Understanding of Network Analysis using Networkx with an example of 2022 FIFA World Cup South Korea vs Portugal Match

2023-02-11
Young-Don Choi

What is Networkx

➔ **NetworkX** is **a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.**

➔ It provides:

- tools for **the study of the structure and dynamics of social, biological, and infrastructure networks**
- a **standard programming interface and graph implementation** that is suitable for many applications
- a rapid development environment for collaborative, multidisciplinary projects
- an interface to existing numerical algorithms and code written in C, C++, and FORTRAN
- and the ability to painlessly work with large nonstandard data sets.

➔ With NetworkX you can load and store networks in standard and nonstandard data formats, generate many types of random and classic networks, analyze network structure, build network models, design new network algorithms, draw networks, and much more.

Reference: Software for Complex Networks — NetworkX 3.0 documentation

Adjacency Matrix

To store/retrieve a network in/from a computer file or memory, we need a way to formally represent its nodes and links. There are several possible network representations. The simplest is the *adjacency matrix*, an $N \times N$ matrix in which each element represents the link between the nodes indexed by the corresponding row and column.

Element a_{ij} of the adjacency matrix represents the link between nodes i and j. $a_{ij} = 1$ if i and j are adjacent, $a_{ij} = 0$ otherwise.

While the adjacency matrix representation matches the mathematical formalism of networks, it is not efficient for storing real networks, which are typically large and sparse. The required storage space grows like the square of the network size (N^2), but if the network is sparse, most of this space is wasted storing zeros (non-existing links). With large sparse networks, a more compact network representation is the *adjacency list*, a data structure that stores the list of neighbors for each node. Adjacency lists represent sparse networks efficiently because the non-existing links are ignored; only the existing links (non-zero values of the adjacency matrix) are considered.

** Reference: A First Course in Network Science (Filippo Menczer, Santo Fortunato and Clayton A. Davis)*

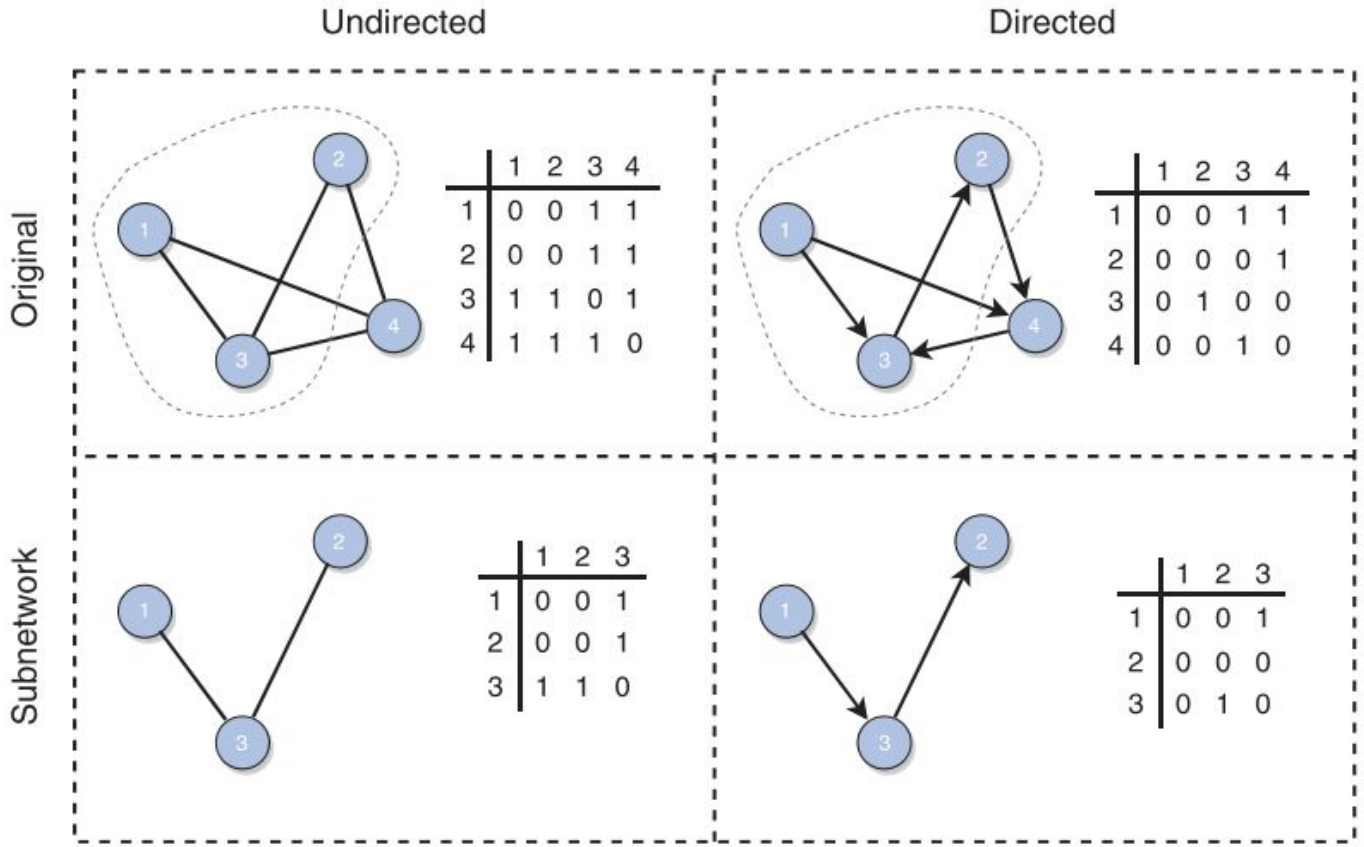

Fig. 1.3 Network and subnetwork examples. We also show the adjacency matrix representation of each network (see Section 1.9).

** Reference: A First Course in Network Science (Filippo Menczer, Santo Fortunato and Clayton A. Davis)*

Degree Correlation Coefficient (= Degree Assortativity Coefficient)

In a social network, nodes may have many properties, such as age, gender identity, ethnicity, sexual preference, location, topics of interests, and so on. Often, nodes that are connected to each other in a social network tend to be similar in their features: for example, relatives may live near each other, and friends may have similar interests. The technical name of this property is *assortativity*. Figure 2.1 illustrates assortativity based on a node feature represented by color. A more striking, real-world example from Twitter is shown in Figure 0.3. Because of assortativity, we are able to make predictions about a person's qualities by inspecting their neighbors. For instance, as we have seen in Section 0.1, researchers found that it is possible to ascertain with reasonable accuracy a Facebook user's sexual orientation and a Twitter user's political preference, even when these features are not present in their profile, by analyzing their circles of friends.

** Reference: A First Course in Network Science (Filippo Menczer, Santo Fortunato and Clayton A. Davis)*

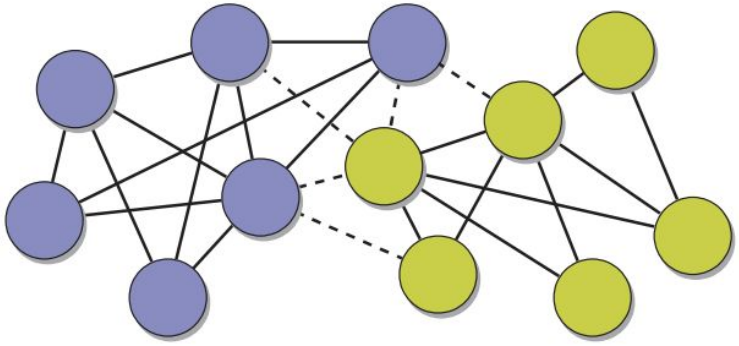

Fig. 2.1 Illustration of network assortativity. Nodes are more likely to be linked to other nodes of the same color than to nodes of different color. In particular, the majority of links for each node go to nodes of the same color, and the majority of links connect nodes of the same color. The few links connecting nodes of different colors are shown as dashed lines.

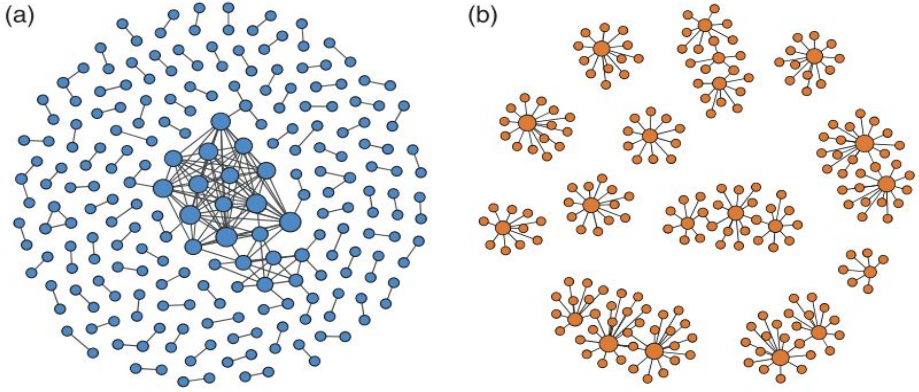

Fig. 2.2 Network degree assortativity illustrated by (a) an assortative network and (b) a disassortative network.

** Reference: A First Course in Network Science (Filippo Menczer, Santo Fortunato and Clayton A. Davis)*

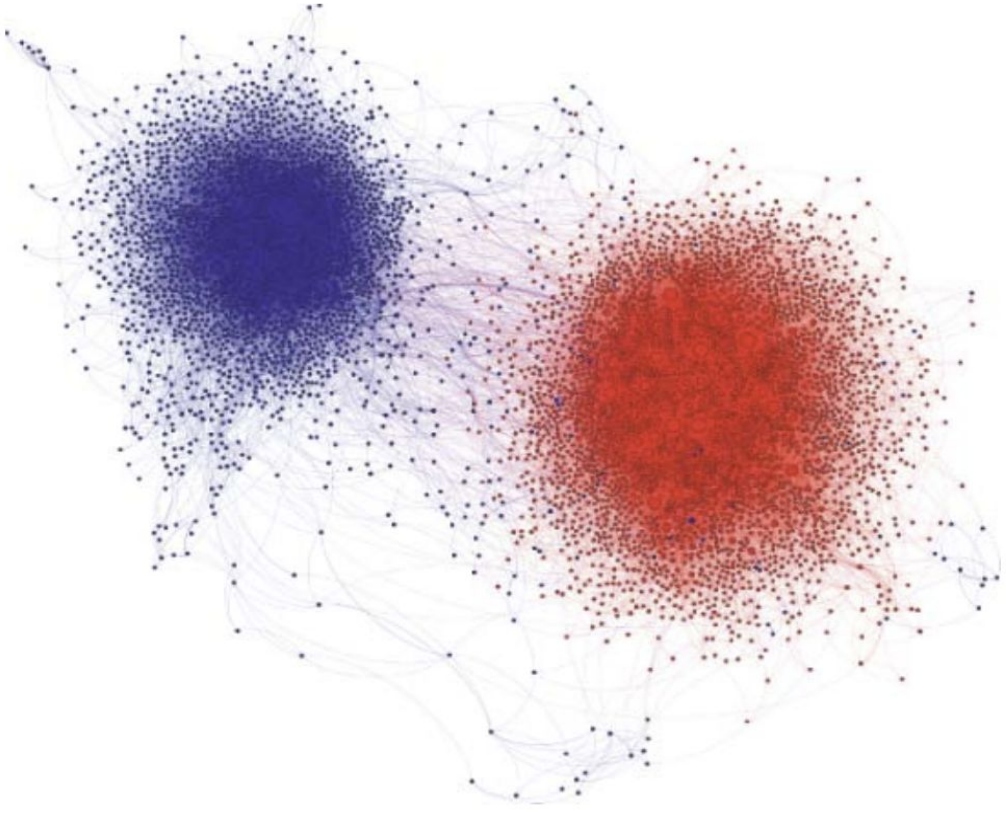

Fig. 0.3 A retweet network on Twitter, among people sharing posts about US politics. Links represent retweets of posts that used hashtags such as `#tcot` and `#p2`, associated with conservative (red) and progressive (blue) messages, respectively, around the 2010 US midterm election. When Bob retweets Alice, we draw a directed link from Alice to Bob to indicate that a message has propagated from her to him. The direction of the links is not shown.

** Reference: A First Course in Network Science (Filippo Menczer, Santo Fortunato and Clayton A. Davis)*

There are two ways to measure the degree assortativity of a network, both based on measuring the *correlation* between degrees of neighbor nodes. We say that two variables are *positively (negatively) correlated* if larger values of one variable tend to correspond to larger (smaller) values of the other. Pearson's correlation coefficient is a common way to measure correlation; it takes values in $[-1, +1]$, with 0 meaning no correlation and ± 1 meaning perfect positive/negative correlation.

One measure of network assortativity is the *assortativity coefficient*, defined as the Pearson correlation between the degrees of pairs of linked nodes. Using NetworkX:

```
r = nx.degree_assortativity_coefficient(G)
```

When the assortativity coefficient is positive, the network is assortative, and when it is negative, the network is disassortative.

** Reference: A First Course in Network Science (Filippo Menczer, Santo Fortunato and Clayton A. Davis)*

Distance analysis: Shortest Path

The concept of a path is the basis of the definition of *distance* among nodes in a network. The natural distance measure between two nodes is defined as the minimum number of links that must be traversed in a path connecting the two nodes. Such a path is called the *shortest path*, and its length is called the *shortest-path length*. There may be multiple shortest paths between two nodes; obviously they all must have the same length. In Section 2.5 we will see how to find the shortest path between two nodes. In some cases, such as transportation networks, one can imagine that a link is associated with a geographic distance between the adjacent nodes. In such cases we can redefine the path length as the *sum of the distances* associated with the links along the path; the length of a path from Berlin to Rome by way of Paris is the sum of the distances from Berlin to Paris and from Paris to Rome. You can think of an unweighted network as a special case in which all links have distance one.

** Reference: A First Course in Network Science (Filippo Menczer, Santo Fortunato and Clayton A. Davis)*

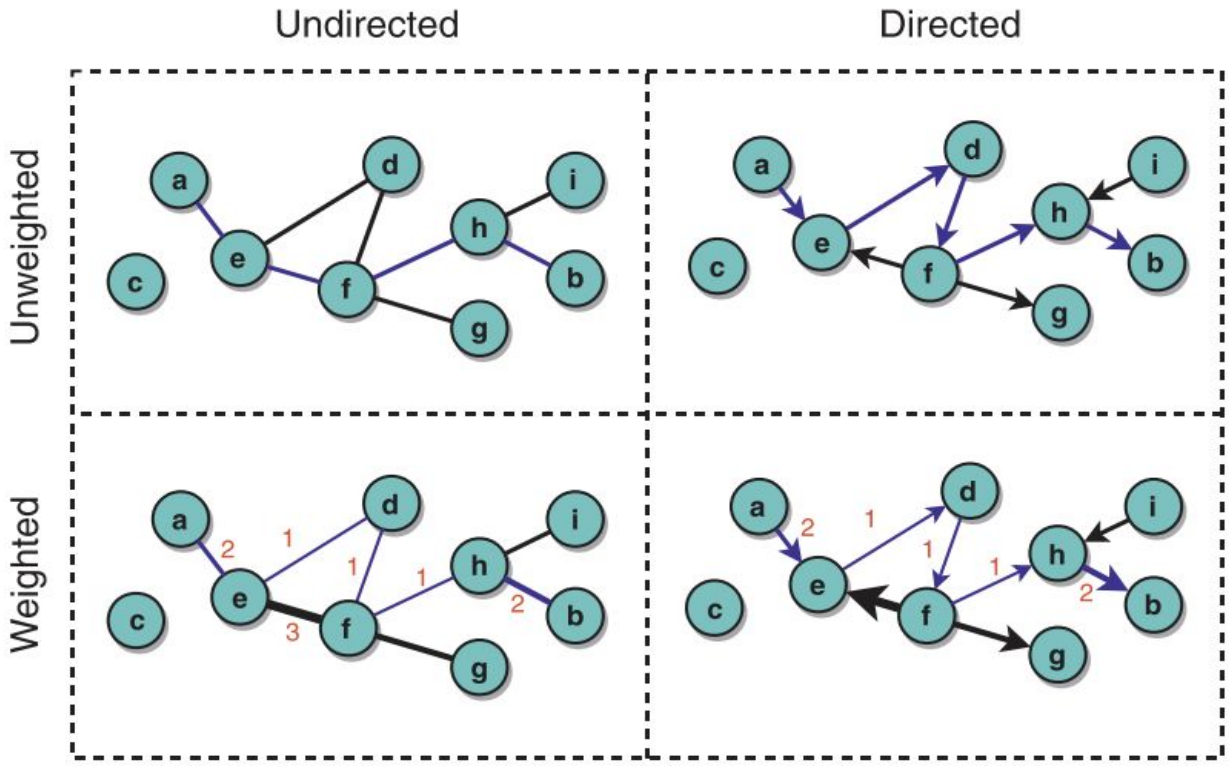

Fig. 2.3 Shortest paths in undirected, directed, unweighted, and weighted networks. Link weights represent distances and are shown in red. In each case the shortest path between nodes **a** and **b**, or from **a** to **b** in directed cases, is highlighted in blue. There is no path between node **c** and any other node. In directed networks, the shortest path must be consistent with the direction of the links along the path; there is no directed path from **b** to **a**.

** Reference: A First Course in Network Science (Filippo Menczer, Santo Fortunato and Clayton A. Davis)*

The undirected, weighted network in Figure 2.3 shows what happens when we use link distances. In this case the shortest path between **a** and **b** goes through **d**: it has an extra link, but the sum of the distances between **e** and **f** through **d** is 1 + 1 = 2, which is less than the distance 3 associated with the link (**e**, **f**). The directed, weighted case is straightforward: the shortest path is obtained by minimizing the sum of the distances along the path, while respecting the directions of the links. In both weighted network examples, the shortest-path length is $\ell_{ab} = 7$.

In many networks, link weights express a measure of similarity or intensity of interaction between two connected nodes. We may then be interested in finding paths with large weights. A common approach is to transform the weights into distances by taking the inverse (one divided by the weight), so that a large weight corresponds to a short distance. Then the problem becomes equivalent to finding short-distance paths.

By using the shortest-path length as a measure of distance among nodes, it is possible to define aggregate distance measures for an entire network: the *average shortest-path length* (or simply *average path length*) is obtained by averaging the shortest-path lengths across all pairs of nodes. The *diameter* of the network is instead the maximum shortest-path length across all pairs of nodes (i.e. the length of the longest shortest path in the network). The name is inspired by geometry, where the diameter is the longest distance between any two points on a circle.

** Reference: A First Course in Network Science (Filippo Menczer, Santo Fortunato and Clayton A. Davis)*

Clustering Coefficient

In a social network, if Alice and Bob are both friends of Charlie's, they are also likely to be friends of each other. In other words, there is a good chance that a friend of my friend is also my friend. This translates into the presence of many *triangles* in the network. As illustrated in Figure 2.11(a), a *triangle* is a triad (set of three nodes) where each pair of nodes is connected. The connectivity among the neighbors of the nodes is an important feature of the local structure of the network because it captures how tightly knit, or *clustered*, the nodes are.

The *clustering coefficient* of a node is the *fraction of pairs of the node's neighbors that are connected to each other*. This is the same as the ratio between the number of triangles that include the node, and the maximum number of triangles in which the node *could* participate.

The *clustering coefficient* of node i is formally defined as

$$C(i) = \frac{\tau(i)}{\tau_{max}(i)} = \frac{\tau(i)}{\binom{k_i}{2}} = \frac{2\tau(i)}{k_i(k_i-1)}, \qquad (2.6)$$

[3] Datasets for these networks are available in the book's GitHub repository: github.com/CambridgeUniversityPress/FirstCourseNetworkScience

** Reference: A First Course in Network Science (Filippo Menczer, Santo Fortunato and Clayton A. Davis)*

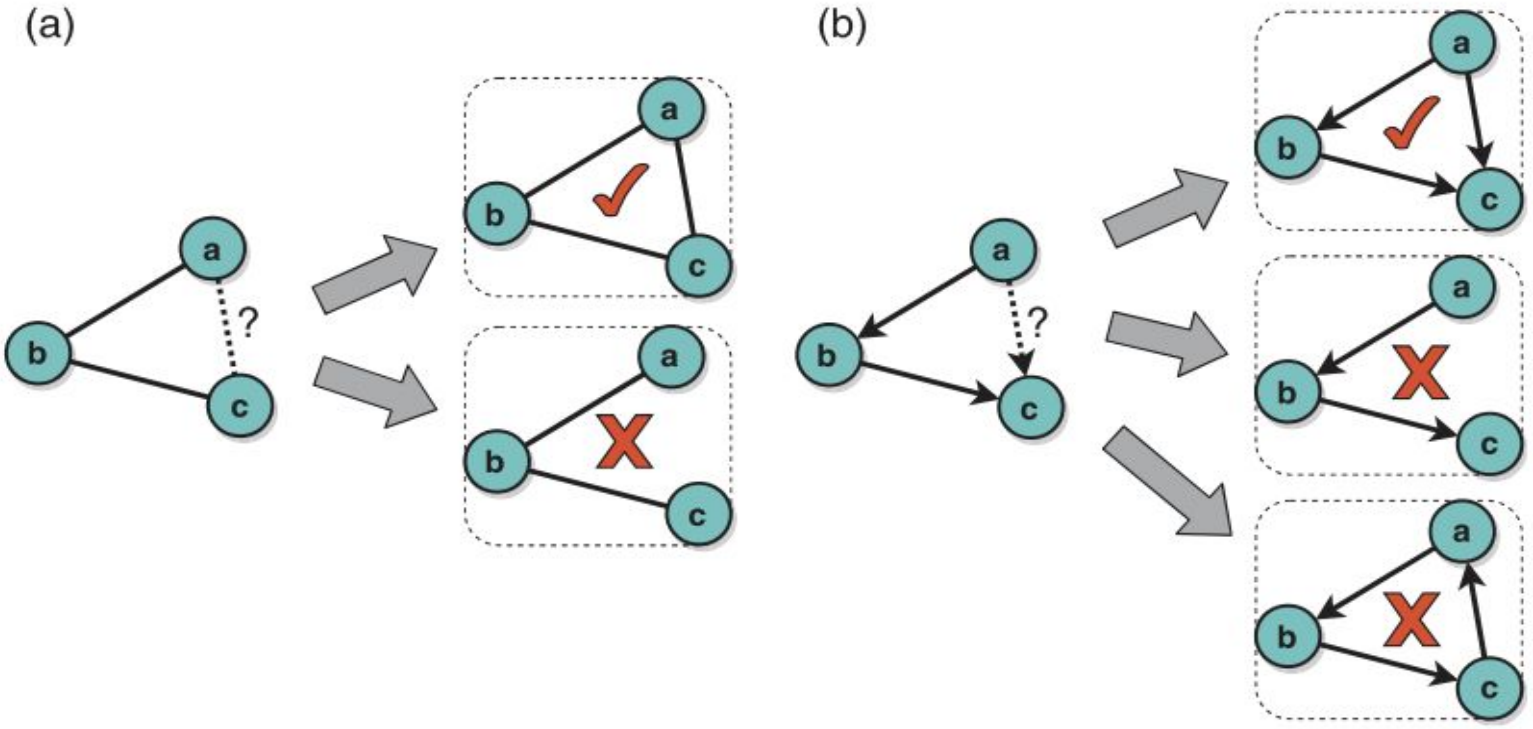

Fig. 2.11 Triads and triangles. (a) In an undirected network, node **b** has neighbors **a** and **c**. They may or may not form a triangle, depending on whether or not **a** and **c** are connected to each other. (b) In a directed network, node **a** links to **b** and node **b** links to **c**. A shortcut link from **a** to **c** would form a directed triangle.

** Reference: A First Course in Network Science (Filippo Menczer, Santo Fortunato and Clayton A. Davis)*

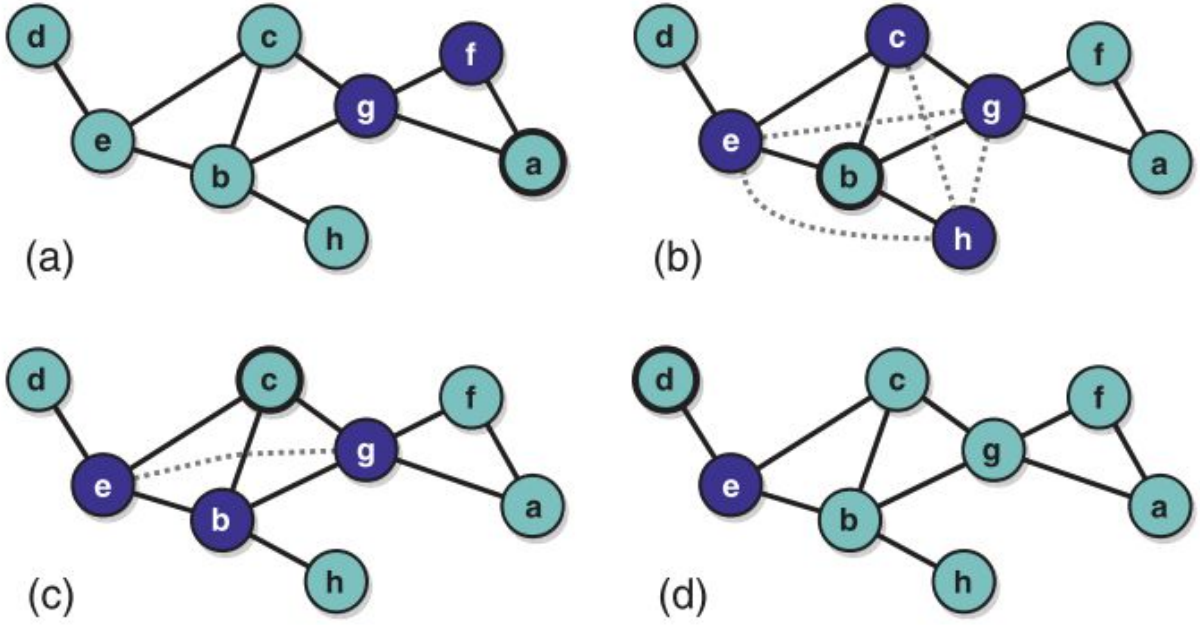

Fig. 2.12 Examples of clustering coefficient. (a) Node **a** has two neighbors **f** and **g** that are connected, forming a triangle. (b) Node **b** has four neighbors **c**, **e**, **g**, and **h**. Two of the six pairs of neighbors are connected, forming two out of six possible triangles. The missing triangle connections are shown by dotted gray lines. (c) Node **c** has three neighbors **e**, **b**, and **g** forming two out of three possible triangles. (d) Node **d** has a single neighbor **e**, therefore there are no possible triangles and the clustering coefficient is undefined.

** Reference: A First Course in Network Science (Filippo Menczer, Santo Fortunato and Clayton A. Davis)*

Between Centrality

Many phenomena taking place in networks are based on diffusion processes (Chapter 7). Examples include the transmission of information across a social network, the traffic of goods through a port, and the spread of epidemics in the network of physical contacts between the individuals of a population. This has suggested a third notion of centrality, called *betweenness*: a node is the more central, the more often it is involved in these processes.

Naturally, betweenness centrality has a different implementation for each distinct type of diffusion. The simplest and most popular implementation considers a simple process where signals are transmitted from each node to every other node, by following shortest paths. This approach is often used in transportation networks to provide an estimate of the traffic handled by the nodes, assuming that the number of shortest paths that traverse a node is a good approximation for the frequency of use of the node. The centrality is then estimated by counting how many times a node is crossed by those paths. The higher the count, the more traffic is controlled by the node, which is therefore more influential in the network.

** Reference: A First Course in Network Science (Filippo Menczer, Santo Fortunato and Clayton A. Davis)*

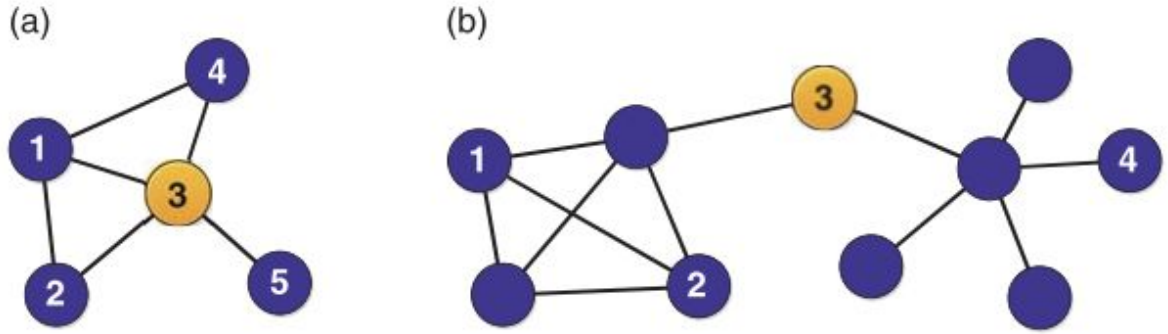

Fig. 3.1 Illustrations of node betweenness centrality. (a) The orange node has high degree ($k_3 = 4$) as well as high betweenness ($b_3 = 3.5$). (b) The orange node has low degree ($k_3 = 2$) but keeps the network connected, acting as the only bridge between nodes in the two subnetworks. For example, the shortest path between nodes **1** and **2** does not go through the orange node, but the path between **1** and **4** does. In fact, all the shortest paths between the four nodes in one subnetwork and the five nodes in the other subnetwork go through the orange node. Therefore its betweenness is $b_3 = 4 \times 5 = 20$.

To calculate betweenness centrality, you take every pair of the network and count how many times a node can interrupt the shortest paths (geodesic distance) between the two nodes of the pair.

** Reference: A First Course in Network Science (Filippo Menczer, Santo Fortunato and Clayton A. Davis)*

Thanks