in response to text queries. It does this by first searching the index for pages matching a given query using relatively simple criteria such as text matching, and then ranking the answers according to scores based on a combination of ingredients of which PageRank is one. Google returns useful answers to queries not because it is better at finding relevant pages, but because it is better at deciding what order to present its findings in: its perceived accuracy arises because the results at the top of the list of answers it returns are often highly relevant to the query, but it is possible and indeed likely that many irrelevant answers also appear on the list, lower down.

PageRank works on the Web precisely because having links to your page from important pages elsewhere is a good indication that your page may be important too. But the added ingredient of dividing by the out-degrees of pages insures that pages that simply point to an enormous number of others do not pass much centrality on to any of them, so that, for instance, network hubs like *Yahoo!* do not have a disproportionate influence on the rankings.

As with the Katz centrality, the formula for PageRank, Eq. (7.17), contains one free parameter α, whose value must be chosen somehow before the algorithm can be used. By analogy with Eq. (7.11) and the argument that follows it, we can see that the value of α should be less than the inverse of the largest eigenvalue of $\mathbf{AD}^{-1}$. For an undirected network this largest eigenvalue turns out to be 1 and the corresponding eigenvector is $(k_1, k_2, k_3, \ldots)$, where k_i is the degree of the *i*th vertex.[83] Thus α should be chosen less than 1. For a directed network, this result does not follow and in general the leading eigenvalue will be different from 1, although in practical cases it is usually still roughly of order 1.

The *Google* search engine uses a value of $\alpha = 0.85$ in its calculations, although it's not clear that there is any rigorous theory behind this choice. More likely it is just a shrewd guess based on experimentation to find out what works well.

As with the Katz centrality we can generalize PageRank to the case where the additive constant term in Eq. (7.15) is different for different vertices:

$$x_i = \alpha \sum_j A_{ij} \frac{x_j}{k_j^{\text{out}}} + \beta_i.$$

(7.18)

In matrix form this gives a solution for the centrality vector of

$$\mathbf{x} = \mathbf{D}(\mathbf{D} - \alpha\mathbf{A})^{-1}\boldsymbol{\beta}.$$

(7.19)

One could, for instance, use this for ranking web pages, giving β_i a value based perhaps on textual relevance to a search query. Pages that contained the word or words being searched for more often or in more prominent places could be given a higher intrinsic centrality than others, thereby pushing them up the rankings. The author is not aware, however, of any cases in which this technique has been implemented in practice.

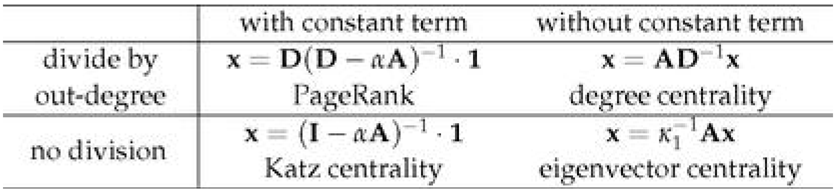
Finally, one can also imagine a version of PageRank that did not have the additive constant term in it at all:

$$x_i = \alpha \sum_j A_{ij} \frac{x_j}{k_j},$$

(7.20)

which is similar to the original eigenvector centrality introduced back in Section 7.2, but now with the extra division by k_j. For an undirected network, however, this measure is trivial: it is easy to see that it gives simply $x_i = k_i$ and therefore is just the same as ordinary degree centrality. For a directed network, on the other hand, it does not reduce to any equivalent simple value and it might potentially be of use, although it does not seem to have found use in any prominent application. (It does suffer from the same problem as the original eigenvector centrality, that it gives non-zero scores only to vertices that fall in a strongly connected component of two or more vertices or in the out-component of such a component. All other vertices get a zero score.)

In Table 7.1 we give a summary of the different matrix centrality measures we have discussed, organized according to their definitions and properties. If you want to use one of these measures in your own calculations and find the many alternatives bewildering, eigenvector centrality and PageRank are probably the two measures to focus on initially. They are the two most commonly used measures of this type. The Katz centrality has found widespread use in the past but has been favored less in recent work, while the PageRank measure without the constant term, Eq. (7.20), is the same as degree centrality for undirected networks and not in common use for directed ones.



	with constant term	without constant term
divide by out-degree	$\mathbf{x} = \mathbf{D}(\mathbf{D} - \alpha\mathbf{A})^{-1} \cdot \mathbf{1}$ PageRank	$\mathbf{x} = \mathbf{A}\mathbf{D}^{-1}\mathbf{x}$ degree centrality
no division	$\mathbf{x} = (\mathbf{I} - \alpha\mathbf{A})^{-1} \cdot \mathbf{1}$ Katz centrality	$\mathbf{x} = \kappa_1^{-1}\mathbf{A}\mathbf{x}$ eigenvector centrality

Table 7.1: Four centrality measures. The four matrix-based centrality measures discussed in the text are distinguished by whether or not they include an additive constant term in their definition and whether they are normalized by dividing by the degrees of neighboring vertices. Note that the diagonal matrix **D**, which normally has elements $D_{ii} = k_i$, must be defined slightly differently for PageRank, as $D_{ii} = \max(1, k_i)$—see Eq. (7.15) and the following discussion. Each of the measures can be applied to directed networks as well as undirected ones, although only three of the four are commonly used in this way. (The measure that appears in the top right corner of the table is

equivalent to degree centrality in the undirected case but takes more complicated values in the directed case and is not widely used.)

7.5 HUBS AND AUTHORITIES

In the case of directed networks, there is another twist to the centrality measures introduced in this section. So far we have considered measures that accord a vertex high centrality if those that point to it have high centrality. However, in some networks it is appropriate also to accord a vertex high centrality if it *points to* others with high centrality. For instance, in a citation network a paper such as a review article may cite other articles that are authoritative sources for information on a particular subject. The review itself may contain relatively little information on the subject, but it tells us where to find the information, and this on its own makes the review useful. Similarly, there are many examples of web pages that consist primarily of links to other pages on a given topic or topics and such a page of links could be very useful even if it does not itself contain explicit information on the topic in question.

Thus there are really two types of important node in these networks: *authorities* are nodes that contain useful information on a topic of interest; *hubs* are nodes that tell us where the best authorities are to be found. An authority may also be a hub, and vice versa: review articles often contain useful discussions of the topic at hand as well as citations to other discussions. Clearly hubs and authorities only exist in directed networks, since in the undirected case there is no distinction between pointing to a vertex and being pointed to.

One can imagine defining two different types of centrality for directed networks, the *authority centrality* and the *hub centrality*, which quantify vertices' prominence in the two roles. This idea was first put forward by Kleinberg [176] and developed by him into a centrality algorithm called *hyperlink-induced topic search* or *HITS*.

The HITS algorithm gives each vertex i in a network an authority centrality x_i and a hub centrality y_i. The defining characteristic of a vertex with high authority centrality is that it is pointed to by many hubs, i.e., by many other vertices with high hub centrality. And the defining characteristic of a vertex with high hub centrality is that it *points to* many vertices with high authority centrality.

Thus an important scientific paper (in the authority sense) would be one cited in many important reviews (in the hub sense). An important review is one that cites many important papers. Reviews, however, are not the only publications that can have high hub centrality. Ordinary papers can have high hub centrality too if they cite many other important papers, and papers can have both high authority and high hub centrality. Reviews too may be cited by other hubs and hence have high authority centrality as well as high hub centrality.

In Kleinberg's approach, the authority centrality of a vertex is defined to be proportional to the sum of the hub centralities of the vertices that point to it:

$$x_i = \alpha \sum_j A_{ij} y_j ,$$

(7.21)

where α is a constant. Similarly the hub centrality of a vertex is proportional to the sum of the authority centralities of the vertices it points to:

$$y_i = \beta \sum_j A_{ji} x_j ,$$

(7.22)

with β another constant. Notice that the indices on the matrix element A_{ji} are swapped around in this second equation: it is the vertices that i points to that define its hub centrality.

In matrix terms these equations can be written as

$$\mathbf{x} = \alpha \mathbf{A}\mathbf{y}, \qquad \mathbf{y} = \beta \mathbf{A}^T \mathbf{x},$$

(7.23)

or, combining the two,

$$\mathbf{A}\mathbf{A}^T\mathbf{x} = \lambda \mathbf{x}, \qquad \mathbf{A}^T\mathbf{A}\mathbf{y} = \lambda \mathbf{y},$$

(7.24)

where $\lambda = (\alpha\beta)^{-1}$. Thus the authority and hub centralities are respectively given by eigenvectors of $\mathbf{AA}^T$ and $\mathbf{A}^T\mathbf{A}$ with the same eigenvalue. By an argument similar to the one we used for the standard eigenvector centrality in Section 7.1 we can show that we should in each case take the eigenvector corresponding to the leading eigenvalue.

A crucial condition for this approach to work, is that $\mathbf{AA}^T$ and $\mathbf{A}^T\mathbf{A}$ have the same leading eigenvalue λ, otherwise we cannot satisfy both conditions in Eq. (7.24). It is easily proved, however, that this is the case, and in fact that all eigenvalues are the same for the two matrices. If $\mathbf{AA}^T\mathbf{x} = \lambda\mathbf{x}$ then multiplying both sides by $\mathbf{A}^T$ gives

$$\mathbf{A}^T\mathbf{A}(\mathbf{A}^T\mathbf{x}) = \lambda(\mathbf{A}^T\mathbf{x}),$$

(7.25)

and hence $\mathbf{A}^T\mathbf{x}$ is an eigenvector of $\mathbf{A}^T\mathbf{A}$ with the same eigenvalue λ. Comparing with Eq. (7.24) this means that

$$\mathbf{y} = \mathbf{A}^T\mathbf{x},$$

(7.26)

which gives us a fast way of calculating the hub centralities once we have the authority ones—there is no need to solve both the eigenvalue equations in Eq. (7.24) separately.

Note that $\mathbf{AA}^T$ is precisely the cocitation matrix defined in Section 6.4.1 (Eq. (6.8)) and the authority centrality is thus, roughly speaking, the eigenvector centrality for the cocitation network.[84] Similarly $\mathbf{A}^T\mathbf{A}$ is the bibliographic coupling matrix, Eq. (6.11), and hub centrality is the eigenvector centrality for the bibliographic coupling network.

A nice feature of the hub and authority centralities is that they circumvent the problems that ordinary eigenvector centrality has with directed networks, that vertices outside of strongly connected components or their out-components always have centrality zero. In the hubs and authorities approach vertices not cited by any others have authority centrality zero (which is reasonable), but they can still have non-zero hub centrality. And the vertices that *they* cite can then have non-zero authority centrality by virtue of being cited. This is perhaps a more elegant solution to the problems of eigenvector centrality in directed networks than the more ad hoc method of introducing an additive constant term as we did in Eq. (7.8). We can still introduce such a constant term into the HITS algorithm if we wish, or employ any of the other variations considered in previous sections, such as normalizing vertex centralities by the degrees of the vertices that point to them. Some variations along these lines are explored in Refs. [52, 256], but we leave the pursuit of such details to the enthusiastic reader.

The HITS algorithm is an elegant construction that should in theory provide more information about vertex centrality than the simpler measures of previous sections, but in practice it has not yet found much application. It is used as the basis for the web search engines *Teoma* and *Ask.com*, and will perhaps in future find further use, particularly in citation networks, where it holds clear advantages over other eigenvector measures.

7.6 CLOSENESS CENTRALITY

An entirely different measure of centrality is provided by the *closeness centrality,* which measures the mean distance from a vertex to other vertices. In Section 6.10.1 we encountered the concept of the geodesic path, the shortest path through a network between two vertices. Suppose d_{ij} is the length of a geodesic path from *i* to *j*, meaning the number of edges along the path.[85] Then the mean geodesic distance from *i* to *j*, averaged over all vertices *j* in the network, is

$$\ell_i = \frac{1}{n}\sum_j d_{ij}.$$

(7.27)

This quantity takes low values for vertices that are separated from others by only a short geodesic distance on average. Such vertices might have better access to information at other vertices or more direct influence on other vertices. In a social network, for instance, a person with lower mean distance to others might find that their opinions reach others in the community more quickly than the opinions of someone with higher mean distance.

In calculating the average distance some authors exclude from the sum in (7.27) the term for *j* = *i*, so that

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

(7.28)

which is a reasonable strategy, since a vertex's influence on itself is usually not relevant to the working of the network. On the other hand, the distance d_{ii} from *i* to itself is zero by definition, so this term in fact contributes nothing to the sum. The only difference the change makes to ℓ_i is in the leading divisor, which becomes 1/(*n* - 1) instead of 1/*n*, meaning that ℓ_i changes by a factor of *n*/(*n* - 1). Since this factor is independent of *i* and since, as we have said, we usually care only about the relative centralities of different vertices and not about their absolute values, we can in most cases ignore the difference between Eqs. (7.27) and (7.28). In this book we use (7.27) because it tends to give slightly more elegant analytic results.

The mean distance ℓ_i is not a centrality measure in the same sense as the others in this chapter, since it gives *low* values for more central vertices and high values for less central ones, which is the opposite of our other measures. In the social networks literature, therefore, researchers commonly calculate the inverse of ℓ_i rather than ℓ_i itself. This inverse is called the *closeness centrality* C_i:

$$C_i = \frac{1}{\ell_i} = \frac{n}{\sum_j d_{ij}}.$$

(7.29)

Closeness centrality is a very natural measure of centrality and is often used in social and other network studies. But it has some problems. One issue is that its values tend to span a rather small dynamic range from largest to smallest. As discussed in Sections 3.6, 8.2, and 12.7, geodesic distances d_{ij} between vertices in most networks tend to be small, the typical distance increasing only logarithmically with the size of the entire network. This means that the ratio between the smallest distance, which is 1, and the largest, which is of order log *n*, is itself only of order log *n*, which is small. But the smallest and largest distances provide lower and upper bounds on the average distance ℓ_i, and hence the range of values of ℓ_i and similarly of C_i is also small. In a typical network the values of C_i might span a factor of five or less. What this means in practice is that it is difficult to distinguish between central and less central vertices using this measure: the values tend to be cramped together with the differences between adjacent values showing up only when you examine the trailing digits. This means that even small fluctuations in the structure of the network can change the order of the values substantially.

For example, it has become popular in recent years to rank film actors according to their closeness centrality in the network of who has appeared in films with who else [323]. Using data from the Internet Movie Database,[86] we find that in the largest component of the network, which includes more than 98% of all actors, the smallest closeness centrality of any actor is 2.4138 for the actor Christopher Lee,[87] while the largest is 8.6681 for an Iranian actress named Leia Zanganeh. The ratio of the two is just 3.6 and about half a million other actors lie in between. As we can immediately see, the values must be very closely spaced. The second best centrality score belongs to actor Donald Pleasence, who scores 2.4164, just a tenth of a percent less than winner Lee. Because of the close spacing of values, the leaders under this dubious measure of superiority change frequently as the small details of the film network shift when new films are made or old ones added to the database. In an analysis using an earlier version of the database, Watts and Strogatz [323] proclaimed Rod Steiger to be the actor with the lowest closeness centrality. Steiger falls in sixth place in our analysis and it is entirely possible that the rankings will have changed again by the time you read this. Other centrality measures, including degree centrality and eigenvector centrality, typically don’t suffer from this problem because they have a wider dynamic range and the centrality values, particular those of the leaders, tend to be widely separated.

The closeness centrality has another problem too. If, as discussed in Section 6.10.1, we define the geodesic distance between two vertices to be infinite if the vertices fall in different components of the network, then ℓ_i is infinite for all *i* in any network with more than one component and C_i is zero. There are two strategies for getting around this. The most common one is simply to average over only those vertices in the same component as *i*. Then *n* in Eq. (7.29) becomes the number of vertices in the component and the sum is over only that component. This gives us a finite measure, but one that has its own problems. In particular, distances tend to be smaller between vertices in small components, so that vertices in such components get lower values of ℓ_i and higher closeness centrality than their counterparts in larger components. This is usually undesirable: in most cases vertices in small components are considered *less* well connected than those in larger ones and should therefore be given lower centrality.

Perhaps a better solution, therefore, is to redefine closeness in terms of the harmonic mean distance between vertices, i.e., the average of the inverse distances:

$$C_i' = \frac{1}{n-1} \sum_{j(\neq i)} \frac{1}{d_{ij}}.$$

(7.30)

(Notice that we are obliged in this case to exclude from the sum the term for $j = i$, since $d_{ii} = 0$ which would make this term infinite. This means that the sum has only n - 1 terms in it, hence the leading factor of 1/(n - 1).)

This definition has a couple of nice properties. First, if $d_{ij} = \infty$ because i and j are in different components, then the corresponding term in the sum is simply zero and drops out. Second, the measure naturally gives more weight to vertices that are close to i than to those far away. Intuitively we might imagine that the distance to close vertices is what matters in most practical situations—once a vertex is far away in a network it matters less exactly how far away it is, and Eq. (7.30) reflects this, having contributions close to zero from all such vertices.

Despite its desirable qualities, however, Eq. (7.30) is rarely used in practice. We have seen it employed only occasionally.

An interesting property of entire networks, which is related to the closeness centrality, is the mean geodesic distance between vertices. In Section 8.2 we will use measurements of mean distance in networks to study the so-called "small-world effect."

For a network with only one component, the mean distance between pairs of vertices, conventionally denoted just ℓ (now without the subscript), is

$$\ell = \frac{1}{n^2} \sum_{ij} d_{ij} = \frac{1}{n} \sum_i \ell_i.$$

(7.31)

In other words ℓ is just the mean of ℓ_i over all vertices.

For a network with more than one component we run into the same problems as before, that d_{ij} is infinite when i and j are in different components and hence ℓ is also infinite. The most common way around this problem is to average only over paths that run between vertices in the same component. Let {C*m*} be the set of components of a network, with m = 1, 2 ... Then we define

$$\ell = \frac{\sum_m \sum_{ij \in \mathscr{C}_m} d_{ij}}{\sum_m n_m^2},$$

(7.32)

where n_m is the number of vertices in component C$_m$. This measure is now finite for all networks, although it is not now equal to a simple average over the values of ℓ_i for each vertex.

An alternative and perhaps better approach would be to use the trick from Eq. (7.30) and define a harmonic mean distance ℓ' according to

$$\frac{1}{\ell'} = \frac{1}{n(n-1)} \sum_{i \neq j} \frac{1}{d_{ij}} = \frac{1}{n} \sum_i C'_i,$$

(7.33)

or equivalently

$$\ell' = \frac{n}{\sum_i C'_i},$$

(7.34)

where C'_i is the harmonic mean closeness of Eq. (7.30). (Note that, as in (7.30), we exclude from the first sum in (7.33) the terms for $i = j$, which would be infinite since $d_{ii} = 0$.)

Equation (7.34) automatically removes any contributions from vertex pairs for which $d_{ij} = \infty$. Despite its elegance, however, Eq. (7.34), like Eq. (7.30), is hardly ever used.

7.7 BETWEENNESS CENTRALITY

A very different concept of centrality is *betweenness centrality*, which measures the extent to which a vertex lies on paths between other vertices. The idea of betweenness is usually attributed to Freeman [128] in 1977, although as Freeman himself has pointed out [129], it was independently proposed some years earlier by Anthonisse [19] in an unpublished technical report.

Suppose we have a network with something flowing around it from vertex to vertex along the edges. For instance, in a social network we might have messages, news, information, or rumors being passed from one person to another. In the Internet we have data packets moving around. Let us initially make the simple assumption that every pair of vertices in the network exchanges a message with equal probability per unit time (more precisely every pair that is actually connected by a path) and that messages always take the shortest (geodesic) path though the network, or one such path, chosen at random, if there are several. Then let us ask the following question: if we wait a suitably long time until many messages have passed between each pair of vertices, how many messages, on average, will have passed through each vertex en route to their destination? The answer is that, since messages are passing down each geodesic path at the same rate, the number passing through each vertex is simply proportional to the number of geodesic paths the vertex lies on. This number of geodesic paths is what we call the betweenness centrality, or just betweenness for short.

Vertices with high betweenness centrality may have considerable influence within a network by virtue of their control over information passing between others. The vertices with highest betweenness in our message-passing scenario are the ones through which the largest number of messages pass, and if those vertices get to see the messages in question as they pass, or if they get paid for passing the messages along, they could derive a lot of power from their position within the network. The vertices with highest betweenness are also the ones whose removal from the network will most disrupt communications between other vertices because they lie on the largest number of paths taken by messages. In real-world situations, of course, not all vertices exchange communications with the same frequency, and in most cases communications do not always take the shortest path. Nonetheless, betweenness centrality may still be an approximate guide to the influence vertices have over the flow of information between others.

Having seen the basic idea of betweenness centrality, let us make things more precise. For the sake of simplicity, suppose for the moment that we have an undirected network in which there is at most one geodesic path between any pair of vertices. (There may be zero paths if the vertices in question are in different components.) Consider the set of all geodesic paths in such a network. Then the betweenness centrality of a vertex *i* is defined to be the number of those paths that pass through *i*.

Mathematically, let n^i_{st} be 1 if vertex *i* lies on the geodesic path from *s* to *t* and 0 if it does not or if there is no such path (because *s* and *t* lie in different components of the network). Then the betweenness centrality x_i is given by

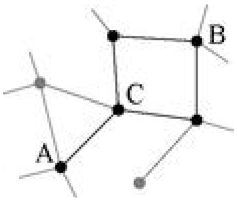
$$x_i = \sum_{st} n^i_{st}.$$

(7.35)

Note that this definition counts separately the geodesic paths in either direction between each vertex pair. Since these paths are the same on an undirected network this effectively counts each path twice. One could compensate for this by dividing x_i by 2, and often this is done, but we prefer the definition given here for a couple of reasons. First, it makes little difference in practice whether one divides the centrality by 2, since one is usually concerned only with the relative magnitudes of the centralities and not with their absolute values. Second, as discussed below, Eq. (7.35) has the advantage that it can be applied unmodified to directed networks, in which the paths in either direction between a vertex pair can differ.

Note also that Eq. (7.35) includes paths from each vertex to itself. Some people prefer to exclude such paths from the definition, so that $x_i = \sum_{s\neq t} n^i_{st}$, but again the difference is typically not important. Each vertex lies on one path from itself to itself, so the inclusion of these terms simply increases the betweenness by 1, but does not change the rankings of the vertices—which ones have higher or lower betweenness—relative to one another.

There is also a choice to be made about whether the path from *s* to *t* should be considered to pass through the vertices *s* and *t* themselves. In the social networks literature it is usually assumed that it does not. We prefer the definition where it does: it seems reasonable to define a vertex to be on a path between itself and someone else, since normally a vertex has control over information flowing from itself to other vertices or vice versa. If, however, we exclude the endpoints of the path as sociologists commonly do, the only effect is to reduce the number of paths through each vertex by twice the size of the component to which the vertex belongs. Thus the betweennesses of all vertices within a single component are just reduced by an additive constant and the ranking of vertices within the component is again unchanged. (The rankings of vertices in different components can change relative to one another, but this is rarely an issue because betweenness centrality is not typically used to compare vertices in different components, since such vertices are not competing for influence in the same arena.)



Vertices A and B are connected by two geodesic paths. Vertex C lies on both paths.

These developments are all for the case in which there is at most one geodesic path between each vertex pair. More generally, however, there may be more than one. The standard extension of betweenness to this case gives each path a weight equal to the inverse of the number of paths. For instance, if there are two geodesic paths between a given pair of vertices, each of them g ets weight $\frac{1}{2}$. Then the betweenness of a vertex is defined to be the sum of the weights of all geodesic paths passing through that vertex.

Note that the geodesic paths between a pair of vertices need not be vertex-independent, meaning they may pass through some of the same vertices (see figure). If two or more paths pass through the same vertex then the betweenness sum includes contributions from each of them. Thus if there are, say, three geodesic paths between a given pair of vertices and two of them pass through a particular vertex, then they contribute $\frac{2}{3}$ to that vertex's betweenness.

Formally, we can express the betweenness for a general network by redefining n^i_{st} to be the number of geodesic paths from *s* to *t* that pass through *i*. And we define g_{st} to be the total number of geodesic paths from *s* to *t*. Then the betweenness centrality of vertex *i* is

$$x_i = \sum_{st} \frac{n^i_{st}}{g_{st}},$$

(7.36)

where we adopt the convention that $n^i_{st}/g_{st} = 0$ if both n^i_{st} and g_{st} are zero. This definition is equivalent to our message-passing thought experiment above, in which messages pass between all pairs of vertices in a network at the same average rate, traveling along shortest paths, and in the case of several shortest paths between a given pair of vertices they choose at random between those several paths. Then x_i is proportional to the average rate at which traffic passes though vertex *i*.

Betweenness centrality can be applied to directed networks as well. In a directed network the shortest path between two vertices depends, in general, on the direction you travel in. The shortest path from A to B is different from the shortest path from B to A. Indeed there may be a path in one direction and no path at all in the other. Thus it is important in a directed network explicitly to include the path counts in either direction between each vertex pair. The definition in Eq. (7.36) already does this and so, as mentioned above, we can use the same definition without modification for the directed case. This is one reason why we prefer this definition to other slight variants that are sometimes used.

Although the generalization of betweenness to directed networks is straightforward, however, it is rarely if ever used, so we won’t discuss it further here, concentrating instead on the much more common undirected case.

Betweenness centrality differs from the other centrality measures we have considered in being not principally a measure of how well-connected a vertex is. Instead it measures how much a vertex falls “between” others. Indeed a vertex can have quite low degree, be connected to others that have low degree, even be a long way from others on average, and still have high betweenness. Consider the situation depicted in Fig. 7.2. Vertex A lies on a bridge between two groups within a network. Since any shortest path (or indeed any path whatsoever) between a vertex in one group and a vertex in the other must pass along this bridge, A acquires very high betweenness, even though it is itself on the periphery of both groups and in other respects may be not well connected: probably A would not have particularly impressive values for eigenvector or closeness centrality, and its degree centrality is only 2, but nonetheless it might have a lot of influence in the network as a result of its control over the flow of information between others. Vertices in roles like this are sometimes referred to in the sociological literature as *brokers*.[88]

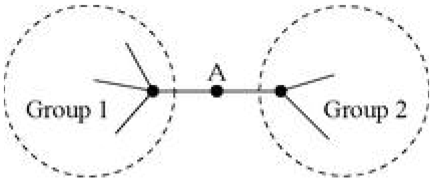
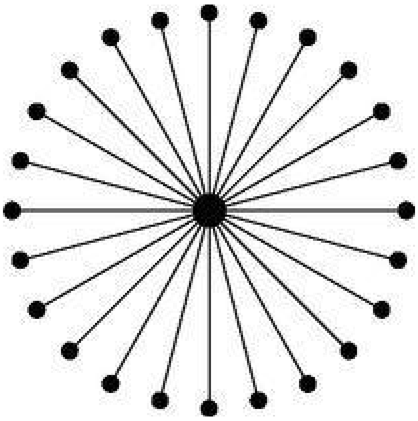


Figure 7.2: A low-degree vertex with high betweenness. In this sketch of a network, vertex A lies on a bridge joining two groups of other vertices. All paths between the groups must pass through A, so it has a high betweenness even though its degree is low.

Betweenness centrality also has another interesting property: its values are typically distributed over a wide range. The maximum possible value for the betweenness of a vertex occurs when the vertex lies on the shortest path between every other pair of vertices. This occurs for the central vertex in a *star graph*, a network composed of a vertex attached to n - 1 others by single edges. In this situation the central vertex lies on all n^2 shortest paths between vertex pairs except for the n - 1 paths from the peripheral vertices to themselves. Thus the betweenness centrality of the central vertex is $n^2 - n + 1$. At the other end of the scale, the smallest possible value of betweenness in a network with a single component is $2n - 1$, since at a minimum each vertex lies on every path that starts or ends with itself. (There are n - 1 paths from a vertex to others, n - 1 paths from others to the vertex, and one path from the vertex to itself, for a total of $2(n - 1) + 1 = 2n - 1$.) This situation occurs, for instance, when a network has a “leaf” attached to it, a vertex connected to the rest of the network by just a single edge.



A star graph.

Thus the ratio of largest and smallest possible betweenness values is

$$\frac{n^2 - n + 1}{2n - 1} \simeq \frac{1}{2}n,$$

(7.37)

where the equality becomes exact in the limit of large n. Thus in theory there could be a factor of almost $\frac{1}{2}n$ between the largest and smallest betweenness centralities, which could become very large for large networks. In real networks the range is usually considerably smaller than this, but is nonetheless large and typically increasing with increasing n.

Taking again the example of the network of film actors from the previous section, the individual with the highest betweenness centrality in the largest component of the actor network is the great Spanish actor Fernando Rey, most famous in the English-speaking world for his 1971 starring role next to Gene Hackman in *The French Connection*.[89] Rey has a betweenness score of 7.47 × 10^8, while the lowest score of any actor[90] in the large component is just 8.91 × 10^5. Thus there is a ratio of almost a thousand between the two limits—a much larger dynamic range than the ratio of 3.6 we saw in the case of closeness centrality. One consequence of this is that there are very clear winners and losers in the betweenness centrality competition. The second highest betweenness in the actor network is that of Christopher Lee (again), with 6.46 × 10^8, a 14% percent difference from winner Fernando Rey. Although betweenness values may shift a little as new movies are made and new actors added to the network, the changes are typically small compared with these large gaps between the leaders, so that the ordering at the top of the list changes relatively infrequently, giving betweenness centrality results a robustness not shared by those for closeness centrality.

The values of betweenness calculated here are raw path counts, but it is sometimes convenient to normalize betweenness in some way. Several of the standard computer programs for network analysis, such as Pajek and UCINET, perform such normalizations. One natural choice is to normalize the path count by dividing by the total number of (ordered) vertex pairs, which is n^2, so that betweenness becomes the fraction (rather than the number) of paths that run through a given vertex:

$$x_i = \frac{1}{n^2} \sum_{st} \frac{n^i_{st}}{g_{st}}.$$

(7.38)

With this definition, the values of the betweenness lie strictly between zero and one.

Some other variations on the betweenness centrality idea are worth mentioning. Betweenness gets at an important idea in network analysis, that of the flow of information or other traffic and of the influence vertices might have over that flow. However, betweenness as defined by Freeman is based on counting only the shortest paths between vertex pairs, effectively assuming that all or at least most traffic passes along those shortest paths. In reality traffic flows along paths other than the shortest in many networks. Most of us, for instance, will have had the experience of hearing news about one of our friends not from that friend directly but from another mutual acquaintance—the message has passed along a path of length two via the mutual acquaintance, rather than along the direct (geodesic) path of length one.

A version of betweenness centrality that makes some allowance for effects like this is the *flow betweenness*, which was proposed by Freeman *et al.* [130] and is based on the idea of maximum flow. Imagine each edge in a network as a pipe that can carry a unit flow of some fluid. We can ask what the maximum possible flow then is between a given source vertex *s* and target vertex *t* through these pipes. In general the answer is that more than a single unit of flow can be carried between source and target by making simultaneous use of several different paths through the network. The flow betweenness of a vertex *i* is defined according to Eq. (7.35), but with n^i_{st} being now the amount of flow through vertex *i* when the maximum flow is transmitted from *s* to *t*.

See Section 6.12 for a discussion of maximum flow in networks.

As we saw in Section 6.12, the maximum flow between vertices *s* and *t* is also equal to the number of edge-independent paths between them. Thus another way equivalent to look at the flow betweenness would be to consider n^i_{st} to be the number of independent paths between *s* and *t* that run through vertex *i*.

A slight problem arises because the independent paths between a given pair of vertices are not necessarily unique. For instance, the network shown in Fig. 7.3 has two edge-independent paths between *s* and *t* but we have two choices about what those paths are, either the paths denoted by the solid arrows, or those denoted by the dashed ones. Furthermore, our result for the flow betweenness will depend on which choice we make; the vertices labeled A and B fall on one set of paths but not the other. To get around this problem, Freeman *et al.* define the flow through a vertex for their purposes to be the *maximum* possible flow over all possible choices of paths, or equivalently the maximum number of independent paths. Thus in the network of Fig. 7.3, the contribution of the flow between *s* and *t* to the betweenness of vertex A would be 1, since this is the maximum value it takes over all possible choices of flow paths.

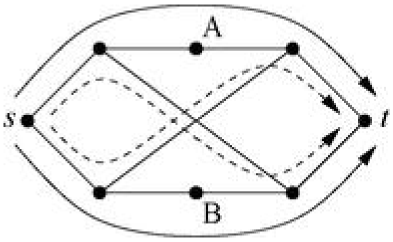


Figure 7.3: Edge-independent paths in a small network. The vertices *s* and *t* in this network have two independent paths between them, but there are two distinct ways of choosing those paths, represented by the solid and dashed curves.

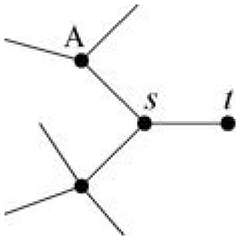
In terms of our information analogy, one can think of flow betweenness as measuring the betweenness of vertices in a network in which a maximal amount of information is continuously pumped between all sources and targets. Flow betweenness takes account of more than just the geodesic paths between vertices, since flow can go along non-geodesic paths as well as geodesic ones. (For example, the paths through vertices A and B in the example above are not geodesic.) Indeed, in some cases *none* of the paths that appear in the solution of the maximum flow problem are geodesic paths, so geodesic paths may not be counted at all by this measure.

But this point highlights a problem with flow betweenness: although it typically counts more paths than the standard shortest-path betweenness, flow betweenness still only counts a subset of possible paths, and some important ones (such as geodesic paths) may be missed out altogether. One way to look at the issue is that both shortest-path betweenness and flow betweenness assume flows that are optimal in some sense—passing only along shortest paths in the first case and maximizing total flow in the second. Just as there is no reason to suppose that information or other traffic always takes the shortest path, there is no reason in general to suppose it should act to maximize flow (although of course there may be special cases in which it does).

See Section 6.14 for a discussion of random walks.

A betweenness variant that does count all paths is the *random-walk betweenness* [243]. In this variant traffic between vertices *s* and *t* is thought of as performing an (absorbing) random walk that starts at vertex *s* and continues un til it reaches vertex *t*. The betweenness is defined according to $x_i = \sum_{st} n^i_{st}$ but with n^i_{st} now being the number of times that the random walk from *s* to *t* passes through *i* on its journey, averaged over many repetitions of the walk.

Note that in this case $n^i_{st} \neq n^i_{ts}$ in general, even on an undirected network. For instance, consider this portion of a network:



A random walk from *s* to *t* may pass through vertex *A* before returning to *s* and stepping thence to *t*, but a walk from *t* to *s* will never pass through A because its first step away from *t* will always take it to *s* and then the walk will finish.

Since every possible path from *s* to *t* occurs in a random walk with some probability (albeit a very small one) the random-walk betweenness includes contributions from all paths.[92] Note, however, that different paths appear in general with different probabilities, so paths do not contribute equally to the betweenness scores, longer paths typically making smaller contributions than shorter ones, a bias that is plausible in some but by no means all cases.

Random walk betweenness would be an appropriate betweenness measure for traffic that traverses a network with no idea of where it is going—it simply wanders around at random until it reaches its destination. Shortest-path betweenness is the exact opposite. It is the appropriate measure for information that knows exactly where it is going and takes the most direct path to get there. It seems likely that most real-world situations fall somewhere in between these two extremes. However, it is found in practice [243] that the two measures often give quite similar results, in which case one can with reasonable justification assume that the "correct" answer, the one lying between the limits set by the shortest-path and random-walk measures, is similar to both. In cases where the two differ by a considerable margin, however, we should be wary of attributing too much authority to either measure—there is no guarantee that either is telling us a great deal about true information flow in the network.

Other generalizations of betweenness are also possible, based on other models of diffusion, transmission, or flow along network edges. We refer the interested reader to the article by Borgatti [51], which draws together many of the possibilities into a broad general framework for betweenness measures.

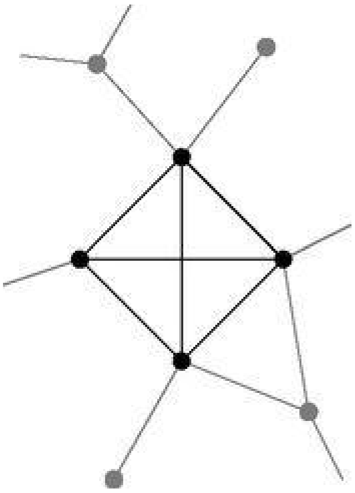
7.8 GROUPS OF VERTICES

Many networks, including social and other networks, divide naturally into groups or communities. Networks of people divide into groups of friends, coworkers, or business partners; the World Wide Web divides into groups of related web pages; biochemical networks divide into functional modules, and so forth. The definition and analysis of groups within networks is a large and fruitful area of network theory. In Chapter 11 we discuss some of the sophisticated computer methods that have been developed for detecting groups, such as hierarchical clustering and spectral partitioning. In this section we discuss some simpler concepts of network groups which can be useful for probing and describing the local structure of networks. The primary constructs we look at are cliques, plexes, cores, and components.

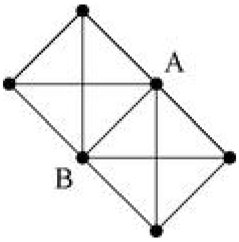
7.8.1 CLIQUES, PLEXES, AND CORES

A *clique* is a maximal subset of the vertices in an undirected network such that every member of the set is connected by an edge to every other. The word “maximal” here means that there is no other vertex in the network that can be added to the subset while preserving the property that every vertex is connected to every other. Thus a set of four vertices in a network would be a clique if (and only if) each of the four is directly connected by edges to the other three *and* if there is no other vertex anywhere in the network that could be added to make a group of five vertices all connected to each other. Note that cliques can overlap, meaning that they can share one or more of the same vertices.

The occurrence of a clique in an otherwise sparse network is normally an indication of a highly cohesive subgroup. In a social network, for instance, one might encounter a set of individuals each of whom was acquainted with each of the others, and such a clique would probably indicate that the individuals in question are closely connected—a set of coworkers in an office for example or a group of classmates in a school.



A clique of four vertices within a network.



Two overlapping cliques. Vertices A and B in this network both belong to two cliques of four vertices.

However, it's also the case that many circles of friends form only nearcliques, rather than perfect cliques. There may be some members of the group who are unacquainted, even if most members know one another. The requirement that every possible edge be present within a clique is a very stringent one, and it seems natural to consider how we might relax this requirement. One construct that does this is the *k-plex*. A *k*-plex of size *n* is a maximal subset of *n* vertices within a network such that each vertex is connected to at least *n* - *k* of the others. If *k* = 1, we recover the definition of an ordinary clique—a 1-plex is the same as a clique. If *k* = 2, then each vertex must be connected to all or all-but-one of the others. And so forth.[93] Like cliques, *k*-plexes can overlap one another; a single vertex can belong to more than one *k*-plex.

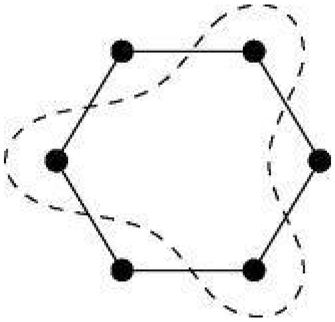
The *k*-plex is a useful concept for discovering groups within networks: in real life many groups in social and other networks form *k*-plexes. There is no solid rule about what value *k* should take. Experimentation starting from small values is the usual way to proceed. Smaller values of *k* tend to be meaningful for smaller groups, whereas in large groups the smaller values impose too stringent a constraint but larger values often give useful results. This suggests another possible generalization of the clique idea: one could specify that each member be connected to a certain *fraction* of the others, say 75% or 50%. (As far as we know, this variant doesn't have a name and it is not in wide use, but perhaps it should be.)

Many other variations on the clique idea have been proposed in the literature. For instance Flake *et al.* [122] proposed a definition of a group as a subset of vertices such that each has at least as many connections to vertices inside the group as to vertices outside. Radicchi *et al.* [276] proposed a weaker definition of a group as a subset of vertices such that the total number of connections of all vertices in the group to others in the group is greater than the total number of connections to vertices outside.[94]

Another concept closely related to the *k*-plex is the *k-core*. A *k*-core is a maximal subset of vertices such that each is connected to at least *k* others in the subset.[95] It should be obvious (or you can easily prove it for yourself) that a *k*-core of *n* vertices is also an (*n* - *k*)-plex. However, the set of all *k*-cores for a given value of *k* is not the same as the set of all *k*-plexes for any value of *k*, since *n*, the size of the group, can vary from one *k*-core to another. Also, unlike *k*-plexes (and cliques), *k*-cores cannot overlap, since by their definition two *k*-cores that shared one or more vertices would just form a single larger *k*-core.

The *k*-core is of particular interest in network analysis for the practical reason that it is very easy to find the set of all *k*-cores in a network. A simple algorithm is to start with your whole network and remove from it any vertices that have degree less than *k*, since clearly such vertices cannot under any circumstances be members of a *k*-core. In so doing, one will normally also reduce the degrees of some other vertices in the network—those that were connected to the vertices just removed. So we then go through the network again to see if there are any more vertices that now have degree less than *k* and if there are we remove those too. And so we proceed, repeatedly pruning the network to remove vertices with degree less than *k* until no such vertices remain.[96] What is left over will, by definition, be a *k*-core or a set of *k*-cores, since each vertex is connected to at least *k* others. Note that we are not necessarily left with a *single* *k*-core—there's no guarantee that the network will be connected once we are done pruning it, even if it was connected to start with.

Two other generalizations of cliques merit a brief mention. A *k-clique* is a maximal subset of vertices such that each is no more than a distance *k* away from any of the others via the edges of the network. For *k* = 1 this just recovers the definition of an ordinary clique. For larger *k* it constitutes a relaxation of the stringent requirements of the usual clique definition. Unfortunately it is not a very well-behaved one, since a *k*-clique by this definition need not be connected via paths that run within the subset (see figure). If we restrict ourselves to paths that run only within the subset then the resulting object is known as either a *k-clan* or a *k-club*. (The difference between the two lies in whether we impose the restriction that paths stay within the group from the outset, or whether we first find *k*-cliques and then discard those with outside paths. The end results can be different in the two cases. For more details see Wasserman and Faust [320].).



The outlined set of three vertices in this network constitute a 2-clique, but one that is not connected via paths within the 2-clique.

7.8.2 COMPONENTS AND *k*-COMPONENTS

In Section 6.11 we introduced the concept of a component. A component in an undirected network is a maximal subset of vertices such that each is reachable by some path from each of the others. A useful generalization of this concept is the *k*-component. A *k-component* (sometimes also called a *k-connected component)* is a maximal subset of vertices such that each is reachable from each of the others by at least *k* vertex-independent paths—see Fig. 7.4. (Recall that two paths are said to be vertex-independent if they share none of the same vertices, except the starting and ending vertices—see Section 6.12.) For the common special cases $k = 2$ and $k = 3$, *k*-components are also called *bicomponents* and *tricomponents* respectively.

A 1-component by this definition is just an ordinary component—there is at least one path between every pair of vertices—and *k*-components for $k \geq 2$ are nested within each other. A 2-component or bicomponent, for example, is necessarily a subset of a 1-component, since any pair of vertices that are connected by at least two paths are also connected by at least one path. Similarly a tricomponent is necessarily a subset of a bicomponent, and so forth. (See Fig. 7.4 again.)

As discussed in Section 6.12, the number of vertex-independent paths between two vertices is equal to the size of the vertex cut set between the same two vertices, i.e., the number of vertices that would have to be removed in order to disconnect the two. So another way of defining a *k*-component would be to say that it is a maximal subset of vertices such that no pair of vertices can be disconnected from each other by removing less than *k* vertices.

A variant of the *k*-component can also be defined using edge-independent paths, so that vertices are in the same *k*-component if they are connected by *k* or more edge-independent paths, or equivalently if they cannot be disconnected by the removal of less than *k* edges. In principal this variant could be useful in certain circumstances but in practice it is rarely used.

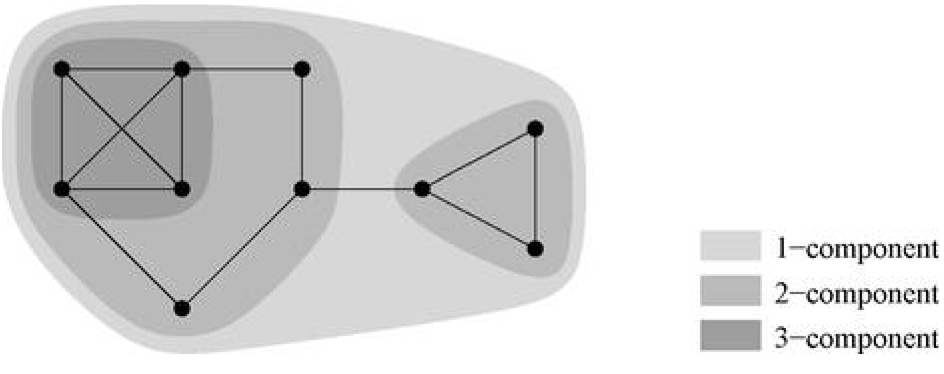
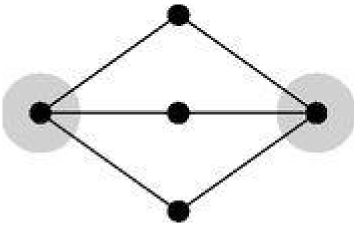


Figure 7.4: The *k*-components in a small network. The shaded regions denote the *k*-components in this small network, which has a single 1-component, two 2-components, one 3-component, and no *k*-components for any higher value of *k*. Note that the *k*-components are nested within one another, the 2-components falling inside the 1-component and the 3-component falling inside one of the 2-components.

The idea of a *k*-component is a natural one in network analysis, being connected with the idea of network robustness. For instance, in a data network such as the Internet, the number of vertex-independent paths between two vertices is also the number of independent routes that data might take between the same two vertices, and the size of the cut set between them is the number of vertices in the network—typically routers—that would have to fail or otherwise be knocked out to sever the data connection between the two endpoints. Thus a pair of vertices connected by two independent paths cannot be disconnected from one another by the failure of any single router. A pair of vertices connected by three paths cannot be disconnected by the failure of any two routers. And so forth. A *k*-component with $k \geq 2$ in a network like the Internet is a subset of the network that has robust connectivity in this sense. One would hope, for instance, that most of the network backbone—the system of high volume world-spanning links that carry long-distance data (see Section 2.1)—is a *k*-component with high *k*, so that it would be difficult for points on the backbone to lose connection with one another.



The two highlighted vertices in this network form a tricomponent, even though they are not directly connected to each other. The other three vertices are not in the tricomponent.

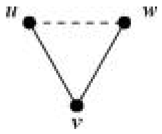
Note that for $k \geq 3$, the *k*-components in a network can be non-contiguous (see figure). Ordinary components (1-components) and bicomponents, by contrast, are always contiguous. Within the social networks literature, where non-contiguous components are often considered undesirable, *k*-components are sometimes defined slightly differently: a *k*-component is defined to be a maximal subset of vertices such that every pair in the set is connected by at least *k* vertex-independent paths *that themselves are contained entirely within the subset*. This definition rules out non-contiguous *k*-components, but it is also mathematically and computationally more difficult to work with than the standard definition. For this reason, and because there are also plenty of cases in which it is appropriate to count non-contiguous *k*-components, the standard definition remains the most widely used one in fields other than sociology.

7.9 TRANSITIVITY

A property very important in social networks, and useful to a lesser degree in other networks too, is *transitivity*. In mathematics a relation “o” is said to be transitive if *a* ◌*b* and *b* ◌*c* together imply *a* ◌*c*. An example would be equality. If $a = b$ and $b = c$, then it follows that $a = c$ also, so “=” is a transitive relation. Other examples are “greater than,” “less than,” and “implies.”

In a network there are various relations between pairs of vertices, the simplest of which is “connected by an edge.” If the “connected by an edge” relation were transitive it would mean that if vertex *u* is connected to vertex *v*, and *v* is connected to *w*, then *u* is also connected to *w*. In common parlance, “the friend of my friend is also my friend.” Although this is only one possible kind of network transitivity—other network relations could be transitive too—it is the only one that is commonly considered, and networks showing this property are themselves said to be transitive. This definition of network transitivity could apply to either directed or undirected networks, but let us take the undirected case first, since it’s simpler.

Perfect transitivity only occurs in networks where each component is a fully connected subgraph or clique, i.e., a subgraph in which all vertices are connected to all others.[97] Perfect transitivity is therefore pretty much a useless concept in networks. However, *partial* transitivity can be very useful. In many networks, particularly social networks, the fact that *u* knows *v* and *v* knows *w* doesn’t *guarantee* that *u* knows *w*, but makes it much more likely. The friend of my friend is not necessarily my friend, but is far more likely to be my friend than some randomly chosen member of the population.



The path *uvw* (solid edges) is said to be closed if the third edge directly from *u* to *w* is present (dashed edge).

We can quantify the level of transitivity in a network as follows. If *u* knows *v* and *v* knows *w*, then we have a path *uvw* of two edges in the network. If *u* also knows *w*, we say that the path is *closed*—it forms a loop of length three, or a triangle, in the network. In the social network jargon, *u*, *v*, and *w* are said to form a *closed triad*. We define the *clustering coefficient*[98] to be the fraction of paths of length two in the network that are closed. That is, we count all paths of length two, and we count how many of them are closed, and we divide the second number by the first to get a clustering coefficient *C* that lies in the range from zero to one:

$$C = \frac{\text{(number of closed paths of length two)}}{\text{(number of paths of length two)}}.$$

(7.39)

$C = 1$ implies perfect transitivity, i.e., a network whose components are all cliques. $C = 0$ implies no closed triads, which happens for various topologies, such as a tree (which has no closed loops of any kind—see Section 6.7) or a square lattice (which has closed loops with even numbers of vertices only and no closed triads).

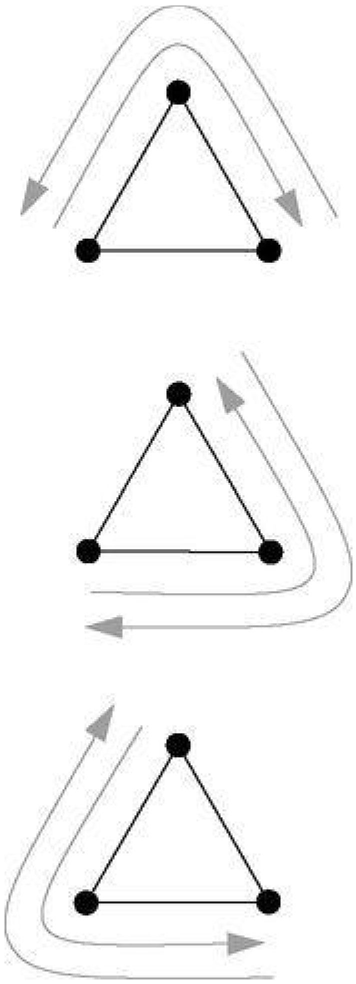
Note that paths in networks, as defined in Section 6.10 have a direction and two paths that traverse the same edges but in opposite directions are counted separately in Eq. (7.39). Thus *uvw* and *wvu* are distinct paths and are counted separately. Similarly, closed paths are counted separately in each direction.[99]

An alternative way to write the clustering coefficient is

$$C = \frac{(\text{number of triangles}) \times 6}{(\text{number of paths of length two})}.$$

(7.40)

Why the factor of six? It arises because each triangle in the network gets counted six times over when we count up the number of closed paths of length two. Suppose we have a triangle *uvw*. Then there are six paths of length two in it: *uvw*, *vwu*, *wuv*, *wvu*, *vuw*, and *uwv*. Each of these six is closed, so the number of closed paths is six times the number of triangles.



A triangle contains six distinct paths of length two, all of them closed.

Yet another way to write the clustering coefficient would be to note that if we have a path of length two, *uvw*, then it is also true to say that vertices *u* and *w* have a common neighbor in *v*—they share a mutual acquaintance in social network terms. If the triad *uvw* is closed then *u* and *w* are themselves acquainted, so the clustering coefficient can be thought of also as the fraction of pairs of people with a common friend who are themselves friends or equivalently as the mean probability that two people with a common friend are themselves friends. This is perhaps the most common way of defining the clustering coefficient. In mathematical notation:

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

(7.41)

Here a "connected triple" means three vertices *uvw* with edges (*u*, *v*) and (*v*, *w*). (The edge (*u*, *w*) can be present or not.) The factor of three in the numerator arises because each triangle gets counted three times when we count the connected triples in the network. The triangle *uvw* for instance contains the triples *uvw*, *vwu*, and *wuv*. In the older social networks literature the clustering coefficient is sometimes referred to as the "fraction of transitive triples," which is a reference to this definition of the coefficient.

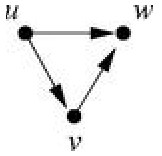
Social networks tend to have quite high values of the clustering coefficient. For example, the network of film actor collaborations discussed earlier has been found to have $C = 0.20$ [241]; a network of collaborations between biologists has been found to have $C = 0.09$ [236]; a network of who sends email to whom in a large university has $C = 0.16$ [103]. These are typical values for social networks. Some denser networks have even higher values, as high as 0.5 or 0.6. (Technological and biological networks by contrast tend to have somewhat lower values. The Internet at the autonomous system level, for instance, has a clustering coefficient of only about 0.01. This point is discussed in more detail in Section 8.6.)

In what sense are these clustering coefficients for social networks high? Well, let us assume, to make things simple, that everyone in a network has about the same number *c* of friends. Consider one of my friends in this network and suppose they pick *their* friends completely at random from the whole population. Then the chance that one of their *c* friends happens to be a particular one of my other friends would be *c/n*, where *n* is the size of the network. Thus in this network the probability of two of my friends being acquainted, which is by definition the clustering coefficient, would be just *c/n*. Of course it is not the case that everyone in a network has the same number of friends, and we will see how to perform better calculations of the clustering coefficient later (Section 13.4), but this crude calculation will serve our purposes for the moment.

For the networks cited above, the value of *c/n* is 0.0003 (film actors), 0.00001 (biology collaborations), and 0.00002 (email messages). Thus the measured clustering coefficients are *much* larger than this estimate based on the assumption of random network connections. Even though the estimate ignores, as we have said, any variation in the number of friends people have, the disparity between the calculated and observed values of the clustering coefficient is so large that it seems unlikely it could be eliminated just by allowing the number of friends to vary. A much more likely explanation is that our other assumption, that people pick their friends at random, is seriously flawed. The numbers suggest that there is a much greater chance that two people will be acquainted if they have another common acquaintance than if they don't.

Although this argument is admittedly crude, we will see in Section 8.6 how to make it more accurate and so show that our basic conclusion is indeed correct.

Some social networks, such as the email network above, are directed networks. In calculating clustering coefficients for direct networks, scientists have typically just ignored their directed nature and applied Eq. (7.41) as if the edges were undirected. It is however possible to generalize transitivity to take account of directed links. If we have a directed relation between vertices such as "*u* likes *v*" then we can say that a triple of vertices is closed or transitive if *u* likes *v*, *v* likes *w*, and also *u* likes *w*. (Note that there are many distinct ways for such a triple to be transitive, depending on the directions of the edges. The example given here is only one of six different possibilities.) One can calculate a clustering coefficient or fraction of transitive triples in the obvious fashion for the directed case, counting all directed paths of length two that are closed and dividing by the total number of directed paths of length two. For some reason, however, such measurements have not often appeared in the literature.



A transitive triple of vertices in a directed network.

7.9.1 LOCAL CLUSTERING AND REDUNDANCY

We can also define a clustering coefficient for a single vertex. For a vertex *i*, we define

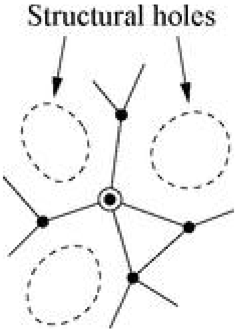
$$C_i = \frac{\text{(number of pairs of neighbors of } i \text{ that are connected)}}{\text{(number of pairs of neighbors of } i\text{)}}.$$

(7.42)

That is, to calculate C_i we go through all distinct pairs of vertices that are neighbors of *i* in the network, count the number of such pairs that are connected to each other, and divide by the total number of pairs, which is $\frac{1}{2}k_i(k_i - 1)$ where k_i is the degree of *i*,. C_i is sometimes called the *local clustering coefficient* and it represents the average probability that a pair of *i*'s friends are friends of one another.

Local clustering is interesting for several reasons. First, in many networks it is found empirically to have a rough dependence on degree, vertices with higher degree having a lower local clustering coefficient on average. This point is discussed in detail in Section 8.6.1.

Second, local clustering can be used as a probe for the existence of so-called “structural holes” in a network. While it is common in many networks, especially social networks, for the neighbors of a vertex to be connected among themselves, it happens sometimes that these expected connections between neighbors are missing. The missing links are called *structural holes* and were first studied in this context by Burt [60]. If we are interested in efficient spread of information or other traffic around a network, as we were in Section 7.7, then structural holes are a bad thing—they reduce the number of alternative routes information can take through the network. On the other hand structural holes can be a good thing for the central vertex *i* whose friends lack connections, because they give *i* power over information flow between those friends. If two friends of *i* are not connected directly and their information about one another comes instead via their mutual connection with *i* then *i* can control the flow of that information. The local clustering coefficient measures how influential *i* is in this sense, taking lower values the more structural holes there are in the network around *i*. Thus local clustering can be regarded as a type of centrality measure, albeit one that takes small values for powerful individuals rather than large ones.



When the neighbors of a node are not connected to one another we say the network contains “structural holes.”

In this sense, local clustering can also be thought of as akin to the betweenness centrality of Section 7.7. Where betweenness measures a vertex’s control over information flowing between all pairs of vertices in its component, local clustering is like a local version of betweenness that measures control over flows between just the immediate neighbors of a vertex. One measure is not necessarily better than another. There may be cases in which we want to take all vertices into account and others where we want to consider only immediate neighbors—the choice will depend on the particular questions we want to answer. It is worth pointing out however that betweenness is much more computationally intensive to calculate than local clustering (see Section 10.3.6), and that in practice betweenness and local clustering are strongly correlated [60]. There may in many cases be little to be gained by performing the more costly full calculation of betweenness and much to be saved by sticking with clustering, given that the two contain much the same information.[101]

In his original studies of structural holes, Burt [60] did not in fact make use of the local clustering coefficient as a measure of the presence of holes.[102] Instead, he used another measure, which he called *redundancy*. The original definition of redundancy was rather complicated, but Borgatti [50] has shown that it can be simplified to the following: the redundancy R_i of a vertex i is the mean number of connections from a neighbor of i to other neighbors of i. Consider the example shown in Fig. 7.5 in which vertex i has four neighbors. Each of those four *could* be acquainted with any of the three others, but in this case none of them is connected to all three. One is connected to none of the others, two are connected to one other, and the last is connected to two others. The redundancy is the average of these numbers $R_i = \frac{1}{4}(0+1+1+2) = 1$. The minimum possible value of the redundancy of a vertex is zero and the maximum is $k_i - 1$, where k_i is the degree of vertex i.

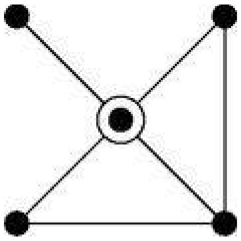


Figure 7.5: Redundancy. The neighbors of the central vertex in this figure have 0, 1, 1, and 2 connections to other neighbors respectively. The redundancy is the mean of these values: $R_i = \frac{1}{4}(0+1+1+2) = 1$.

It’s probably obvious that R_i is related to the local clustering C_i. To see precisely what the relation is, we note that if the average number of connections from a friend of i to other friends is R_i, then the total number of connections between friends is $\frac{1}{2}k_iR_i$. And the total number of pairs of friends of i is $\frac{1}{2}k_i(k_i-1)$. The local clustering coefficient, Eq. (7.42), is the ratio of these two quantities:

$$C_i = \frac{\frac{1}{2}k_iR_i}{\frac{1}{2}k_i(k_i-1)} = \frac{R_i}{k_i-1}.$$

(7.43)

Given that $k_i - 1$ is the maximum value of R_i, the local clustering coefficient can be thought of as simply a version of the redundancy rescaled to have a maximum value of 1. Applying Eq. (7.43) to the example of Fig. 7.5 implies that the local clustering coefficient for the central vertex should be $C_i = \frac{1}{3}$, and the reader can easily verify that this is indeed the case.

A third context in which the local clustering coefficient arises is in the calculation of the global clustering coefficient itself. Watts and Strogatz [323] proposed calculating a clustering coefficient for an entire network as the mean of the local clustering coefficients for each vertex:

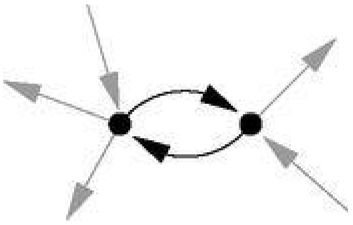
$$C_{\mathrm{WS}} = \frac{1}{n} \sum_{i=1}^{n} C_i,$$

(7.44)

where n is the number of vertices in the network. This is a different definition for the clustering coefficient from the one given earlier, Eq. (7.41), and the two definitions are not equivalent. Furthermore, they can give substantially different numbers for a given network and because both definitions are in reasonably common use this can give rise to confusion. We favor our first definition for C, Eq. (7.41), because it has a simple interpretation and because it is normally easier to calculate. Also the second definition, Eq. (7.44), tends to be dominated by vertices with low degree, since they have small denominators in Eq. (7.42), and the measure thus gives a rather poor picture of the overall properties of any network with a significant number of such vertices.[103] It's worth noting, however, that the definition of Eq. (7.44) was actually proposed before Eq. (7.41) and, perhaps because of this, it finds moderately wide use in network studies. So you need at least to be aware of both definitions and clear which is being used in any particular situation.

7.10 RECIPROCITY

The clustering coefficient of Section 7.9 measures the frequency with which loops of length three—triangles—appear in a network. Of course, there is no reason why one should concentrate only on loops of length three, and people have occasionally looked at the frequency of loops of length four or more [44, 61,133,140, 238]. Triangles occupy a special place however because in an undirected simple graph the triangle is the shortest loop we can have (and usually the most commonly occurring). However, in a *directed* network this is not the case. In a directed network, we can have loops of length two—a pair of vertices between which there are directed edges running in both directions—and it is interesting to ask about the frequency of occurrence of these loops also.



A loop of length two in a directed network.

The frequency of loops of length two is measured by the *reciprocity*, and tells you how likely it is that a vertex that you point to also points back at you. For instance, on the World Wide Web if my web page links to your web page, how likely is it, on average, that yours link back again to mine? In general, it's found that you are much more likely to link to me if I link to you than if I don't. (That probably isn't an Earth-shattering surprise, but it's good to know when the data bear out one's intuitions.) Similarly in friendship networks, such as the networks of schoolchildren described in Section 3.2 where respondents were asked to name their friends, it is much more likely that you will name me if I name you than if I do not.

If there is a directed edge from vertex *i* to vertex *j* in a directed network and there is also an edge from *j* to *i* then we say the edge from *i* to *j* is *reciprocated*. (Obviously the edge from *j* to *i* is also reciprocated.) Pairs of edges like this are also sometimes called *co-links*, particularly in the context of the World Wide Web [104].

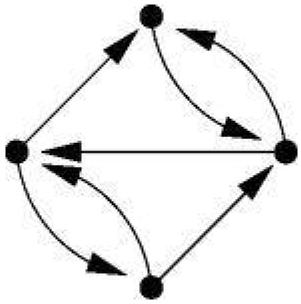
The reciprocity *r* is defined as the fraction of edges that are reciprocated. Noting that the product of adjacency matrix elements $A_{ij}A_{ji}$ is 1 if and only if there is an edge from *i* to *j* and an edge from *j* to *i* and is zero otherwise, we can sum over all vertex pairs *i*, *j* to get an expression for the reciprocity:

$$r = \frac{1}{m}\sum_{ij} A_{ij}A_{ji} = \frac{1}{m}\operatorname{Tr}\mathbf{A}^2,$$

(7.45)

where *m* is, as usual, the total number of (directed) edges in the network.

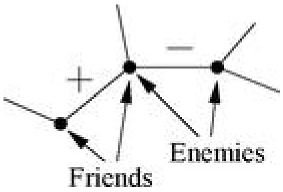
Consider for example this small network of four vertices:



There are seven directed edges in this network and four of them are reciprocated, so the reciprocity is $r = \frac{4}{7} \simeq 0.57$. In fact, this is about the same value as seen on the World Wide Web. There is about a 57% percent chance that if web page A links to web page B then B also links back to A.[104] As another example, in a study of a network of who has whom in their email address book it was found that the reciprocity was about $r = 0.23$ [248].

7.11 SIGNED EDGES AND STRUCTURAL BALANCE

In some social networks, and occasionally in other networks, edges are allowed to be either “positive” or “negative.” For instance, in an acquaintance network we could denote friendship by a positive edge and animosity by a negative edge:



One could also consider varying degrees of friendship or animosity—networks with more strongly positive or negative edges in them—but for the moment let’s stick to the simple case where each edge is in just one of two states, positive or negative, like or dislike. Such networks are called *signed networks* and their edges are called *signed edges*.

It is important to be clear here that a negative edge is not the same as the absence of an edge. A negative edge indicates, for example, two people who interact regularly but dislike each other. The absence of an edge represents two people who do not interact. Whether they would like one another if they did interact is not recorded.

Now consider the possible configurations of three edges in a triangle in a signed network, as depicted in Fig. 7.6. If “+” and “−” represent like and dislike, then we can imagine some of these configurations creating social problems if they were to arise between three people in the real world. Configuration (a) is fine: everyone likes everyone else. Configuration (b) is probably also fine, although the situation is more subtle than (a). Individuals *u* and *v* like one another and both dislike *w*, but the configuration can still be regarded as stable in the sense that *u* and *v* can agree over their dislike of *w* and get along just fine, while *w* hates both of them. No one is conflicted about their allegiances.

Put another way, *w* is *u*′s enemy and *v* is *w*′s enemy, but there is no problem with *u* and *v* being friends if one considers that the “enemy of my enemy is my friend.”

Configuration (c) however could be problematic. Individual *u* likes individual *v* and *v* likes *w*, but *u* thinks *w* is an idiot. This is going to place a strain on the friendship between *u* and *v* because *u* thinks *v*′s friend is an idiot. Alternatively, from the point of view of *v*, *v* has two friends, *u* and *w* and they don’t get along, which puts *v* in an awkward position. In many real-life situations of this kind the tension would be resolved by one of the acquaintances being broken, i.e., the edge would be removed altogether. Perhaps *v* would simply stop talking to one of his friends, for instance.

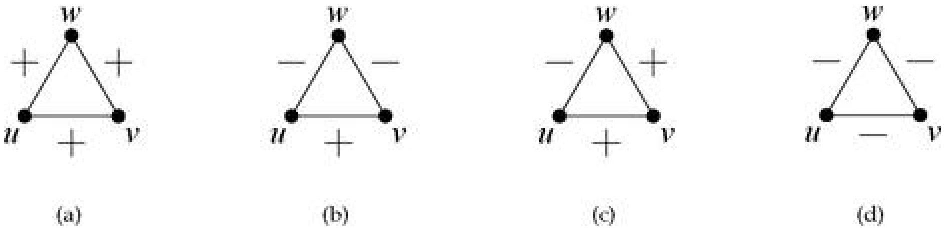
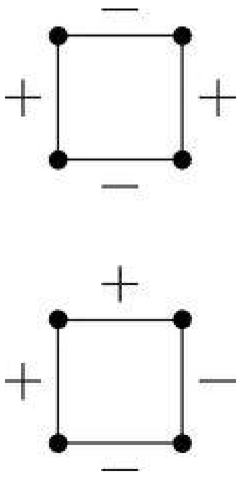


Figure 7.6: Possible triad configurations in a signed network. Configurations (a) and (b) are balanced and hence relatively stable, but configurations (c) and (d) are unbalanced and liable to break apart.

Configuration (d) is somewhat ambiguous. On the one hand, it consists of three people who all dislike each other, so no one is in doubt about where things stand: everyone just hates everyone else. On the other hand, the “enemy of my enemy” rule does not apply here. Individuals *u* and *v* might like to form an alliance in recognition of their joint dislike of *w*, but find it difficult to do so because they also dislike each other. In some circumstances this might cause tension. (Think of the uneasy alliance of the US and Russia against Germany during World War II, for instance.) But what one can say definitely is that configuration (d) is often unstable. There may be little reason for the three to stay together when none of them likes the others. Quite probably three enemies such as these would simply sever their connections and go their separate ways.



Two stable configurations in loops of length four.

The feature that distinguishes the two stable configurations in Fig. 7.6 from the unstable ones is that they have an even number of minus signs around the loop.[105] One can enumerate similar configurations for longer loops, of length four or greater, and again find that loops with even numbers of minus signs appear stable and those with odd numbers unstable.

This alone would be an observation of only slight interest, where it not for the intriguing fact that this type of stability really does appear have an effect on the structure of networks. In surveys it is found that the unstable configurations in Fig. 7.6, the ones with odd numbers of minus signs, occur far less often in real social networks than the stable configurations with even numbers of minus signs.

Networks containing only loops with even numbers of minus signs are said to show *structural balance*, or sometimes just *balance*. An important consequence of balance in networks was proved by Harary [154]:

> A balanced network can be divided into connected groups of vertices such that all connections between members of the same group are positive and all connections between members of different groups are negative.

Note that the groups in question can consist of a single vertex or many vertices, and there may be only one group or there may be very many. Figure 7.7 shows a balanced network and its division into groups. Networks that can be divided into groups like this are said to be *clusterable*. Harary's theorem tells us that all balanced networks are clusterable.

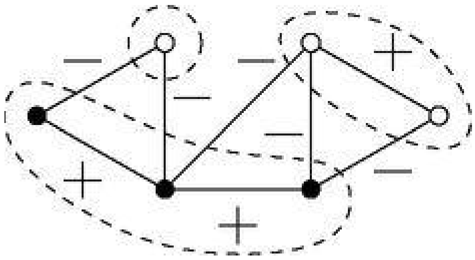


Figure 7.7: A balanced, clusterable network. Every loop in this network contains an even number of minus signs. The dotted lines indicate the division of the network into clusters such that all acquaintances within clusters have positive connections and all acquaintances in different clusters have negative connections.

Harary's theorem is straightforward to prove, and the proof is "constructive," meaning that it shows not only when a network is clusterable but also tells us what the groups are.[106] We consider initially only networks that are connected—they have just one component. In a moment we will relax this condition. We will color in the vertices of the network each in one of two colors, denoted by the open and filled circles in Fig. 7.7, for instance. We start with any vertex we please and color it with whichever color we please. Then we color in the others according to the following algorithm:

1. A vertex *v* connected by a positive edge to another *u* that has already been colored gets colored the same as *u*.
2. A vertex *v* connected by a negative edge to another *u* that has already been colored gets colored the opposite color from *u*.

For most networks it will happen in the course of this coloring process that we sometimes come upon a vertex whose color has already been assigned. When this happens there is the possibility of a conflict arising between the previously assigned color and the one that we would like to assign to it now according to the rules above. However, as we now show, this conflict only arises if the network as a whole is unbalanced.

If in coloring in a network we come upon a vertex that has already been colored in, it immediately implies that there must be another path by which that vertex can be reached from our starting point and hence that there is at least one, and possibly more than one, loop in the network to which this vertex belongs—the loop consisting of the two paths between the starting point and the vertex. Since the network is balanced, every loop to which our vertex belongs must have an even number of negative edges around it. Now let us suppose that the color already assigned to the vertex is in conflict with the one we would like to assign it now. There are two ways in which this could happen, as illustrated in Fig. 7.8. In case (a), we color in a vertex *u* and then move onto its neighbor *v*, only to find that *v* has already been colored the opposite color to *u*, even though the edge between them is positive. This presents a problem. But if *u* and *v* are opposite colors, then around any loop containing them both there must be an *odd* number of minus signs, so that the color changes an odd number of times and ends up the opposite of what it started out as. And if there is an odd number of minus signs around the loop, then the network is not balanced.

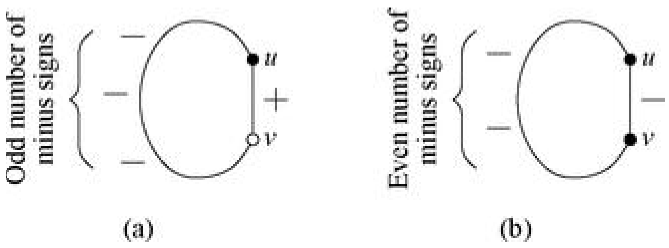


Figure 7.8: Proof that a balanced network is clusterable. If we fail to color a network in two colors as described in the text, then there must exist a loop in the network that has one or other of the two configurations shown here, both of which have an odd number of minus signs around them (counting the one between the vertices *u* and *v*), and hence the network is not balanced.

In case (b) vertices *u* and *v* have the same color but the edge between them is negative. Again we have a problem. But if *u* and *v* are the same color then there must be an *even* number of negative edges around the rest of the loop connecting them which, along with the negative edge between *u* and *v*, gives us again an odd total number of negative edges around the entire loop, and hence the network is again not balanced.

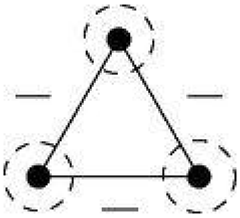
Either way, if we ever encounter a conflict about what color a vertex should have then the network must be unbalanced. If the network is balanced, therefore, we will never encounter such a conflict and we will be able to color the entire network with just two colors while obeying the rules.

Once we have colored the network in this way, we can immediately deduce the identity of the groups that satisfy Harary’s theorem: we simply divide the network into contiguous clusters of vertices that have the same color—see Fig. 7.7 again. In every such cluster, since all vertices have the same color, they must be joined by positive edges. Conversely, all edges that connected different clusters must be negative, since the clusters have different colors. (If they did not have different colors they would be considered the same cluster.)

Thus Harary′s theorem is proved and at the same time we have deduced a method for constructing the clusters.[107] It only remains to extend the proof to networks that have more than one component, but this is trivial, since we can simply repeat the proof above for each component separately.

The practical importance of Harary’s result rests on the fact that, as mentioned earlier, many real social networks are found naturally to be in a balanced or mostly balanced state. In such cases it would be possible, therefore, for the network to form into groups such that everyone likes others within their group with whom they have contact and dislikes those in other groups. It is widely assumed in social network theory that this does indeed often happen. Structural balance and clusterability in networks are thus a model for cliquishness or insularity, with people tending to stick together in like-minded groups and disdaining everyone outside their immediate community.

It is worth asking whether the inverse of Harary’s clusterability theorem is also true. Is it also the case that a network that is clusterable is necessarily balanced? The answer is no, as this simple counter-example shows:



In this network all three vertices dislike each other, so there is an odd number of minus signs around the loop, but there is no problem dividing the network into three clusters of one vertex each such that everyone dislikes the members of the other clusters. This network is clusterable but not balanced.

7.12 SIMILARITY

Another central concept in social network analysis is that of similarity between vertices. In what ways can vertices in a network be similar, and how can we quantify that similarity? Which vertices in a given network are most similar to one another? Which vertex *v* is most similar to a given vertex *u*? Answers to questions like these can help us tease apart the types and relationships of vertices in social networks, information networks, and others. For instance, one could imagine that it might be useful to have a list of web pages that are similar—in some appropriate sense—to another page that we specify. In fact, several web search engines already provide a feature like this: “Click here for pages similar to this one.”

Similarity can be determined in many different ways and most of them have nothing to do with networks. For example, commercial dating and matchmaking services try to match people with others to whom they are similar by using descriptions of people’s interests, background, likes, and dislikes. In effect, these services are computing similarity measures between people based on personal characteristics. Our focus in this book, however, is on networks, so we will concentrate on the more limited problem of determining similarity between the vertices of a network using the information contained in the network structure.

There are two fundamental approaches to constructing measures of network similarity, called *structural equivalence* and *regular equivalence*. The names are rather opaque, but the ideas they represent are simple enough. Two vertices in a network are structurally equivalent if they share many of the same network neighbors. In Fig. 7.9a we show a sketch depicting structural equivalence between two vertices *i* and *j*—the two share, in this case, three of the same neighbors, although both also have other neighbors that are not shared.

Regular equivalence is more subtle. Two regularly equivalent vertices do not necessarily share the same neighbors, but they have neighbors *who are themselves similar*. Two history students at different universities, for example, may not have any friends in common, but they can still be similar in the sense that they both know a lot of other history students, history instructors, and so forth. Similarly, two CEOs at two different companies may have no colleagues in common, but they are similar in the sense that they have professional ties to their respective CFO, CIO, members of the board, company president, and so forth. Regular equivalence is illustrated in Fig. 7.9b.

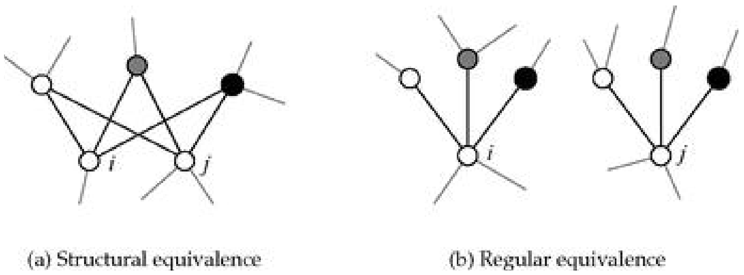


Figure 7.9: Structural equivalence and regular equivalence. (a) Vertices *i* and *j* are structurally equivalent if they share many of the same neighbors. (b) Vertices *i* and *j* are regularly equivalent if their neighbors are themselves equivalent (indicated here by the different shades of vertices).

In the next few sections we describe some mathematical measures that quantify these ideas of similarity. As we will see, measures for structural equivalence are considerably better developed than those for regular equivalence.

7.12.1 COSINE SIMILARITY

We start by looking at measures of structural equivalence and we will concentrate on undirected networks. Perhaps the simplest and most obvious measure of structural equivalence would be just a count of the number of common neighbors two vertices have. In an undirected network the number n_{ij} of common neighbors of vertices i and j is given by

$$n_{ij} = \sum_k A_{ik}A_{kj},$$

(7.46)

which is the ijth element of $\mathbf{A}^2$. This quantity is closely related to the “cocitation” measure introduced in Section 6.4.1. Cocitation is defined for directed networks whereas we are here considering undirected ones, but otherwise it is essentially the same thing.

However, a simple count of common neighbors for two vertices is not on its own a very good measure of similarity. If two vertices have three common neighbors is that a lot or a little? It’s hard to tell unless we know, for instance, what the degrees of the vertices are, or how many common neighbors other pairs of vertices share. What we need is some sort of normalization that places the similarity value on some easily understood scale. One strategy might be simply to divide by the total number of vertices in the network n, since this is the maximum number of common neighbors two vertices can have in a simple graph. (Technically the maximum is actually $n - 2$, but the difference is small when n is large.) However, this unduly penalizes vertices with low degree: if a vertex has degree three, then it can have at most three neighbors in common with another vertex, but the two vertices would still receive a small similarity value if the divisor n were very large. A better measure would allow for the varying degrees of vertices. Such a measure is the *cosine similarity*, sometimes also called *Salton’s cosine*.

In geometry, the inner or dot product of two vectors $\mathbf{x}$ and $\mathbf{y}$ is given by $\mathbf{x} \cdot \mathbf{y} = |\mathbf{x}|\ |\mathbf{y}| \cos\theta$, where $|\mathbf{x}|$ is the magnitude of $\mathbf{x}$ and θ is the angle between the two vectors. Rearranging, we can write the cosine of the angle as

$$\cos\theta = \frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}|\,|\mathbf{y}|}.$$

(7.47)

Salton [290] proposed that we regard the ith and jth rows (or columns) of the adjacency matrix as two vectors and use the cosine of the angle between them as our similarity measure. Noting that the dot product of two rows is simply $\sum_k A_{ik}A_{kj}$ for an undirected network, this gives us a similarity

$$\sigma_{ij} = \cos\theta = \frac{\sum_k A_{ik}A_{kj}}{\sqrt{\sum_k A_{ik}^2}\sqrt{\sum_k A_{jk}^2}}.$$

(7.48)

Assuming our network is an unweighted simple graph, the elements of the adjacency matrix take only the values 0 and 1, so that $A_{ij}^2 = A_{ij}$ for all *i*, *j*. Then $\sum_k A_{ik}^2 = \sum_k A_{ik} = k_i$, where k_i is the degree of vertex *i* (see Eq. (6.19)). Thus

$$\sigma_{ij} = \frac{\sum_k A_{ik}A_{kj}}{\sqrt{k_i k_j}} = \frac{n_{ij}}{\sqrt{k_i k_j}}.$$

(7.49)

The cosine similarity of *i* and *j* is therefore the number of common neighbors of the two vertices divided by the geometric mean of their degrees. For the vertices *i* and *j* depicted in Fig. 7.9a, for instance, the cosine similarity would be

$$\sigma_{ij} = \frac{3}{\sqrt{4 \times 5}} = 0.671\ldots$$

(7.50)

Notice that the cosine similarity is technically undefined if one or both of the vertices has degree zero, but by convention we normally say in that case that $\sigma_{ij} = 0$.

The cosine similarity provides a natural scale for our similarity measure. Its value always lies in the range from 0 to 1. A cosine similarity of 1 indicates that two vertices have exactly the same neighbors. A cosine similarity of zero indicates that they have none of the same neighbors. Notice that the cosine similarity can never be negative, being a sum of positive terms, even though cosines in general can of course be negative.

7.12.2 PEARSON COEFFICIENTS

An alternative way to normalize the count of common neighbors is to compare it with the expected value that count would take on a network in which vertices choose their neighbors at random. This line of argument leads us to the *Pearson correlation coefficient*.

Suppose vertices i and j have degrees k_i and k_j respectively. How many common neighbors should we expect them to have? This is straightforward to calculate if they choose their neighbors purely at random. Imagine that vertex i chooses k_i neighbors uniformly at random from the n possibilities open to it (or $n - 1$ on a network without self-loops, but the distinction is slight for a large network), and vertex j similarly chooses k_j neighbors at random. For the first neighbor that j chooses there is a probability of k_i/n that it will choose one of the ones k_i chose, and similarly for each succeeding choice. (We neglect the possibility of choosing the same neighbor twice, since it is small for a large network.) Then in total the expected number of common neighbors between the two vertices will be k_j times this, or k_ik_j/n.

A reasonable measure of similarity between two vertices is the actual number of common neighbors they have minus the expected number that they *would* have if they chose their neighbors at random:

$$\begin{aligned}
\sum_k A_{ik}A_{jk} - \frac{k_ik_j}{n} &= \sum_k A_{ik}A_{jk} - \frac{1}{n}\sum_k A_{ik}\sum_l A_{jl} \\
&= \sum_k A_{ik}A_{jk} - n\langle A_i\rangle\langle A_j\rangle \\
&= \sum_k [A_{ik}A_{jk} - \langle A_i\rangle\langle A_j\rangle] \\
&= \sum_k (A_{ik} - \langle A_i\rangle)(A_{jk} - \langle A_j\rangle),
\end{aligned}$$

(7.51)

where A_i denotes the mean $n^{-1}\sum_k A_{ik}$ of the elements of the ith row of the adjacency matrix. Equation (7.51) will be zero if the number of common neighbors of i and j is exactly what we would expect on the basis of random chance. If it is positive, then i and j have more neighbors than we would expect by chance, which we take as an indication of similarity between the two. Equation (7.51) can also be negative, indicating that i and j have fewer neighbors than we would expect, a possible sign of dissimilarity.

Equation (7.51) is simply n times the covariance $\text{cov}(A_i, A_j)$ of the two rows of the adjacency matrix. It is common to normalize the covariance, as we did with the cosine similarity, so that its maximum value is 1. The maximum value of the covariance of any two sets of quantities occurs when the sets are exactly the same, in which case their covariance is just equal to the variance of either set, which we could write as σ_i^2 or σ_j^2, or in symmetric form as $\sigma_i\sigma_j$. Normalizing by this quantity then gives us the standard Pearson correlation coefficient:

$$r_{ij} = \frac{\mathrm{cov}(A_i, A_j)}{\sigma_i \sigma_j} = \frac{\sum_k (A_{ik} - \langle A_i \rangle)(A_{jk} - \langle A_j \rangle)}{\sqrt{\sum_k (A_{ik} - \langle A_i \rangle)^2} \sqrt{\sum_k (A_{jk} - \langle A_j \rangle)^2}} .$$

(7.52)

This quantity lies strictly in the range $-1 \leq r_{ij} \leq 1$.

The Pearson coefficient is a widely used measure of similarity. It allows us to say when vertices are both similar or dissimilar compared with what we would expect if connections in the network were formed at random.

7.12.3 OTHER MEASURES OF STRUCTURAL EQUIVALENCE

There are many other possible measures of structural equivalence. For instance, one could also normalize the number n_{ij} of common neighbors by dividing by (rather than subtracting) the expected value of k_ik_j/n. That would give us a similarity of

$$\frac{n_{ij}}{k_ik_j/n} = n\frac{\sum_k A_{ik}A_{jk}}{\sum_k A_{ik}\sum_k A_{jk}}.$$

(7.53)

This quantity will be 1 if the number of common neighbors is exactly as expected on the basis of chance, greater than one if there are more common neighbors than that, and less than one for dissimilar vertices with fewer common neighbors than we would expect by chance. It is never negative and has the nice property that it is zero when the vertices in question have no common neighbors. This measure could be looked upon as an alternative to the cosine similarity: the two differ in that one has the product of the degrees k_ik_j in the denominator while the other has the square root of the product $\sqrt{k_ik_j}$. It has been suggested that Eq. (7.53) may in some cases be a superior measure to the cosine similarity because, by normalizing with respect to the expected number of common neighbors rather than the maximum number, it allows us to easily identify statistically surprising coincidences between the neighborhoods of vertices, which cosine similarity does not [195].

Another measure of structural equivalence is the so-called *Euclidean distance* ,[108] which is equal to the number of neighbors that differ between two vertices. That is, it is the number of vertices that are neighbors of *i* but not of *j*, or vice versa. Euclidean distance is really a dissimilarity measure, since it is larger for vertices that differ more.

In terms of the adjacency matrix the Euclidean distance d_{ij} between two vertices can be written

$$d_{ij} = \sum_k (A_{ik} - A_{jk})^2.$$

(7.54)

As with our other measures it is sometimes convenient to normalize the Euclidean distance by dividing by its possible maximum value. The maximum value of d_{ij} occurs when two vertices have no neighbors in common, in which case the distance is equal to the sum of the degrees of the vertices: $d_{ij} = k_i + k_j$. Dividing by this maximum value the normalized distance is

$$\frac{\sum_k (A_{ik} - A_{jk})^2}{k_i + k_j} = \frac{\sum_k (A_{ik} + A_{jk} - 2A_{ik}A_{jk})}{k_i + k_j} = 1 - 2\frac{n_{ij}}{k_i + k_j},$$

(7.55)

where we have made use of the fact that $A_{ij}^2 = A_{ij}$ because A_{ij} is always zero or one, and n_{ij} is again the number of neighbors that i and j have in common. To within additive and multiplicative constants, this normalized Euclidean distance can thus be regarded as just another alternative normalization of the number of common neighbors.

7.12.4 REGULAR EQUIVALENCE

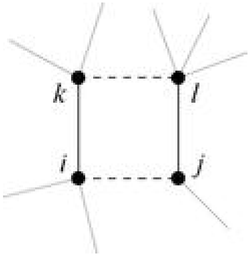
The similarity measures discussed in the preceding sections are all measures of structural equivalence, i.e., they are measures of the extent to which two vertices share the same neighbors. The other main type of similarity considered in social network analysis is regular equivalence. As described above, regularly equivalent vertices are vertices that, while they do not necessarily share neighbors, have neighbors who are themselves similar—see Fig. 7.9b again.

Quantitative measures of regular equivalence are less well developed than measures of structural equivalence. In the 1970s social network analysts came up with some rather complicated computer algorithms, such as the "REGE" algorithm of White and Reitz [320, 327], that were intended to discover regular equivalence in networks, but the operation of these algorithms is involved and not easy to interpret. More recently, however, some simpler algebraic measures have been developed that appear to work reasonably well. The basic idea [45, 162, 195] is to define a similarity score σ_{ij} such that i and j have high similarity if they have neighbors k and l that themselves have high similarity. For an undirected network we can write this as

$$\sigma_{ij} = \alpha \sum_{kl} A_{ik} A_{jl} \sigma_{kl},$$

(7.56)

or in matrix terms $\boldsymbol{\sigma} = \alpha \mathbf{A} \boldsymbol{\sigma} \mathbf{A}$. Although it may not be immediately clear, this expression is a type of eigenvector equation, where the entire matrix $\boldsymbol{\sigma}$ of similarities is the eigenvector. The parameter α is the eigenvalue (or more correctly, its inverse) and, as with the eigenvector centrality of Section 7.2, we are normally interested in the leading eigenvalue, which can be found by standard methods.



Vertices i and j are considered similar (dashed line) if they have respective neighbors k and l that are themselves similar.

See Section 11.1 for a discussion of computer algorithms for finding eigenvectors.

This formula however has some problems. First, it doesn't necessarily give a high value for the "self-similarity" σ_{ii} of a vertex to itself, which is counter-intuitive. Presumably, all vertices are highly similar to themselves! As a consequence of this, Eq. (7.56) also doesn't necessarily give a high similarity score to vertex pairs that have a lot of common neighbors, which in the light of our examination of structural equivalence in the preceding few sections we perhaps feel it should. If we had high self-similarity scores for all vertices, on the other hand, then Eq. (7.56) would automatically give high similarity also to vertices with many common neighbors.

We can fix these problems by introducing an extra diagonal term in the similarity thus:

$$\sigma_{ij} = \alpha \sum_{kl} A_{ik} A_{jl} \sigma_{kl} + \delta_{ij},$$

(7.57)

or in matrix notation

$$\boldsymbol{\sigma} = \alpha \mathbf{A} \boldsymbol{\sigma} \mathbf{A} + \mathbf{I}.$$

(7.58)

However, while expressions like this have been proposed as similarity measures, they still suffer from some problems. Suppose we evaluate Eq. (7.58) by repeated iteration, taking a starting value, for example, of $\boldsymbol{\sigma}^{(0)} = 0$ and using it to compute $\boldsymbol{\sigma}^{(1)} = \alpha \mathbf{A} \boldsymbol{\sigma} \mathbf{A} + \mathbf{I}$, and then repeating the process many times until σ converges. On the first few iterations we will get the following results:

$$\boldsymbol{\sigma}^{(1)} = \mathbf{I},$$

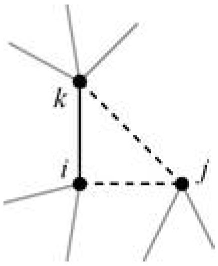
(7.59a)

$$\boldsymbol{\sigma}^{(2)} = \alpha \mathbf{A}^2 + \mathbf{I},$$

(7.59b)

$$\boldsymbol{\sigma}^{(3)} = \alpha^2 \mathbf{A}^4 + \alpha \mathbf{A}^2 + \mathbf{I}.$$

(7.59c)



In the modified definition of regular equivalence vertex *i* is considered similar to vertex *j* (dashed line) if it has a neighbor *k* that is itself similar to *j*.

The pattern is clear: in the limit of many iterations, we will get a sum over even powers of the adjacency matrix. However, as discussed in Section 6.10, the elements of the *r*th power of the adjacency matrix count paths of length *r* between vertices, and hence this measure of similarity is a weighted sum over the numbers of paths of even length between pairs of vertices.

But why should we consider only paths of even length? Why not consider paths of all lengths? These questions lead us to a better definition of regular equivalence as follows: vertices *i* and *j* are similar if *i* has a neighbor *k* that is itself similar to *j*.[109] Again we assume that vertices are similar to themselves, which we can represent with a diagonal δ_{ij} term in the similarity, and our similarity measure then looks like

$$\sigma_{ij} = \alpha \sum_k A_{ik}\sigma_{kj} + \delta_{ij},$$

(7.60)

or

$$\boldsymbol{\sigma} = \alpha \mathbf{A}\boldsymbol{\sigma} + \mathbf{I},$$

(7.61)

in matrix notation. Evaluating this expression by iterating again starting from $\boldsymbol{\sigma}^{(0)} = 0$, we get

$$\boldsymbol{\sigma}^{(1)} = \mathbf{I},$$

(7.62a)

$$\sigma^{(2)} = \alpha\mathbf{A} + \mathbf{I},$$

(7.62b)

$$\sigma^{(3)} = \alpha^2\mathbf{A}^2 + \alpha\mathbf{A} + \mathbf{I}.$$

(7.62c)

In the limit of a large number of iterations this gives

$$\sigma = \sum_{m=0}^{\infty}(\alpha\mathbf{A})^m = (\mathbf{I} - \alpha\mathbf{A})^{-1},$$

(7.63)

which we could also have deduced directly by rearranging Eq. (7.61). Now our similarity measure includes counts of paths at all lengths, not just even paths. In fact, we can see now that this similarity measure could be defined a completely different way, as a weighted count of all the paths between the vertices i and j with paths of length r getting weight α^r. So long as $\alpha < 1$, longer paths will get less weight than shorter ones, which seems sensible: in effect we are saying that vertices are similar if they are connected either by a few short paths or by very many long ones.

Equation (7.63) is reminiscent of the formula for the Katz centrality, Eq. (7.10). We could call Eq. (7.63) the "Katz similarity" perhaps, although Katz himself never discussed it. The Katz centrality of a vertex would then be simply the sum of the Katz similarities of that vertex to all others. Vertices that are similar to many others would get high centrality, a concept that certainly makes intuitive sense. As with the Katz centrality, the value of the parameter α is undetermined—we are free to choose it as we see fit—but it must satisfy $\alpha < 1/\kappa_1$ if the sum in Eq. (7.63) is to converge, where κ_1 is the largest eigenvalue of the adjacency matrix.

In a sense, this regular equivalence measure can be seen as a generalization of our structural equivalence measures in earlier sections. With those measures we were counting the common neighbors of a pair of vertices, but the number of common neighbors is also of course the number of paths of length two between the vertices. Our "Katz similarity" measure merely extends this concept to counting paths of all lengths.

Some variations of this similarity measure are possible. As defined it tends to give high similarity to vertices that have high degree, because if a vertex has many neighbors it tends to increase the number of those neighbors that are similar to any other given vertex and hence increases the total similarity to that vertex. In some cases this might be desirable: maybe the person with many friends *should* be considered more similar to others than the person with few. However, in other cases it gives an unwanted bias in favor of high-degree nodes. Who is to say that two hermits are not "similar" in an interesting sense? If we wish, we can remove the bias in favor of high degree by dividing by vertex degree thus:

$$\sigma_{ij} = \frac{\alpha}{k_i} \sum_k A_{ik}\sigma_{kj} + \delta_{ij},$$

(7.64)

or in matrix notation $\boldsymbol{\sigma} = \alpha\mathbf{D}^{-1}\mathbf{A}\boldsymbol{\sigma} + \mathbf{I}$, where, as previously, **D** is the diagonal matrix with elements $D_{ii} = k_i$. This expression can be rearranged to read:

$$\sigma = (\mathbf{I} - \alpha\mathbf{D}^{-1}\mathbf{A})^{-1} = (\mathbf{D} - \alpha\mathbf{A})^{-1}\mathbf{D}.$$

(7.65)

Another useful variant is to consider cases where the last term in Eqs. (7.60) or (7.64) is not simply diagonal, but includes off-diagonal terms too. Such a generalization would allow us to specify explicitly that particular pairs of vertices are similar, based on some other (probably non-network) information that we have at our disposal. Going back to the example of CEOs at companies that we gave at the beginning of Section 7.12, we might, for example, want to state explicitly that the CFOs and CIOs and so forth at different companies are similar, and then our similarity measure would, we hope, correctly deduce from the network structure that the CEOs are similar also. This kind of approach is particularly useful in the case of networks that consist of more than one component, so that some pairs of vertices are not connected at all. If, for instance, we have two separate components representing people in two different companies, then there will be no paths of any length between individuals in different companies, and hence a measure like (7.60) or (7.64) will never assign a non-zero similarity to such individuals. If however, we explicitly insert some similarities between members of the different companies, our measure will then be able to generalize and extend those inputs to deduce similarities between other members.

This idea of generalizing from a few given similarities arises in other contexts too. For example, in the fields of machine learning and information retrieval there is a considerable literature on how to generalize known similarities between a subset of the objects in a collection of, say, text documents to the rest of the collection, based on network data or other information.

7.13 HOMOPHILY AND ASSORTATIVE MIXING

Consider Fig. 7.10, which shows a friendship network of children at an American school, determined from a questionnaire of the type discussed in Section 3.2.[111] One very clear feature that emerges from the figure is the division of the network into two groups. It turns out that this division is principally along lines of race. The different shades of the vertices in the picture correspond to students of different race as denoted in the legend, and reveal that the school is sharply divided between a group composed principally of black children and a group composed principally of white.

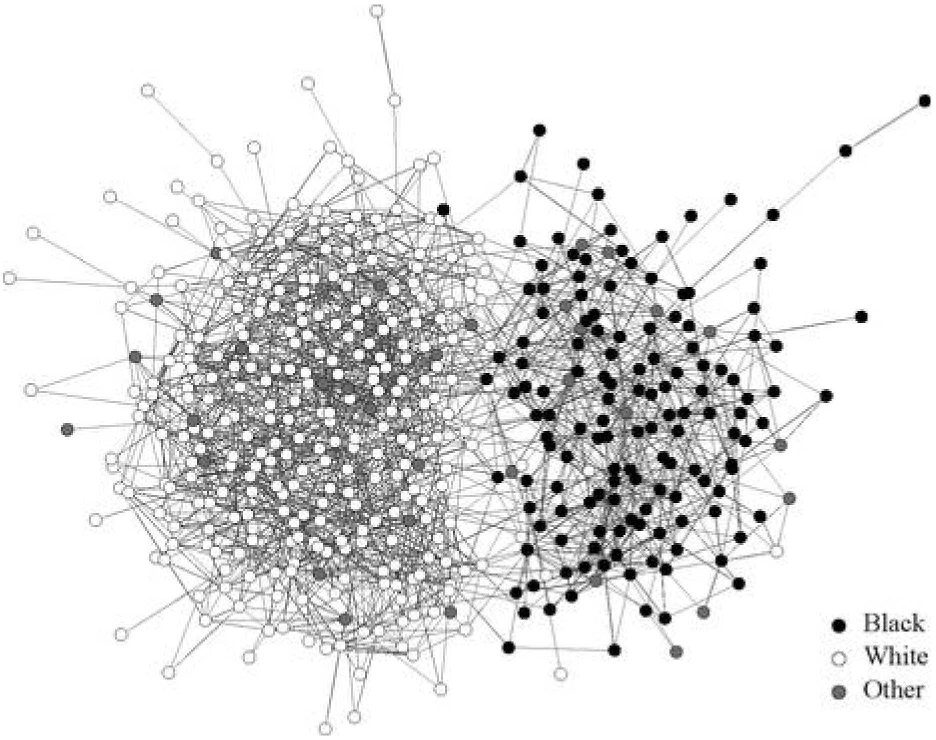


Figure 7.10: Friendship network at a US high school. The vertices in this network represent 470 students at a US high school (ages 14 to 18 years). The vertices are color coded by race as indicated in the key. Data from the National Longitudinal Study of Adolescent Health [34, 314].

This is not news to sociologists, who have long observed and discussed such divisions [225]. Nor is the effect specific to race. People are found to form friendships, acquaintances, business relations, and many other types of tie based on all sorts of characteristics, including age, nationality, language, income, educational level, and many others. Almost any social parameter you can imagine plays into people's selection of their friends. People have, it appears, a strong tendency to associate with others whom they perceive as being similar to themselves in some way. This tendency is called *homophily* or *assortative mixing*.

More rarely, one also encounters *disassortative mixing*, the tendency for people to associate with others who are *unlike* them. Probably the most widespread and familiar example of disassortative mixing is mixing by gender in sexual contact networks. The majority of sexual partnerships are between individuals of opposite sex, so they represent connections between people who differ in their gender. Of course, same-sex partnerships do also occur, but they are a much smaller fraction of the ties in the network.

Assortative (or disassortative) mixing is also seen in some nonsocial networks. Papers in a citation network, for instance, tend to cite other papers in the same field more than they do papers in different fields. Web pages written in a particular language tend to link to others in the same language.

In this section we look at how assortative mixing can be quantified. Assortative mixing by discrete characteristics such as race, gender, or nationality is fundamentally different from mixing by a scalar characteristic like age or income, so we treat the two cases separately.

7.13.1 ASSORTATIVE MIXING BY ENUMERATIVE CHARACTERISTICS

Suppose we have a network in which the vertices are classified according to some characteristic that has a finite set of possible values. The values are merely enumerative—they don't fall in any particular order. For instance, the vertices could represent people and be classified according to nationality, race, or gender. Or they could be web pages classified by what language they are written in, or biological species classified by habitat, or any of many other possibilities.

The network is assortative if a significant fraction of the edges in the network run between vertices of the same type, and a simple way to quantify assortativity would be to measure that fraction. However, this is not a very good measure because, for instance, it is 1 if all vertices belong to the same single type. This is a trivial sort of assortativity: all friends of a human being, for example, are also human beings,[112] but this is not really an interesting statement. What we would like instead is a measure that is large in non-trivial cases but small in trivial ones.

A good measure turns out to be the following. We find the fraction of edges that run between vertices of the same type, and then we subtract from that figure the fraction of such edges we would *expect* to find if edges were positioned at random without regard for vertex type. For the trivial case in which all vertices are of a single type, for instance, 100% of edges run between vertices of the same type, but this is also the expected figure, since there is nowhere else for the edges to fall. The difference of the two numbers is then zero, telling us that there is no non-trivial assortativity in this case. Only when the fraction of edges between vertices of the same type is significantly greater than we would expect on the basis of chance will our measure give a positive score.

In mathematical terms, let us denote by c_i the class or type of vertex i, which is an integer 1 ... n_c , with n_c being the total number of classes. Then the total number of edges that run between vertices of the same type is

$$\sum_{\text{edges }(i,j)} \delta(c_i, c_j) = \tfrac{1}{2}\sum_{ij} A_{ij}\,\delta(c_i, c_j),$$

(7.66)

where $\delta(m, n)$ is the Kronecker delta and the factor of $\frac{1}{2}$ accounts for the fact that every vertex pair i, j is counted twice in the second sum.

Calculating the expected number of edges between vertices if edges are placed at random takes a little more work. Consider a particular edge attached to vertex i, which has degree k_i. There are by definition $2m$ ends of edges in the entire network, where m is as usual the total number of edges, and the chances that the other end of our particular edge is one of the k_j ends attached to vertex j is thus $k_j/2m$ if connections are made purely at random.[113] Counting all k_i edges attached to i, the total expected number of edges between vertices i and j is then $k_ik_j/2m$, and the expected number of edges between all pairs of vertices of the same type is

$$\tfrac{1}{2}\sum_{ij} \frac{k_ik_j}{2m}\,\delta(c_i, c_j),$$

(7.67)

where the factor of $\frac{1}{2}$, as before, prevents us from double-counting vertex pairs. Taking the difference of (7.66) and (7.67) then gives us an expression for the difference between the actual and expected number of edges in the network that join vertices of like types:

$$\tfrac{1}{2}\sum_{ij} A_{ij}\,\delta(c_i,c_j) - \tfrac{1}{2}\sum_{ij} \frac{k_i k_j}{2m}\,\delta(c_i,c_j) = \tfrac{1}{2}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(c_i,c_j).$$

(7.68)

Conventionally, one calculates not the number of such edges but the fraction, which is given by this same expression divided by the number m of edges:

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(c_i,c_j).$$

(7.69)

This quantity Q is called the *modularity* [239, 250] and is a measure of the extent to which like is connected to like in a network. It is strictly less than 1, takes positive values if there are more edges between vertices of the same type than we would expect by chance, and negative ones if there are less.

For Fig. 7.10, for instance, where the types are the three ethnic classifications “black,” “white,” and “other,” we find a modularity value of Q = 0.305, indicating (positive) assortative mixing by race in this particular network.[114] Negative values of the modularity indicate disassortative mixing. We might see a negative modularity, for example, in a network of sexual partnerships where most partnerships were between individuals of opposite sex.

The quantity

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

(7.70)

in Eq. (7.69) appears in a number of situations in the study of networks. We will encounter it, for instance, in Section 11.8 when we study community detection in networks. In some contexts it is useful to consider B_{ij} to be an element of a matrix **B**, which itself is called the *modularity matrix*.

The modularity, Eq. (7.69), is always less than 1 but in general it does not achieve the value Q = 1 even for a perfectly mixed network, one in which every vertex is connected only to others of the same type. Depending on the sizes of the groups and the degrees of vertices, the maximum value of Q can be considerably less than 1. This is in some ways unsatisfactory: how is one to know when one has strong assortative mixing and when one doesn’t? To rectify the problem, we can normalize Q by dividing by its value for the perfectly mixed network. With perfect mixing all edges fall between vertices of the same type and hence $\delta(c_i,c_j) = 1$ whenever $A_{ij} = 1$. This means that the first term in the sum in Eq. (7.69) sums to $2m$ and the modularity for the perfectly mixed network is

$$Q_{\max} = \frac{1}{2m}\left(2m - \sum_{ij}\frac{k_i k_j}{2m}\delta(c_i,c_j)\right).$$

(7.71)

Then the normalized value of the modularity is given by

$$\frac{Q}{Q_{\max}} = \frac{\sum_{ij}(A_{ij} - k_i k_j/2m)\delta(c_i,c_j)}{2m - \sum_{ij}(k_i k_j/2m)\delta(c_i,c_j)}.$$

(7.72)

This quantity, sometimes called an *assortativity coefficient*, now takes a maximum value of 1 on a perfectly mixed network.

Although it can be a useful measure in some circumstances, however, Eq. (7.72) is only rarely used. Most often, the modularity is used in its unnormalized form, Eq. (7.69).

An alternative form for the modularity, which is sometimes useful in practical situations, can be derived in terms of the quantities

$$e_{rs} = \frac{1}{2m}\sum_{ij} A_{ij}\,\delta(c_i,r)\,\delta(c_j,s),$$

(7.73)

which is the fraction of edges that join vertices of type r to vertices of type s, and

$$a_r = \frac{1}{2m}\sum_i k_i\,\delta(c_i,r),$$

(7.74)

which is the fraction of ends of edges attached to vertices of type *r*. Then, noting that

$$\delta(c_i, c_j) = \sum_r \delta(c_i, r)\delta(c_j, r),$$

(7.75)

we have, from Eq. (7.69)

$$\begin{aligned} Q &= \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\sum_r \delta(c_i, r)\delta(c_j, r) \\ &= \sum_r \left[\frac{1}{2m}\sum_{ij} A_{ij}\,\delta(c_i, r)\delta(c_j, r) - \frac{1}{2m}\sum_i k_i\,\delta(c_i, r)\frac{1}{2m}\sum_j k_j \delta(c_j, r)\right] \\ &= \sum_r (e_{rr} - a_r^2). \end{aligned}$$

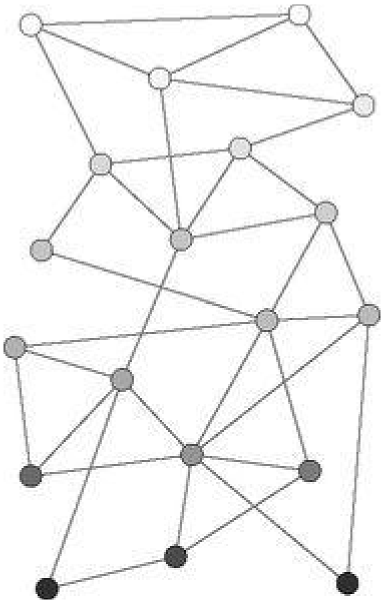
(7.76)

This form can be useful, for instance, when we have network data in the form of a list of edges and the types of the vertices at their ends, but no explicit data on vertex degrees. In such a case e_{rs} and a_r are relatively easy to calculate, while Eq. (7.69) is quite awkward.

7.13.2 ASSORTATIVE MIXING BY SCALAR CHARACTERISTICS

We can also have homophily in a network according to scalar characteristics like age or income. These are characteristics whose values come in a particular order, so that it is possible say not only when two vertices are exactly the same according to the characteristic but also when they are approximately the same. For instance, while two people can certainly be of exactly the same age—born on the same day even—they can also be approximately the same age—born within a couple of years of one another, say—and people could (and in fact often do) choose who they associate with on the basis of such approximate ages. There is no equivalent approximate similarity for the enumerative characteristics of the previous section: there is no sense in which people from France and Germany, say, are more nearly of the same nationality than people from France and Spain.[115]

If network vertices with similar values of a scalar characteristic tend to be connected together more often that those with different values then the network is considered assortatively mixed according to that characteristic. If, for example, people are friends with others around the same age as them, then the network is assortatively mixed by age. Sometimes you may also hear it said that the network is *stratified* by age, which means the same thing—one can think of age as a one-dimensional scale or axis, with individuals of different ages forming connected “strata” within the network.



A sketch of stratified network in which most connections run between vertices at or near the same “level” in the network, with level along the vertical axis in this case and also denoted by the shades of the vertices.

Consider Fig. 7.11, which shows friendship data for the same set of US schoolchildren as Fig. 7.10 but now as a function of age. Each dot in the figure corresponds to one pair of friends and the position of the dot along the two axes gives the ages of the friends, with ages measured by school grades.[116] As the figure shows, there is substantial assortative mixing by age among the students: many dots lie within the boxes close to the diagonal line that represent friendships between students in the same grade. There is also, in this case, a notable tendency for students to have more friends of a wider range of ages as their age increases so there is a lower density of points in the top right box than in the lower left one.

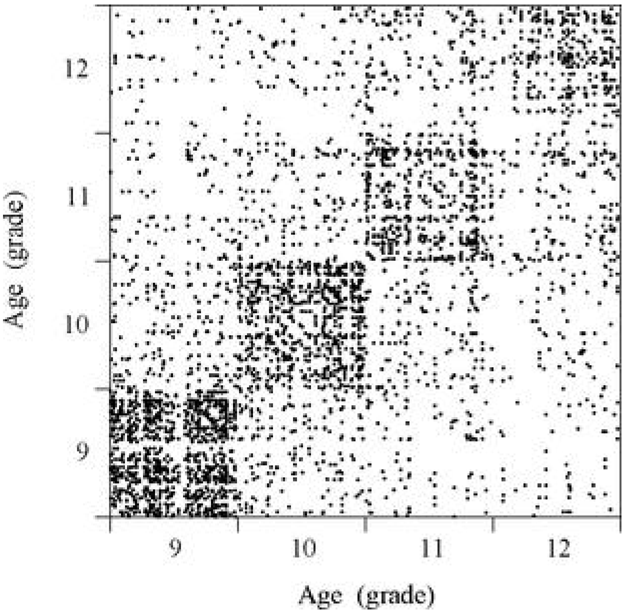


Figure 7.11: Ages of pairs of friends in high school. In this scatter plot each dot corresponds to one of the edges in Fig. 7.10, and its position along the horizontal and vertical axes gives the ages of the two individuals at either end of that edge. The ages are measured in terms of the grades of the students, which run from 9 to 12. In fact, grades in the US school system don't correspond precisely to age since students can start or end their high-school careers early or late, and can repeat grades. (Each student is positioned at random within the interval representing their grade, so as to spread the points out on the plot. Note also that each friendship appears twice, above and below the diagonal.)

One could make a crude measure of assortative mixing by scalar characteristics by adapting the ideas of the previous section. One could group the vertices into bins according to the characteristic of interest (say age) and then treat the bins as separate "types" of vertex in the sense of Section 7.13.1. For instance, we might group people by age in ranges of one year or ten years. This however misses much of the point about scalar characteristics, since it considers vertices falling in the same bin to be of identical types when they may be only approximately so, and vertices falling in different bins to be entirely different when in fact they may be quite similar.

A better approach is to use a covariance measure as follows. Let x_i be the value for vertex i of the scalar quantity (age, income, etc.) that we are interested in. Consider the pairs of values (x_i, x_j) for the vertices at the ends of each edge (i, j) in the network and let us calculate their covariance over all edges as follows. We define the mean μ of the value of x_i at the end of an edge thus:

$$\mu = \frac{\sum_{ij} A_{ij} x_i}{\sum_{ij} A_{ij}} = \frac{\sum_i k_i x_i}{\sum_i k_i} = \frac{1}{2m} \sum_i k_i x_i .$$

(7.77)

Note that this is not simply the mean value of x_i averaged over all vertices. It is an average over edges, and since a vertex with degree k_i lies at the ends of k_i edges it appears k_i times in the average (hence the factor of k_i in the sum).

Then the covariance of x_i and x_j over edges is

$$\begin{aligned}
\operatorname{cov}(x_i, x_j) &= \frac{\sum_{ij} A_{ij}(x_i - \mu)(x_j - \mu)}{\sum_{ij} A_{ij}} \\
&= \frac{1}{2m} \sum_{ij} A_{ij}(x_i x_j - \mu x_i - \mu x_j + \mu^2) \\
&= \frac{1}{2m} \sum_{ij} A_{ij} x_i x_j - \mu^2 \\
&= \frac{1}{2m} \sum_{ij} A_{ij} x_i x_j - \frac{1}{(2m)^2} \sum_{ij} k_i k_j x_i x_j \\
&= \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) x_i x_j,
\end{aligned}$$

(7.78)

where we have made use of Eqs. (6.21) and (7.77). Note the strong similarity between this expression and Eq. (7.69) for the modularity—only the delta function $\delta(c_i , c_j)$ in (7.69) has changed, being replaced by $x_i x_j$.

The covariance will be positive if, on balance, values x_i, x_j at either end of an edge tend to be both large or both small and negative if they tend to vary in opposite directions. In other words, the covariance will be positive when we have assortative mixing and negative for disassortative mixing.

Just as with the modularity measure of Section 7.13.1, it is sometimes convenient to normalize the covariance so that it takes the value 1 in a perfectly mixed network—one in which all edges fall between vertices with precisely equal values of x_i (although in most cases such an occurrence would be extremely unlikely in practice). Putting $x_j = x_i$ in Eq. (7.78) gives a perfect mixing value of

$$\frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) x_i^2 = \frac{1}{2m} \sum_{ij} \left(k_i \delta_{ij} - \frac{k_i k_j}{2m} \right) x_i x_j,$$

(7.79)

and the normalized measure, sometimes called an *assortativity coefficient*, is the ratio of the two:

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

(7.80)

Although it may not be immediately obvious, this is in fact an example of a (Pearson) correlation coefficient, having a covariance in its numerator and a variance in the denominator. We encountered another example in a different context in Section 7.12.2. The correlation coefficient varies in value between a maximum of 1 for a perfectly assortative network and a minimum of −1 for a perfectly disassortative one. A value of zero implies that the values of x_i at the ends of edges are uncorrelated.[117]

For the data of Fig. 7.11 the correlation coefficient is found to take a value of $r = 0.616$, indicating that the student friendship network has significant assortative mixing by age—students tend to be friends with others who have ages close to theirs.

It would be possible in principle also to have assortative (or disassortative) mixing according to vector characteristics, with vertices whose vectors have similar values, as measured by some appropriate metric, being more (or less) likely to be connected by an edge. One example of such mixing is the formation of friendships between individuals according to their geographic locations, location being specified by a two-dimensional vector of, for example, latitude/longitude coordinates. It is certainly the case that in general people tend to be friends with others who live geographically close to them, so one would expect mixing of this type to be assortative. Formal treatments of vector assortative mixing, however, have not been much pursued in the network literature so far.

7.13.3 ASSORTATIVE MIXING BY DEGREE

A special case of assortative mixing according to a scalar quantity, and one of particular interest, is that of mixing by degree. In a network that shows assortative mixing by degree the high-degree vertices will be preferentially connected to other high-degree vertices, and the low to low. In a social network, for example, we have assortative mixing by degree if the gregarious people are friends with other gregarious people and the hermits with other hermits. Conversely, we could have disassortative mixing by degree, which would mean that the gregarious people were hanging out with hermits and vice versa.

The reason this particular case is interesting is because, unlike age or income, degree is itself a property of the network structure. Having one structural property (the degrees) dictate another (the positions of the edges) gives rise to some interesting features in networks. In particular, in an assortative network, where the high-degree nodes tend to stick together, one expects to get a clump or *core* of such high-degree nodes in the network surrounded by a less dense *periphery* of nodes with lower-degree. This *core/periphery structure* is a common feature of social networks, many of which are found to be assortatively mixed by degree. Figure 7.12a shows a small assortatively mixed network in which the core/periphery structure is clearly visible.

On the other hand, if a network is disassortatively mixed by degree then high-degree vertices tend to connected to low-degree ones, creating star-like features in the network that are often readily visible. Figure 7.12b shows an example of a small disassortative network. Disassortatively networks do not usually have a core/periphery split but are instead more uniform.

Assortative mixing by degree can be measured in the same way as mixing according to any other scalar quantity. We define a covariance of the type described by Eq. (7.78), but with x_i now equal to the degree k_i:

$$\operatorname{cov}(k_i, k_j) = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) k_i k_j,$$

(7.81)

or if we wish we can normalize by the maximum value of the covariance to get a correlation coefficient or assortativity coefficient:

$$r = \frac{\sum_{ij} (A_{ij} - k_i k_j / 2m) k_i k_j}{\sum_{ij} (k_i \delta_{ij} - k_i k_j / 2m) k_i k_j}.$$

(7.82)

We give examples of the application of this formula to a number of networks in Section 8.7.

One point to notice is that the evaluation of Eq. (7.81) or Eq. (7.82) requires only the structure of the network and no other information (unlike the calculations for other forms of assortative mixing). Once we know the adjacency matrix (and hence the degrees) of all vertices we can calculate *r*. Perhaps for this reason mixing by degree is one of the most frequently studied types of assortative mixing.

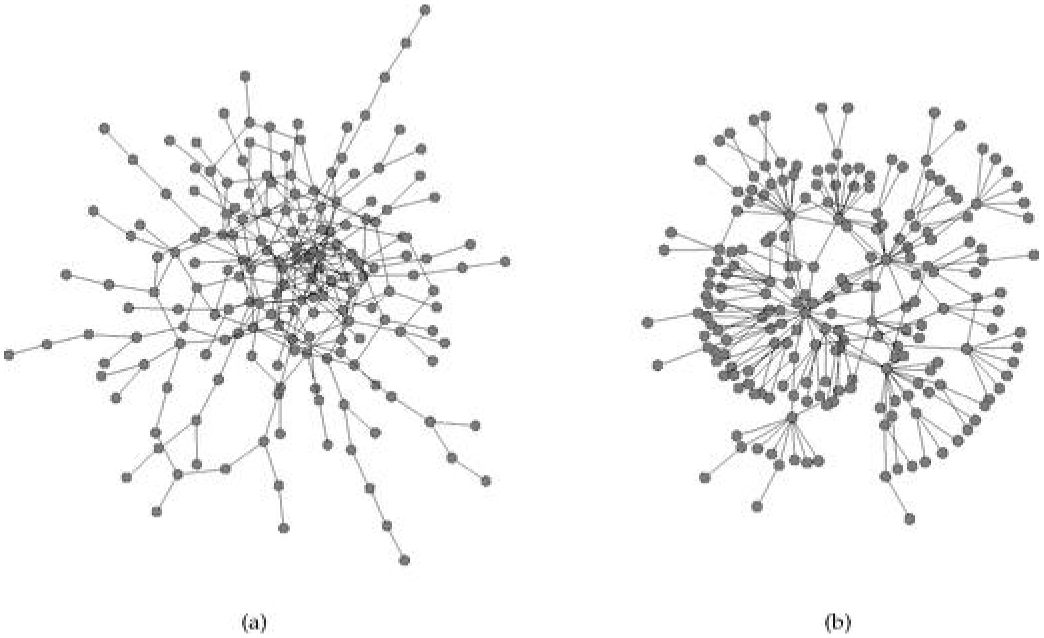


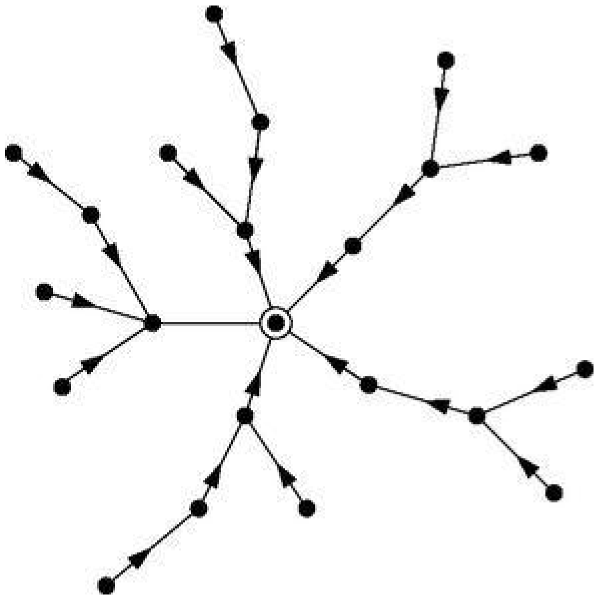
Figure 7.12: Assortative and disassortative networks. These two small networks are not real networks—they were computer generated to display the phenomenon of assortativity by degree. (a) A network that is assortative by degree, displaying the characteristic dense core of high-degree vertices surrounded by a periphery of lower-degree ones. (b) A disassortative network, displaying the star-like structures characteristic of this case. Figure from Newman and Girvan [249]. Copyright 2003 Springer-Verlag Berlin Heidelberg. Reproduced with kind permission of Springer Science and Business Media.

PROBLEMS

7.1 Consider a k-regular undirected network (i.e., a network in which every vertex has degree k).

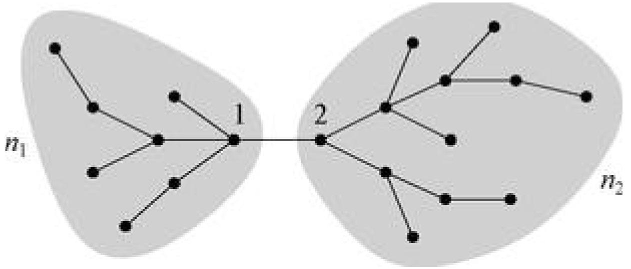
a. Show that the vector **1** = (1, 1, 1, ...) is an eigenvector of the adjacency matrix with eigenvalue k.
b. By making use of the fact that eigenvectors are orthogonal (or otherwise), show that there is no other eigenvector that has all elements positive. The Perron- Frobenius theorem says that the eigenvector with the largest eigenvalue always has all elements non-negative (see footnote 2 on page 346), and hence the eigenvector **1** gives, by definition, the eigenvector centrality of our k-regular network and the centralities are the same for every vertex.
c. Find the Katz centralities of all vertices in a k-regular network.
d. You should have found that, as with the eigenvector centrality, the Katz centralities of all vertices in the network are the same. Name a centrality measure that could give different centrality values for different vertices in a regular network.

7.2 Suppose a directed network takes the form of a tree with all edges pointing inward towards a central vertex:



What is the PageRank centrality of the central vertex in terms of the single parameter α appearing in the definition of PageRank and the geodesic distances d_i from each vertex i to the central vertex?

7.3 Consider an undirected tree of n vertices. A particular edge in the tree joins vertices 1 and 2 and divides the tree into two disjoint regions of n_1 and n_2 vertices as sketched here:



Show that the closeness centralities C_1 and C_2 of the two vertices, defined according to Eq. (7.29), are related by

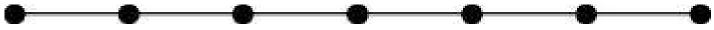
$$\frac{1}{C_1}+\frac{n_1}{n}=\frac{1}{C_2}+\frac{n_2}{n}.$$

7.4 Consider an undirected (connected) tree of n vertices. Suppose that a particular vertex in the tree has degree k, so that its removal would divide the tree into k disjoint regions, and suppose that the sizes of those regions are $n_1 \ldots n_k$.

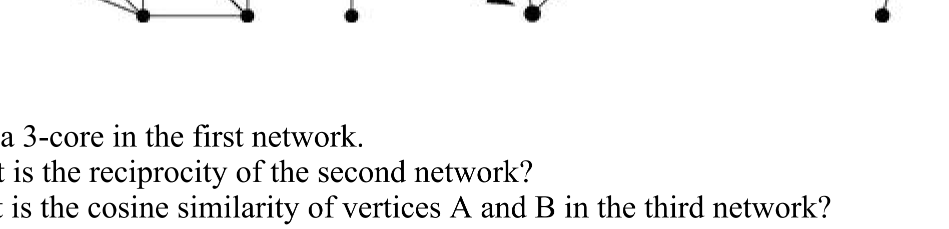
a. Show that the unnormalized betweenness centrality x of the vertex, as defined in Eq. (7.36), is

$$x=n^2-\sum_{m=1}^{k} n_m^2.$$

b. Hence, or otherwise, calculate the betweenness of the ith vertex from the end of a "line graph" of n vertices, i.e., n vertices in a row like this:



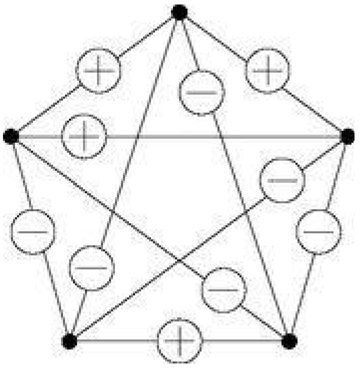
7.5 Consider these three networks:



a. Find a 3-core in the first network.
b. What is the reciprocity of the second network?
c. What is the cosine similarity of vertices A and B in the third network?

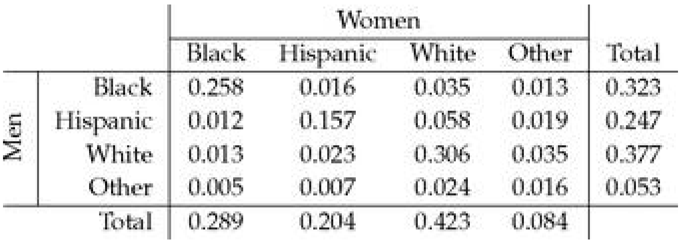
7.6 Among all pairs of vertices in a directed network that are connected by an edge or edges, suppose that half are connected in only one direction and the rest are connected in both directions. What is the reciprocity of the network?

7.7 In this network + and − indicate pairs of people who like each other or don't, respectively:



a. Is the network structurally balanced and why?
b. Is it clusterable and, if so, what are the clusters?

7.8 In a survey of couples in the US city of San Francisco, Catania *et al.* [65] recorded, among other things, the ethnicity of their interviewees and calculated the fraction of couples whose members were from each possible pairing of ethnic groups. The fractions were as follows:



		Women				
		Black	Hispanic	White	Other	Total
Men	Black	0.258	0.016	0.035	0.013	0.323
	Hispanic	0.012	0.157	0.058	0.019	0.247
	White	0.013	0.023	0.306	0.035	0.377
	Other	0.005	0.007	0.024	0.016	0.053
	Total	0.289	0.204	0.423	0.084	

Assuming the couples interviewed to be a representative sample of the edges in the undirected network of relationships for the community studied, and treating the vertices as being of four types—black, Hispanic, white, and other—calculate the numbers e_{rr} and a_r that appear in Eq. (7.76) for each type. Hence calculate the modularity of the network with respect to ethnicity.