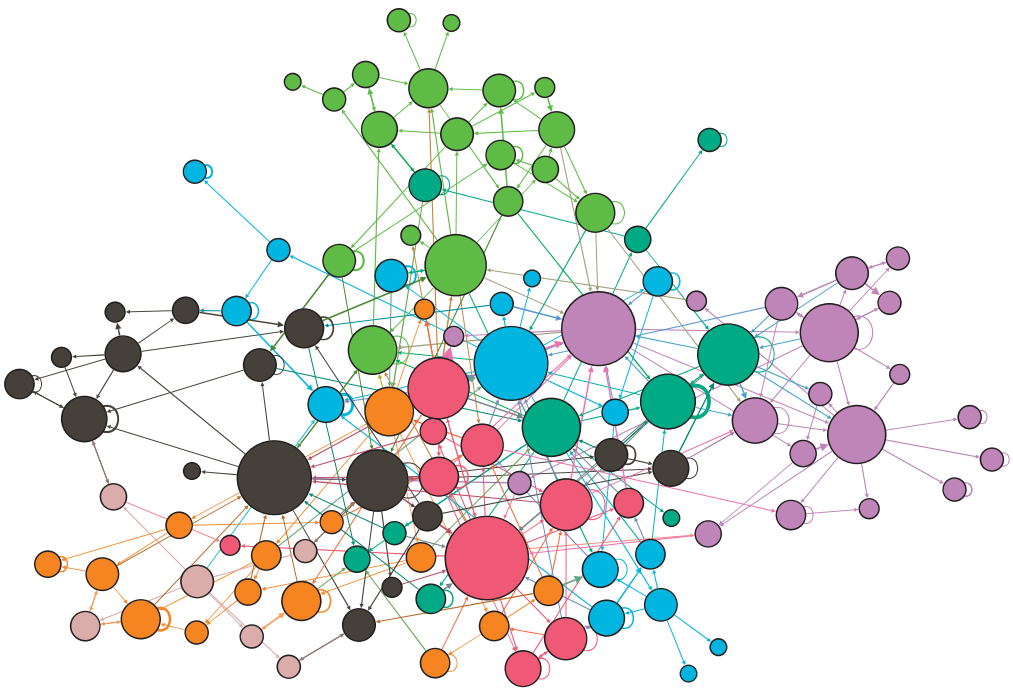
Preface

Networks are present in all aspects of our lives: networks of friends, communications, computers, the Web, and transportation are examples that we experience outwardly, while our brain cells and the proteins within our body form networks that determine our survival and intelligence. When people communicate through Facebook or Twitter, buy stuff on Amazon, search on Google, or buy an air ticket to visit family, they use networks without knowing it. Today, a basic understanding of network processes is required in job sectors from technology to marketing, from management to design, and from biology to the arts and humanities. This textbook explores the study of networks and how they help us understand the complex patterns of relationships that shape our lives.



This book is also a network! The relationships between chapters, sections, and subsections are depicted in the above image. Links represent both the hierarchical structure of the book (as seen in the Contents) and cross-references among chapters, sections, figures, tables, equations, and boxes. Node colors represent chapters and node size is proportional to the number of neighbors.

0 Introduction

net·work: (*n.*) an interconnected or interrelated chain, group, or system.

Imagine a world where people have no friends. Where roads never intersect. Where computers are not interconnected. This world without networks would be a very sad and boring place, where nothing happens — and even if something happened, nobody would know. Such a world is unimaginable, because our life is completely defined by networks: relationships, interactions, communications, and the Web. Biological networks governing the interactions between genes in our cells determine our development, neural networks in our brain make us think, information networks guide our knowledge and culture, transportation networks allow us to move, and social networks sustain our life.

Networks are a general yet powerful way to represent and study simple and complex interactions. This book explores the study of networks and how they help us understand the patterns of connections and relationships that shape our lives. In essence, a network is the simplest description of a set of interconnected entities, which we call *nodes*, and their connections, which we call *links*. The network representation is so general and powerful because it strips out many details of a particular system and focuses on the interactions among its elements. Networks are thus used to study widely diverse systems. Nodes can represent all sorts of entities: people, cities, computers, websites, concepts, cells, genes, species, and so on. Links represent relationships or interactions between these entities: friendships among people, flights between airports, packets exchanged among computers on the Internet, links between Web pages, synapses between neurons, and so on.

Before we introduce the basic concepts, definitions, and nomenclature about networks, let us explore a few examples of social, infrastructure, information, and biological networks. Data for all the examples presented here is available on the book's GitHub repository.[1] The networks on which we focus in this book tend to be large, even though one can learn a lot from studying smaller systems, such as social networks built from surveys or interviews. In these cases it is meaningful to examine individual nodes and connections in great detail, whereas analyses of large networks tend to focus on macroscopic properties, classes of nodes and links, typical behaviors, and anomalies.

[1] github.com/CambridgeUniversityPress/FirstCourseNetworkScience

0.1 Social Networks

A social network is a group of people connected by some type of relationship. Friendship, collaboration, romance, or mere acquaintance are all examples of social relationships that connect pairs of people. When we talk about a social network, we typically think of a particular type of relationship. A person is represented by a node in the social network, and the relationship is represented by a link between two people. The network is therefore a representation of the relationship. It allows us to talk about the relationship, to describe it and analyze it at a level that goes beyond a pair of people.

There are many different types of social networks, and they are important to study. Health workers analyze networks of sexual relationships to find ways to combat the spread of sexually transmitted diseases. Economists study job referral networks to address inequality and segregation in labor markets. And scientists inspect coauthorship networks in scholarly publications to identify influential thinkers and ideas.

These days we use online social networking sites to keep track of our social ties. Platforms like Facebook and Twitter allow us to keep in touch with many people — partners, friends, colleagues, and acquaintances, sometimes in the hundreds — and communicate with them conveniently, irrespective of distance. Figure 0.1 illustrates a familiar network, a portion of the Facebook social graph. In this network, nodes are people with a Facebook account at a US university, and connections may represent different types of relationship, from real friendship to mere acquaintance. Just looking at the network visualization reveals something about the underlying social structure. Some people have more connections; we represent this by making the corresponding nodes larger and darker. These might be popular students, teachers, or administrators. We also notice that the network is roughly divided into two parts. The data is anonymized so we cannot tell for sure, but a possible interpretation is that the larger subnetwork comprises mostly undergraduate students, and the smaller one includes mostly graduate students. There are connections between nodes in the two groups, but not as many as among nodes within each group. In other words, undergraduate students are more likely to be friends with other undergraduates than with graduate students. Later we will introduce formal names for all these observations, which are typical of most social networks.

The availability of data from online social networks is very exciting for scientists. We can study human interactions at a scale and resolution that was never possible in the past: who befriends whom, who pays attention to what, who likes what, what gets recommended, and how this information propagates through the network. This data provides us with an unprecedented opportunity to discover, track, mine, and model what people do. Like the telescope gave us a first view of distant planets and stars, and the microscope allowed us to peek into living tissue and micro-organisms, social media are enabling the study of social systems and human activity. However, as exciting as these opportunities are to researchers, they don't come without risks of abuse. Online interactions expose our private personal information. We've all heard stories about employers finding embarrassing pictures of prospective employees, or scandals related to hackers and political organizations

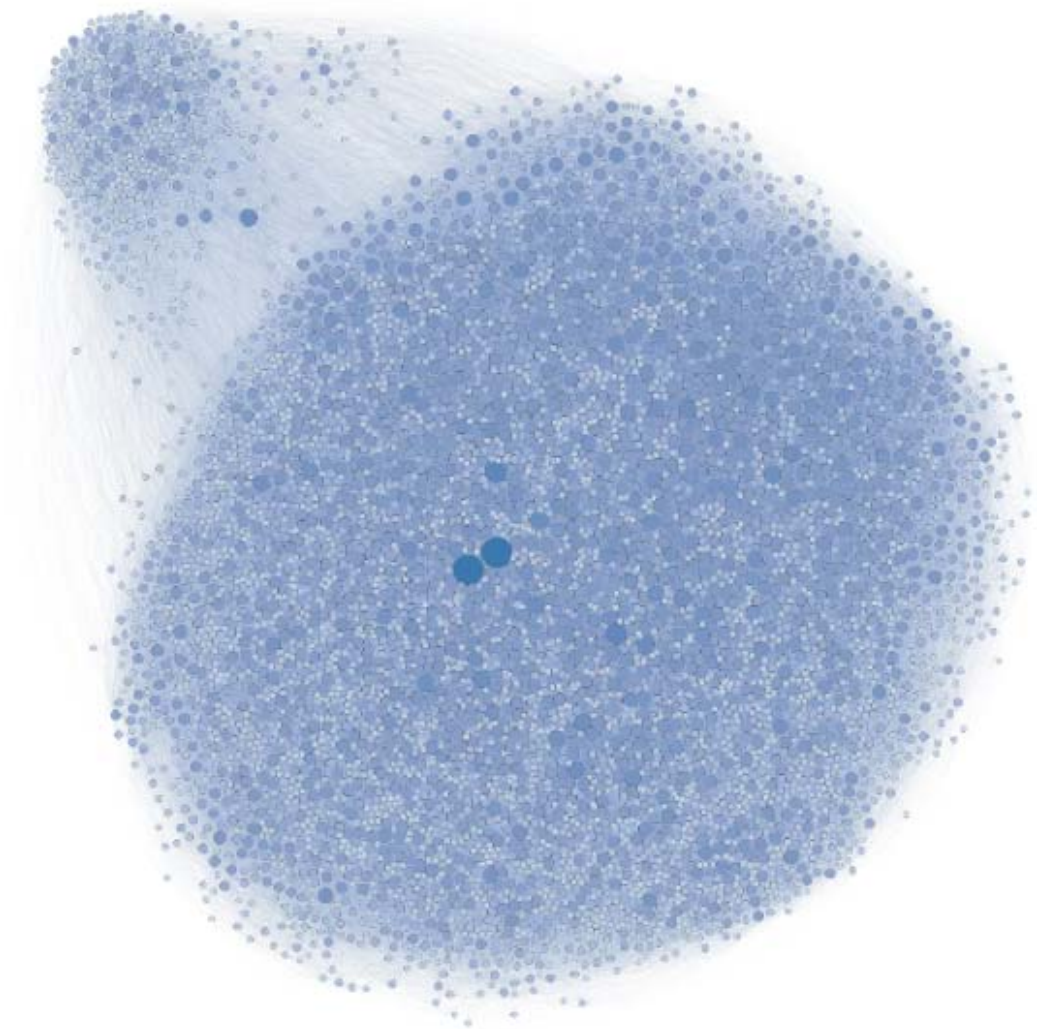


Fig. 0.1 Visualization of a network of Facebook users at Northwestern University. Nodes represent people, and links stand for Facebook friend connections.

amassing data about millions of users. The dangers can be subtle. Knowing a little bit of information about a large number of people can reveal a lot more than intended. Using data from Facebook, two MIT students found that just by looking at the gender and sexuality of a person's online friends, they could predict whether that person was gay. Online social networks also make impersonation easy to do and hard to detect. Social phishing is the technique of impersonating a victim's friend (as inferred from an online social network) to induce the victim to disclose sensitive information. Two Indiana University students demonstrated that they were able to obtain the secret passwords of 72% of victims in this way.

Data about a social network can be extracted from many sources. If we want to map human mobility patterns to improve urban transportation networks, we can collect call

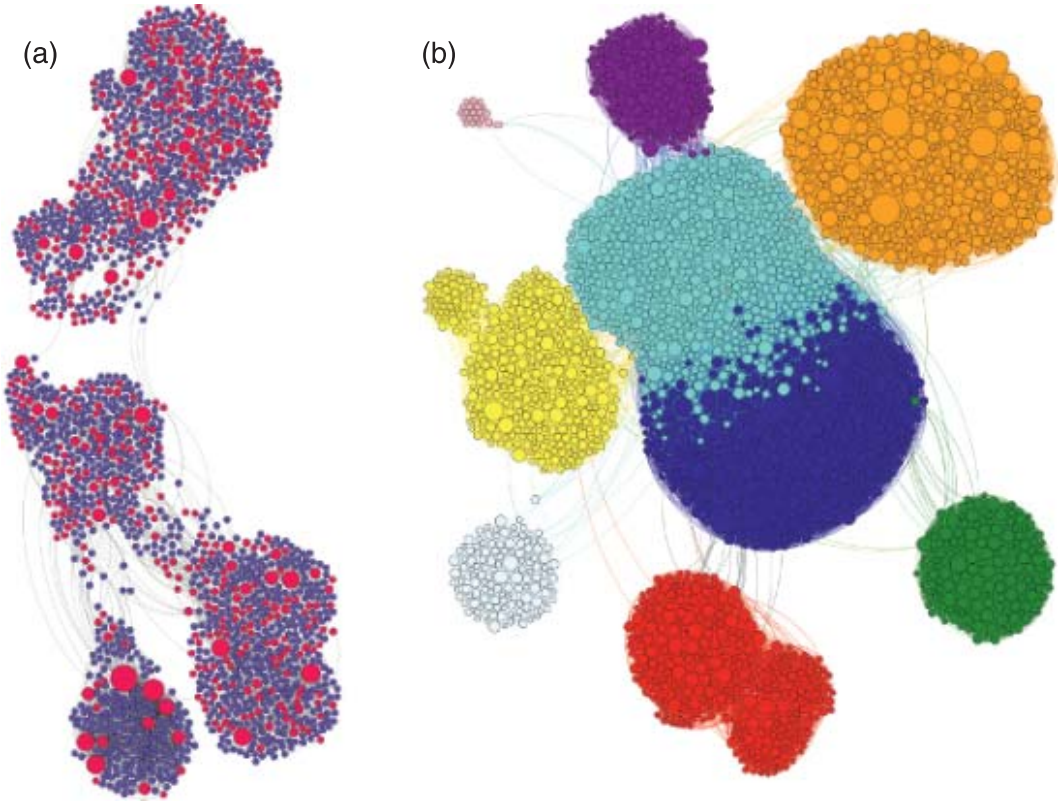


Fig. 0.2 (a) A movie-star network, based on a small sample of movies, actors, and actresses from the Internet Movie Database. Nodes represent movies (blue) or actors/actresses (red). A link connects an actor or actress to a movie in which they starred. (b) A movie co-star network, based on a small sample of actors and actresses from the Internet Movie Database. A link connects two people who have co-starred in at least one movie. Colors represent film genres or languages/countries.

data from cell phones. If we want to map coauthorship among scientists, we can extract the names from a database of scientific publications; two coauthors of the same paper will be linked to each other. (This is not a trivial exercise, because several scientists may have common names.) If we want to map the collaboration among movie stars, we can extract movie credits data from the Internet Movie Database (IMDB.com). Figure 0.2 illustrates two such networks. In one case, there are actually two kinds of nodes: movies and actors/actresses. We draw a link between an actress and a movie in which she has starred. In the other case, we focus on links between actors/actresses who have co-starred in movies. Although the depicted networks capture only tiny portions of the movie database, we again notice some clear patterns. Larger nodes have more connections, representing stars who acted in many movies. We also see that the networks are structured into several dense groups associated with periods, languages, or film genres: Hollywood (blue), Western (cyan), Mexican (purple), Chinese (yellow), Filipino (orange), Turkish and Eastern European (green), Indian (red), Greek (white), and adult (pink) stars in Figure 0.2(b). In Chapter 6 you will learn how to discover these groups and find out what they are about.

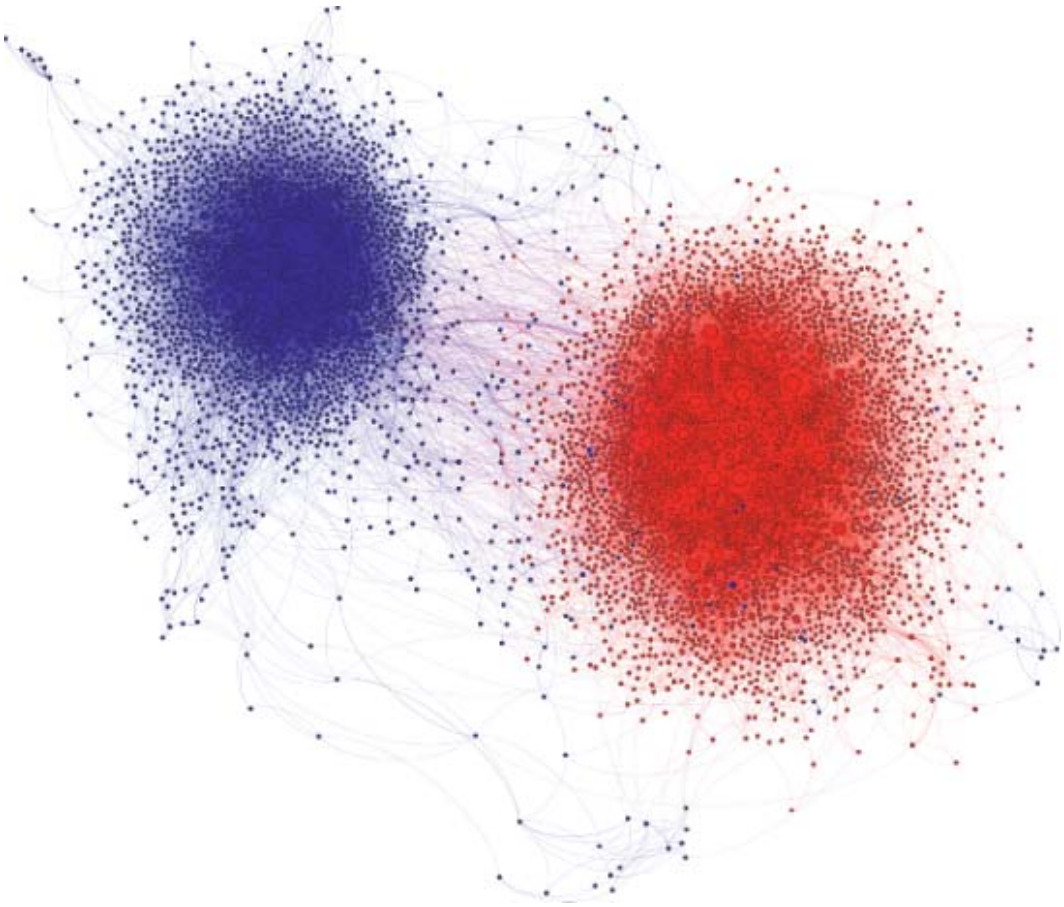


Fig. 0.3 A retweet network on Twitter, among people sharing posts about US politics. Links represent retweets of posts that used hashtags such as `#tcot` and `#p2`, associated with conservative (red) and progressive (blue) messages, respectively, around the 2010 US midterm election. When Bob retweets 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.

0.2 Communication Networks

In the Facebook and movie networks, links are reciprocal: you cannot friend someone on Facebook unless they agree, and you cannot star in a movie without being listed in the credits. Not all social networks have reciprocal links, however. For example, Twitter is a popular social network with links that are not necessarily reciprocal: Alice can follow Bob without Bob necessarily following Alice back. As a result, the relationships captured by the Twitter network are not friendship; you follow someone to see what they post. When you retweet a post, your followers see it. This is a good way to share information broadly, so Twitter is a social network mainly aimed at spreading information — a communication network. The retweet network in Figure 0.3 illustrates the spread of political messages during a US election. Larger nodes are those with more outgoing links, because how many times users are retweeted by others is a way to measure their influence. You probably

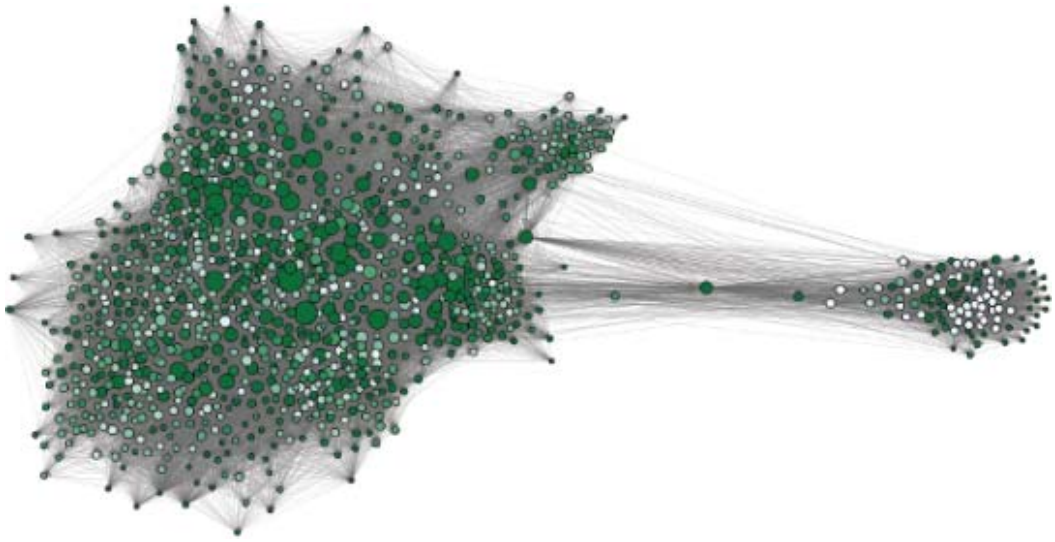


Fig. 0.4 A network based on a database of emails generated by employees of the Enron energy company. The data was acquired by the US Federal Energy Regulatory Commission during its investigation after the company's collapse in 2001. At the conclusion of the investigation, the emails were deemed to be in the public domain and made publicly available for historical research and academic purposes. Only a small portion of the central core of the network is shown. The direction of the links is shown by arrows.

noticed immediately a more striking pattern: conservative users (red nodes) mostly retweet messages from other conservatives, while progressive users (blue nodes) similarly share progressive content. In fact, such preferential patterns of social connections allow us to guess a person's political leaning with high accuracy. This property, called *homophily*, will be discussed in Chapter 2; the algorithm for inferring political preference from the network's structure will be presented in Chapter 6.

Networks like Twitter let us trace the diffusion of hashtags and news, observing how ideas and cultural concepts spread from person to person. But social media are also used to spread misinformation, which is unknowingly passed on by gullible users. Using fake news sites and automated or semi-automated accounts known as "social bots," a malicious entity can cheaply and effectively generate and amplify a disinformation campaign, either for political purposes or to monetize traffic through ads. In recent years we have observed a sharp increase in these types of manipulation of social media on a global scale. If one can control what information people see online, one can manipulate their opinions. This is a threat to democracy in many countries, because without well-informed voters one cannot have free elections. Academic researchers and industry engineers are working hard to develop countermeasures. Understanding the structure and dynamics of the networks that enable the spread of information is a critical component of these efforts.

The social links in Twitter are in place before a user generates a post, which is typically broadcast to all of the user's followers. In email, just like in social networks, nodes are people. However, each message is intended for one or more specific recipients. Links are

based on the messages exchanged. Email does not depend on a particular platform; the protocol is open and distributed, so that no single organization controls all of the traffic. As a result, email is still among the most widely used communication networks. Figure 0.4 illustrates an example of an email network. Again, links are directed from the sender to the receiver of an email, indicated by arrows. Node size and color represent two different features: number of incoming and outgoing links, respectively. A larger node receives emails from more people, and a darker node sends emails to more people. The fact that larger nodes tend to be darker and vice versa tells us that there is a correlation between sending and receiving emails.

0.3 The Web and Wikipedia

The Web is the largest information network. While it is now used to provide all kinds of services, it was originally just a network of documents (pages) connected by "hyperlinks," or clickable links. In the early 1990s, Tim Berners-Lee wanted to simplify access by scientists to information about high-energy physics experiments at the European Organization for Nuclear Research (CERN) near Geneva. He came up with three key ideas: (1) a naming system for pages, the Uniform Resource Locator (URL); (2) a simple language for writing documents, called HyperText Markup Language (HTML), including hyperlinks from one page to another; and (3) a simple protocol called HyperText Transfer Protocol (HTTP) for clients (browsers) to talk to servers. With these three components, the Web was born. Berners-Lee even implemented the first Web server and browser software to download pages and media from servers by clicking on links. We can actually see two networks at play here: the static "link graph" made up of a snapshot of Web pages and links at a given time, and the dynamic traffic network emerging from people navigating the Web. To paraphrase the classic philosophical riddle, if there is a link between two pages but nobody clicks on it, is it really part of the Web? The answer of course depends on which of the two networks we are thinking about when we say "Web." In later chapters we will spend more time exploring both of these information networks.

The Web is too large to visualize even a small portion of it in a meaningful way. Let us focus on Wikipedia, which is a network of pages (articles) on a single website. Wikipedia is a collaborative encyclopedia edited by thousands of volunteers around the world, and it is one of the most popular destinations on the Web. There are versions of Wikipedia in many languages, so let us focus on the English one. Still, the English Wikipedia is a huge network with millions of articles (and growing!). So let us focus on just a small subset of articles about math, shown in Figure 0.5. Here, node size represents *PageRank*, a measure of centrality that captures how important an article is based on other articles that link to it — something we will discuss in Chapter 4. For example, the large white node in the middle is the general article about *Mathematics*. Another feature of this network is the presence of a large "core" (gray) and several smaller groups. These groups are tightly connected clusters of articles on specific topics or branches of math. For example, articles

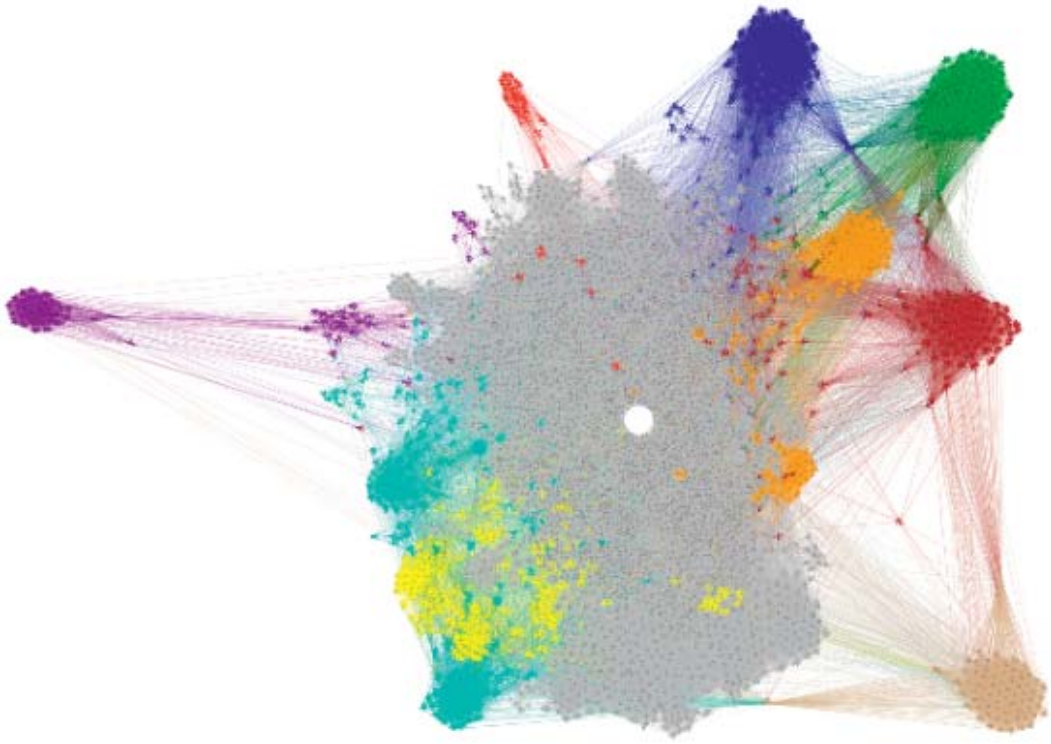


Fig. 0.5 A portion of the Wikipedia information network. Nodes are articles about math. We only consider links among Wikipedia articles, and disregard links to external pages. Node size is proportional to article importance, and colors highlight communities discussed in the text.

about historical Greek (blue), Arab (green), and Indian (brown) mathematicians; about contemporary Indian mathematicians (tan); about math and art (orange), statistics (cyan), game theory (yellow), mathematical software (purple), and pedagogical theory (red). We also observe several "bridge" nodes that connect multiple clusters. These features are found in many real-world networks.

0.4 The Internet

We often think of the Internet as a network of computers and other connected devices, but in reality it is a *network of networks.* In fact the word originates from *internetworking,* or connecting different computer networks through special nodes called *routers*. We can therefore observe the Internet at many levels: at the lowest level we have hardware devices that connect individual computers in the same local or wide-area network. These networks are connected by routers, so we can zoom out and think of the network of routers. If we zoom out further we find groups of networks managed by an Internet Service Provider (ISP). This organization decides its internal network topology (how

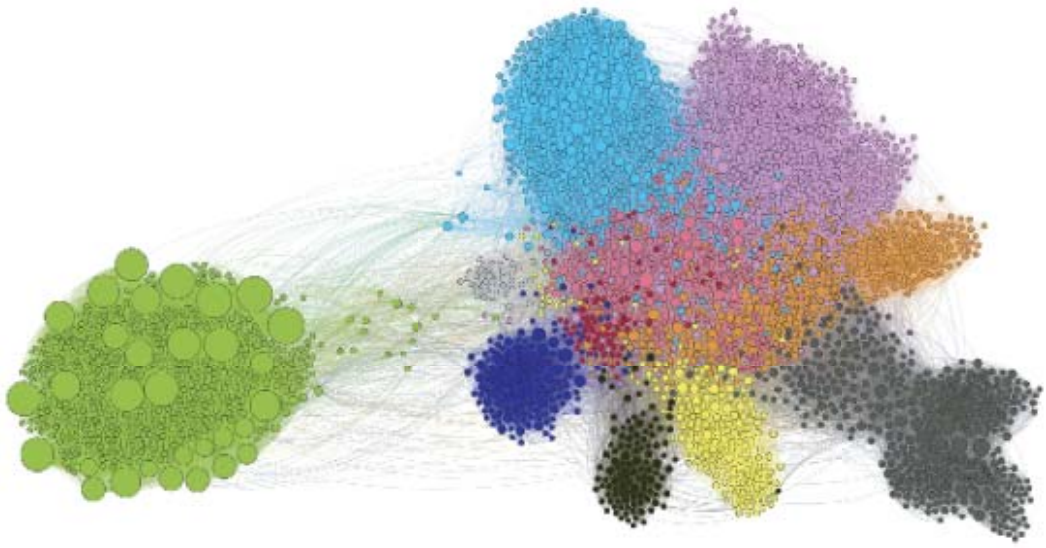


Fig. 0.6 A portion of the Internet router network. The map is a snapshot generated by the Center for Applied Internet Data Analysis (CAIDA.org) using tools that send out small packets of data (probes) between Internet hosts. Colors are assigned according to a community detection algorithm that identifies dense clusters reflecting the geographic distribution of routers. In Chapter 6 you will learn how to use this methodology to study what those clusters represent.

routers are connected) autonomously, and therefore is also called an "autonomous system" (AS). Special "border" routers connect one AS to another, forming what we call the AS network.

Figure 0.6 shows a small portion of the Internet router network. Although the Internet has evolved without central control or coordination, ISPs follow local rules on how to connect their routers. They try to provide the best service at the lowest cost. Certain regularities emerge as a result. For instance, the portion of the Internet that carries the most traffic is often referred to as the "backbone." The large telecommunication companies that manage the Internet backbone have a significant interest in preventing disruption, so they engineer their networks with a lot of redundancy. We thus observe a dense "core," with large routers connected to each other. As we move toward the "periphery" of the Internet — our home routers — the network is more sparsely connected. Such a hierarchical *core–periphery structure* is common in many different types of networks, and will be discussed in Chapter 2. In the router network depicted in Figure 0.6, the green cluster on the left appears well separated from the rest of the network. This is likely due to a bias in the probe methodology used to map these networks: most measurements were taken from the United States, and the routers in this cluster are located there. A related peculiarity is the presence of very large nodes in the green cluster, indicating routers with many connections. This may actually be a measurement error resulting from the same bias. In fact, a router can only have a limited number of connections due to hardware constraints. Let it serve as a reminder that if we use a flawed method to collect data about a network, its analysis may lead to wrong conclusions.

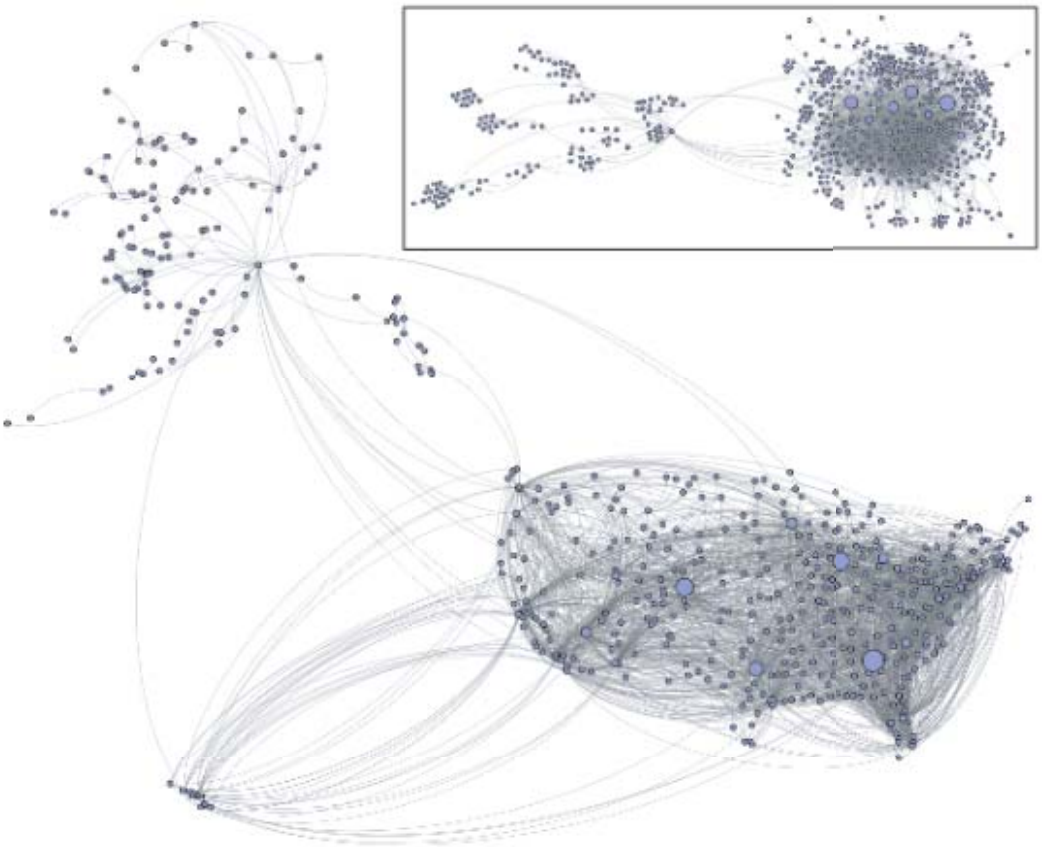


Fig. 0.7 The US air transportation network (flight data from OpenFlights.org). Nodes are positioned according to the geographic coordinates of the corresponding airports, so that we can make out the shape of the continental United States, Alaska, and Hawaii. Note that the map projection makes Alaska appear bigger than its actual size due to its latitude. The airport hubs with most connections (e.g. Atlanta, Chicago, Denver) are clearly recognizable. The inset maps the same network, but with a different "force-directed" layout, discussed in Section 1.10.

0.5 Transportation Networks

Another important class of networks concerns various types of transportation. Nodes are locations: cities, road intersections, airports, ports, train or subway stations. These networks are very different from one another, however. Road networks, for example, evolve in a local fashion to minimize the distance traveled between nearby cities. This leads to the emergence of grid-like structures, in which most nodes have a comparable number of connections — say, four-way intersections. Figure 0.7 shows an air transportation network, which does not have a grid structure. The reason is that airlines try to minimize the number of hops between source and destination without adding costly direct flights between low-traffic airports. The simple solution is to add flights connecting airports to existing hubs. As a result, air flight networks display a "hub and spoke" structure: a few hubs have huge numbers of links, while the majority of nodes have very few connections.

When studying certain types of networks, especially related to transportation and communications, we can think of them in terms of their static structure, or the dynamic processes that occur on these networks. Consider the air transportation network, for instance. We might view the picture in Figure 0.7 as a set of routes that exist between airports, independently of the actual travel that takes place on them; or as a traffic network that emerges from people moving between airports. In the latter sense, links are diverse because they carry different amounts of traffic, and they also change over time. Both the structure and dynamics of networks are important. Sometimes we simply capture the dynamics by representing traffic through link directions and weights, as we discuss in Chapter 4. Other times we may wish to study the actual processes that allow a network to grow and change over time, or the interactions that take place on a network. Chapters 5 and 7 are dedicated to these topics about network dynamics.

0.6 Biological Networks

Within the cells inside our bodies, special molecules called proteins interact in a variety of ways. For example, when a protein folds, its change in structure can regulate the function of another protein or the activity of an enzyme. Enzymes (themselves proteins) catalyze biochemical reactions and are vital to metabolism, which maintains life by harvesting energy for building and supporting the proteins that make up our tissues and organs. Proteins also regulate cell signaling and immune responses. All of these interactions can be seen as networks: protein interaction networks, metabolic networks, gene regulatory networks,

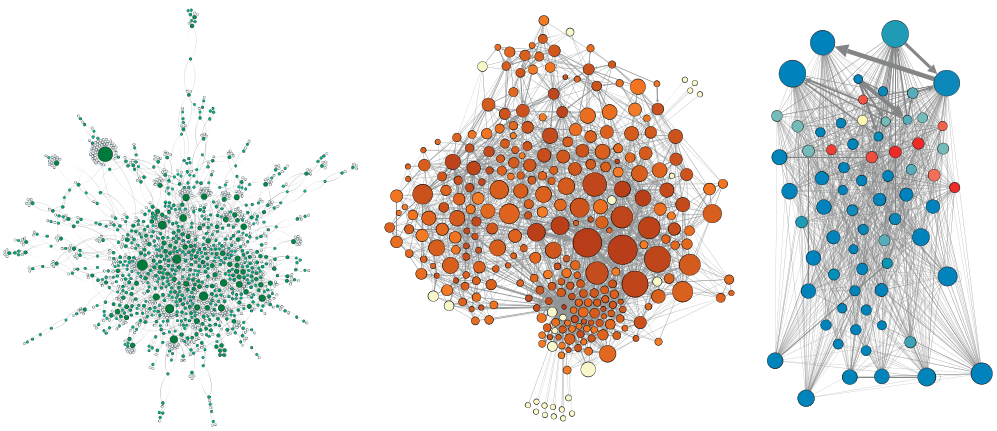


Fig. 0.8 Three biological networks. Left: Protein interaction network of yeast. Node size is proportional to the number of interacting proteins. Center: Neural network of the roundworm *Caenorhabditis elegans*. Large and red nodes represent neurons with more outgoing and incoming synapses, respectively. Right: Food web of species in the Florida Everglades. A directed link goes from a prey to a predator species. The weight (width) of a link represents the energy flux between the two species. Node size and color represent incoming and outgoing links, respectively, so that large blue nodes are the species at the top of the food chain, while small red nodes are the species at the bottom.

and so on. These biological networks exist within a cell. At a higher level, within a body, connections between neural cells (synapses) give rise to the neural networks that form our brains. And at an even higher level, entire species interact. An animal of one species may see another species as food, creating an ecological network, or food web among species. When we think of this network, ecological balance depends on the availability of species that sustain each other. Removing a node in such a food web — when a species goes extinct, for example — affects the survival of other parts of the ecosystem network. Figure 0.8 illustrates three types of biological networks: a protein interaction network, a neural network, and a food web. They are all essential elements of life on our planet.

0.7 Summary

Networks are a general way to model and study complex systems with many interacting elements. We have seen several examples of networks. Nodes can represent many different types of objects, from people to Web pages, from proteins to species, from Internet routers to airports. Nodes can have features associated with them beside labels: geographic location, wealth, activity, number of connections, and so on. Links can also represent many different kinds of relationships, from physical to virtual, from chemical to social, from communicative to informative. They can have a direction (like Web hyperlinks and email) or be reciprocal (like marriage). They can all be the same or have different features such as similarity, distance, traffic, volume, weight, and so on.

0.8 Further Reading

The use of networks to graphically represent social relationships among individuals was introduced by Moreno and Jennings (1934), who called these social networks *sociograms*.

Much more recently, studies have shown that online social networks can reveal a person's sexual orientation (Jernigan and Mistree, 2009) and facilitate highly effective phishing attacks (Jagatic *et al.*, 2007). Conover *et al.* (2011b) showed that political information diffusion networks on Twitter are very polarized and segregated. As a result we can predict the political leaning of most users with high accuracy by starting with a few node labels and propagating them through network neighbors (Conover *et al.*, 2011a).

You can read about the vision, design, and history of the Web in a book coauthored by its inventor (Berners-Lee and Fischetti, 2000).

Spring *et al.* (2002) explain how probes are used to measure the topology of the Internet. Achlioptas *et al.* (2009) show that these approaches have sampling bias. Computer scientists analyze the structure of routers and autonomous system networks to develop models called "topology generators," which can help in the design of these networks (Rossi *et al.*, 2013). To learn more about Internet networks, we recommend the book by Pastor-Satorras and Vespignani (2007).

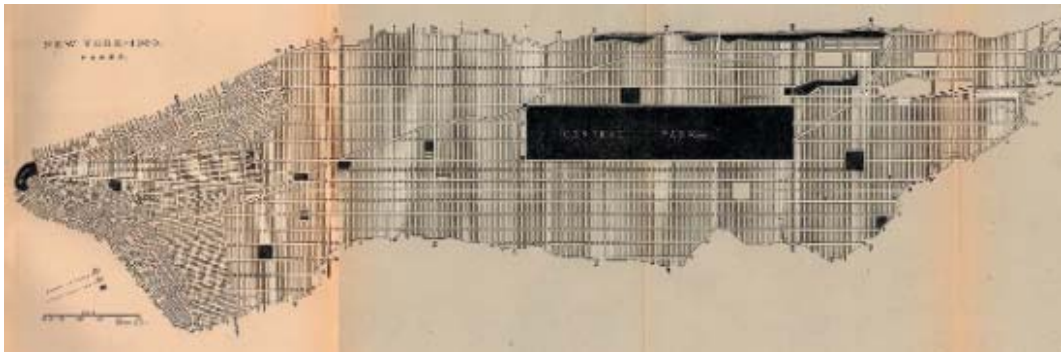


Fig. 0.9 Map of New York in 1880. From Report on the Social Statistics of Cities, Compiled by George E. Waring, Jr., U.S. Census Office, 1886. Image courtesy of University of Texas Libraries.

Data about the yeast protein interaction network is from Jeong *et al.* (2001). *C. elegans* neural network data is from White *et al.* (1986). To learn about the human brain network, or "connectome," we recommend Sporns (2012). The Everglades ecological network is derived from Ulanowicz and DeAngelis (1998). To learn more about food webs, we refer to Dunne *et al.* (2002) and Melián and Bascompte (2004).

Data for several of the real-world network examples shown in this book is provided by the Network Repository (Rossi and Ahmed, 2015). The visualizations are done using Gephi (Bastian *et al.*, 2009). Layout algorithms are discussed in Chapter 1.

Exercises

0.1 Consider the road map in Figure 0.9. If one were creating a network representation of traffic patterns, which of the following would be the best choice to make up the links of the network? (*Hint*: Your answer to the next question may inform your answer to this question, and vice versa.)

- **a.** Pedestrians traveling along the streets
- **b.** Road segments (e.g. 5th Ave. between 12th and 13th streets)
- **c.** Entire roads (e.g. 5th Ave.)
- **d.** Vehicles traveling on the roads

0.2 Consider the road map in Figure 0.9. In a network representation of traffic patterns, which of the following would be the best choice to make up the nodes of the network? (*Hint*: Your answer to the previous question may inform your answer to this question, and vice versa.)

- **a.** City blocks (e.g. the block between 5th–6th avenues and 12th–13th streets)
- **b.** Street intersections (e.g. 5th Ave. and 12th St.)
- **c.** Pedestrians moving along the streets
- **d.** Vehicles traveling on the roads

0.3 Consider the US air transportation network shown in Figure 0.7. Nodes in this network represent airports. What could a link between two airports represent?

0.4 Compare the US air transportation network in Figure 0.7 with the Manhattan road map in Figure 0.9. The air transportation network displays a distinguishing feature that the Manhattan road network lacks. What is this key characteristic?

a. Singleton nodes with no links
b. Multiple routes between nodes
c. Nodes with more than one connected link
d. Hub nodes with many links

0.5 In a social graph from Facebook, which type of link best represents the "friend" relation? Directed or undirected?

0.6 In a social graph from Twitter, which type of link best represents the "follower" relation? Directed or undirected?

1 Network Elements

node: (*n.*) a point in a network or diagram at which lines or pathways intersect or branch.

Having seen several examples of real networks in Chapter 0, let us now learn about the basic definitions and quantities that allow us to describe a network.

1.1 Basic Definitions

In very general terms a network, or graph, is a set of elements, which we call *nodes*, along with a set of connections between pairs of nodes, which we call *links*. The links represent the presence of a relation among the elements represented by the nodes. As we have seen earlier, links can correspond to social, physical, communication, geographic, conceptual, chemical, biological, or other interactions. We say that two nodes are *adjacent* or *connected* if there is a link between them. It is also common to call connected nodes *neighbors*.

Networks provide a general theoretical framework allowing for a convenient conceptual representation of interrelations in a wide array of systems; we have seen several examples in Chapter 0. The study of networks has a long tradition in mathematics, computer science, sociology, and communications research. Recently, networks have also been studied intensely in physics and biology. Different fields concerned with networks often introduce their own nomenclature. For example, in some fields a network is called a *graph*,

Box 1.1 Definition of a Network

A network G has two parts, a set of N elements, called *nodes* or *vertices*, and a set of L pairs of nodes, called *links* or *edges*. The link (i, j) joins the nodes i and j. A network can be *undirected* or *directed*. A directed network is also called a *digraph*. In directed networks, links are called *directed links* and the order of the nodes in a link reflects the direction: the link (i, j) goes from the source node i to the target node j. In undirected networks, all links are bi-directional and the order of the two nodes in a link does not matter. A network can be *unweighted* or *weighted*. In a weighted network, links have associated *weights*: the *weighted link* (i, j, w) between nodes i and j has weight w. A network can be both directed and weighted, in which case it has directed weighted links.

a node is referred to as a *vertex*, and a link is an *edge*. (We will occasionally use these terms.) The rigorous language for the description of networks is found in graph theory, a field of mathematics that can be traced back to the pioneering work of Leonhard Euler in the eighteenth century. Here we do not want to provide a rigorous introduction to graph theory. We are mostly interested in building a vocabulary and introducing a set of basic notions that will allow us to take our first steps into the world of networks. However, sometimes a formal notation is helpful. In these cases we will include the formal notation in a shaded area or in a box. For example, a more rigorous definition of a network is provided in Box 1.1. In the following chapters we will introduce additional concepts and definitions as needed to analyze real-world systems.

Each network is characterized by the total number of nodes N and the total number of links L. We call N the *size* of the network because it identifies the number of distinct elements composing the system. The numbers of nodes and links do not suffice in defining a network; we have to specify the way in which the nodes are connected by the links.

There are different types of links, which define different classes of networks. In some networks, such as Facebook (Figure 0.1), the links do not have a direction and we represent them as line segments. We call these networks *undirected*. In other cases, such as Wikipedia (Figure 0.5), links are directed and we represent them as arrows. Networks with directed links are called *directed networks*. We say more about directed networks in Section 1.6 and Chapter 4.

In some cases, such as air transportation networks (Figure 0.7), links have associated weights. These are called *weighted networks*. A network can be both directed and weighted. The email network is an example of a weighted directed network, in which link weights and directions represent communication traffic (number of messages) between nodes. We will

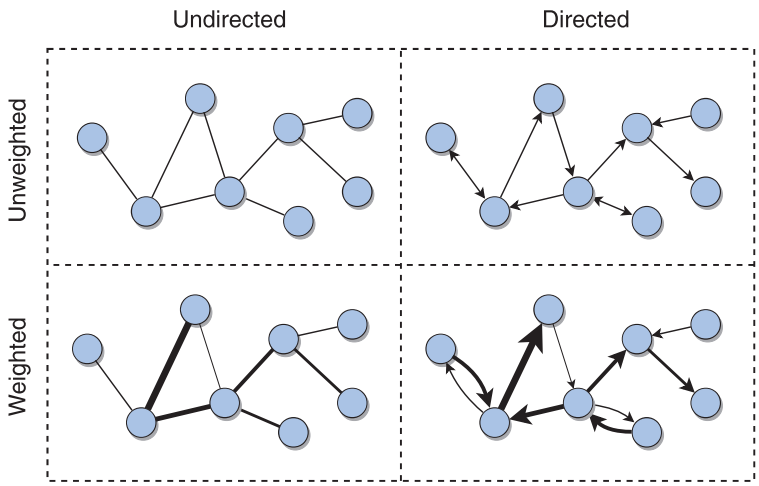


Fig. 1.1 Graphical representations of undirected, directed, and weighted networks. The circles represent the nodes. Pairs of adjacent nodes are connected by a line (link) or arrow (directed link). Arrows indicate the direction of the links. The thickness of a link represents its weight in weighted networks.

return to weighted networks in Section 1.7 and Chapter 4. Figure 1.1 provides illustrations of undirected, directed, and weighted networks.

There are several other classes of networks. A network might have more than one type of node. For example, the movie-star network [Figure 0.2(a)] has two types of nodes representing movies and people. In this network, a link connects an actor or actress to a movie, but there are no links among people or among movies. This is an instance of a so-called *bipartite network*. In a bipartite network, there are two groups of nodes such that links only connect nodes from different groups and not nodes from the same group. Other examples of bipartite networks include those that capture the relationships between songs and artists, between classes and students, and between products and customers. More on bipartite networks in Chapter 4.

A network might have multiple types of links, in which case it is called a *multiplex* network. To use the movie-star example again, we could imagine adding links between actors and/or actresses who are married to each other. In the example of Wikipedia (Figure 0.5), in addition to the hyperlinks, we might have weighted links representing clicks from Wikipedia users, and/or undirected links between articles that share editors. These and other more complex types of networks are discussed further in Section 1.8.

1.2 Handling Networks in Code

To manage, analyze, and visualize networks with more than a handful of nodes and links, we need to use software tools or write our own code. There are many network analysis and visualization tools, as well as libraries to handle networks in many programming languages. Throughout the book we will occasionally mention a couple of these tools. For instance, the visualizations in Chapter 0 are generated with an application called *Gephi*. However, we believe that to get a hands-on understanding of networks it is necessary to "get our hands dirty" and write some code. We assume that students using this book have some familiarity with Python, a popular programming language among both novice and expert coders.[1] To make life easier, we will use *NetworkX* (networkx.github.io), a Python package for the creation, manipulation, and study of the structure, dynamics, and function of networks. NetworkX provides data structures, algorithms, measures, and generators for networks, as well as rudimentary visualization facilities.[2]

Once we import NetworkX, we can easily create an undirected network ("Graph") and add a few nodes and links. Nodes are referred to by integer IDs and links are called edges:

[1] We offer an introductory tutorial on Python in Appendix A; it can also be downloaded from the book's GitHub repository at github.com/CambridgeUniversityPress/FirstCourseNetworkScience.

[2] We offer an introductory tutorial on NetworkX on the book's GitHub repository.

```
import networkx as nx # always import NetworkX first!
G = nx.Graph()
G.add_node(1)
G.add_node(2)
G.add_edge(1,2)
```

We can add several nodes or links at once:

```
G.add_nodes_from([3,4,5,...])
G.add_edges_from([(3,4),(3,5),...])
```

Here is how we get lists of nodes, links, and neighbors of a given node:

```
G.nodes()
G.edges()
G.neighbors(3)
```

And here is how you loop over nodes or links:

```
for n in G.nodes:
    print(n, G.neighbors(n))
for u,v in G.edges:
    print(u, v)
```

Similarly, we can create a directed network ("DiGraph"):

```
D = nx.DiGraph()
D.add_edge(1,2)
D.add_edge(2,1)
D.add_edges_from([(2,3),(3,4),...])
```

Note that the link from node **1** to node **2** is distinct from the link from node **2** to node **1** because this network is directed. Also note that when we add a link, the nodes are added automatically if they don't already exist. This is convenient. There are functions for getting the size and number of links:

```
D.number_of_nodes()
D.number_of_edges()
```

In a directed network, when we ask for the neighbors of a node, we get both the nodes linking to and from it. But there are also functions to get only the edges linking to or from, respectively called predecessors and successors:

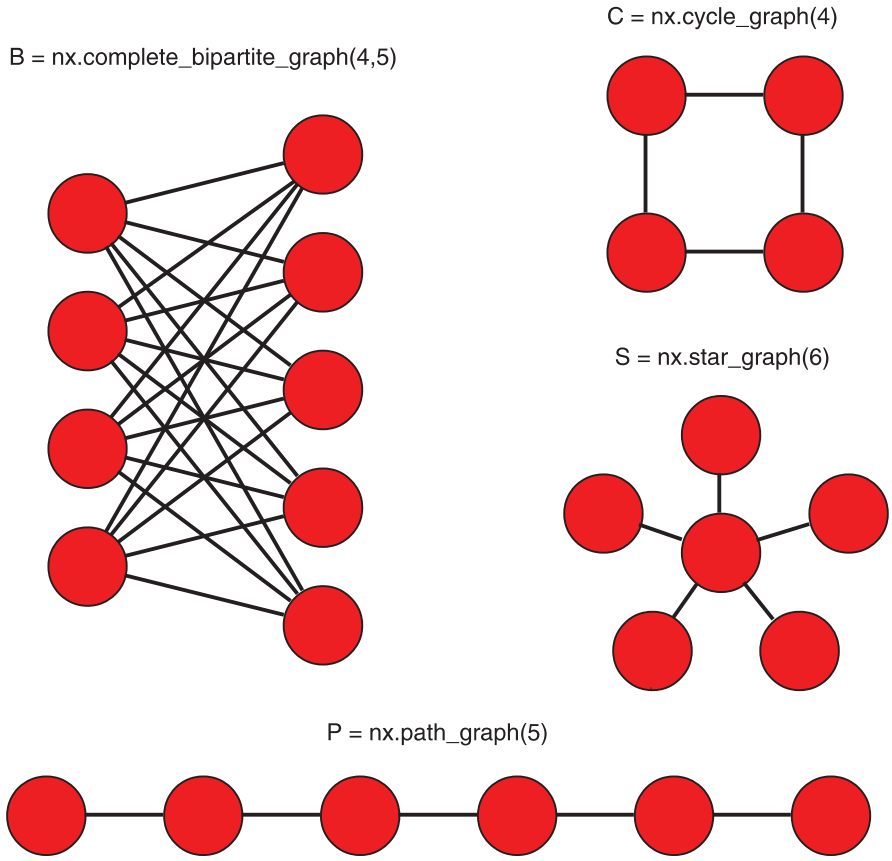


Fig. 1.2 A few simple networks generated by NetworkX functions: complete bipartite (B), cycle (C), star (S), and path (P). The concept of a *complete* network is introduced in the next section.

```
D.neighbors(2)
D.predecessors(2)
D.successors(2)
```

Finally, there are functions to generate networks of many types. Typically these functions need arguments that specify the number of nodes or links. Here is code to generate a few networks, shown in Figure 1.2:

```
B = nx.complete_bipartite_graph(4,5)
C = nx.cycle_graph(4)
P = nx.path_graph(5)
S = nx.star_graph(6)
```

We strongly recommend that you read the NetworkX tutorial[3] and bookmark its documentation.[4] And remember, Google and Stack Overflow are your friends when you are stuck!

[3] networkx.github.io/documentation/stable/tutorial.html
[4] networkx.github.io/documentation/stable/

1.3 Density and Sparsity

The maximum number of links in a network is bounded by the possible number of distinct connections among the nodes of the system. The maximum number of links is therefore given by the number of pairs of nodes. A network with the maximum number of links, in which all possible pairs of nodes are connected by links, is called a *complete network*.

The maximum number of links in an undirected network with N nodes is the number of distinct pairs of nodes:

$$L_{max} = \binom{N}{2} = N(N-1)/2. \tag{1.1}$$

Intuitively each node can connect to $N-1$ other nodes, and there are N of them. However, that would count each pair twice, so we divide by two. In a directed network, each pair of nodes should be counted twice, once for each direction, so $L_{max} = N(N-1)$. Counting the possible pairs of objects among a set of N objects is something that we will encounter again later in the book. Mathematicians have a name for the formula $\binom{N}{2}$: "N choose two."

A bipartite network is *complete* if each node in one group is connected to all nodes in the other group (see example B in Figure 1.2). In this case $L_{max} = N_1 \times N_2$, where N_1 and N_2 are the sizes of the two groups.

The fraction of possible links that actually exist, which is the same as the fraction of pairs of nodes that are actually connected, is called the *density* of the network. A complete network has maximal density: one. However, the actual number of links is typically much smaller than the maximum, as most pairs of nodes are not directly connected to each other. Therefore the density is often much, much smaller than one — by orders of magnitude in most real-world, large networks. This is an important feature that helps in dealing with network structure. We call it *sparsity*. Intuitively the fewer edges are in a network, the sparser it is.

The density of a network with N nodes and L links is

$$d = L/L_{max}. \tag{1.2}$$

In an undirected network this is given by

$$d = L/L_{max} = \frac{2L}{N(N-1)} \tag{1.3}$$

and in a directed network the density is

$$d = L/L_{max} = \frac{L}{N(N-1)}. \tag{1.4}$$

Table 1.1 Basic statistics of network examples. Network types can be (D)irected and/or (W)eighted. When there is no label, the network is undirected and unweighted. For directed networks, we provide the average in-degree (which coincides with the average out-degree)

Network	Type	Nodes (N)	Links (L)	Density (d)	Average degree ($\langle k \rangle$)
Facebook Northwestern Univ.		10,567	488,337	0.009	92.4
IMDB movies and stars		563,443	921,160	0.000006	3.3
IMDB co-stars	W	252,999	1,015,187	0.00003	8.0
Twitter US politics	DW	18,470	48,365	0.0001	2.6
Enron email	DW	87,273	321,918	0.00004	3.7
Wikipedia math	D	15,220	194,103	0.0008	12.8
Internet routers		190,914	607,610	0.00003	6.4
US air transportation		546	2,781	0.02	10.2
World air transportation		3,179	18,617	0.004	11.7
Yeast protein interactions		1,870	2,277	0.001	2.4
C. elegans brain	DW	297	2,345	0.03	7.9
Everglades ecological food web	DW	69	916	0.2	13.3

In a complete network, $d = 1$ by definition, since $L = L_{max}$. In a sparse network, $L \ll L_{max}$ and therefore $d \ll 1$. When a network grows very large, we can observe how the number of links increases as a function of the number of nodes. We say that the network is sparse if the number of links grows proportionally to the number of nodes ($L \sim N$), or even slower. If instead the number of links grows faster, e.g. quadratically with network size ($L \sim N^2$), then we say that the network is dense.

To illustrate the importance of network sparsity, let us consider the example of Facebook. At the time of writing, Facebook has around 2 billion users ($N \approx 2 \times 10^9$). If this was a complete network, there would be $L \approx 10^{18}$ links — that is a number with 18 zeros, and there is no way to store so much data! But fortunately, social networks are very sparse and Facebook is no exception. Each user has on average 1000 friends or less, so that the density is approximately $d \approx 10^{-6}$. That is still a lot of data, but Facebook can manage it.

Table 1.1 presents basic statistics about the size and density of the network examples illustrated in Chapter 0.[5] Although these networks are very different from each other, they are all sparse.

NetworkX makes it easy to measure the density of directed and undirected networks:

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

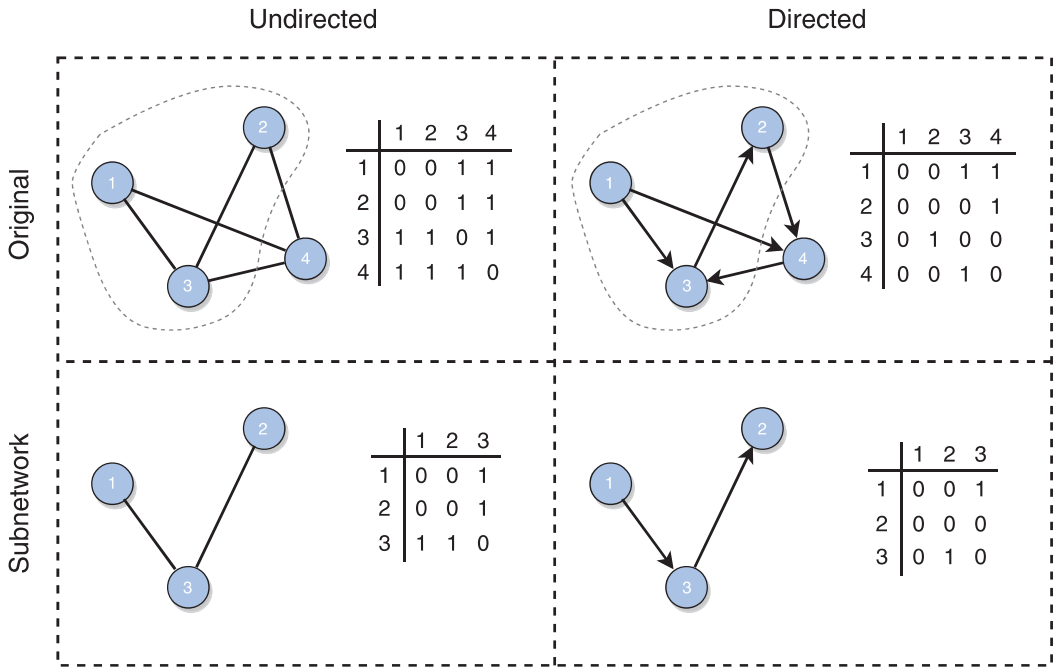


Fig. 1.3 Network and subnetwork examples. We also show the adjacency matrix representation of each network (see Section 1.9).

```
nx.density(G)
nx.density(D)
CG = nx.complete_graph(8471) # a large complete network
print(nx.density(CG))        # no need for a calculator!
```

1.4 Subnetworks

In many cases, we are interested in a subset of a network, which is itself a network and is called a *subnetwork* (or *subgraph*). A subnetwork is obtained by selecting a subset of the nodes and *all* of the links among these nodes.

Figure 1.3 provides some illustrations of subnetworks of undirected and directed networks. The abundance of certain types of subnetworks and their properties is important in the characterization of real networks. As an example, a *clique* is a complete subnetwork: a subset of nodes all linked to each other. Any subnetwork of a complete network is a clique because all pairs of nodes in the network are connected and therefore all pairs of nodes in any subnetwork are also connected.

A special type of subnetwork is the *ego network* of a node, which is the subnetwork consisting of the chosen node — called the *ego* — and its neighbors. Ego networks are often studied in social network analysis.

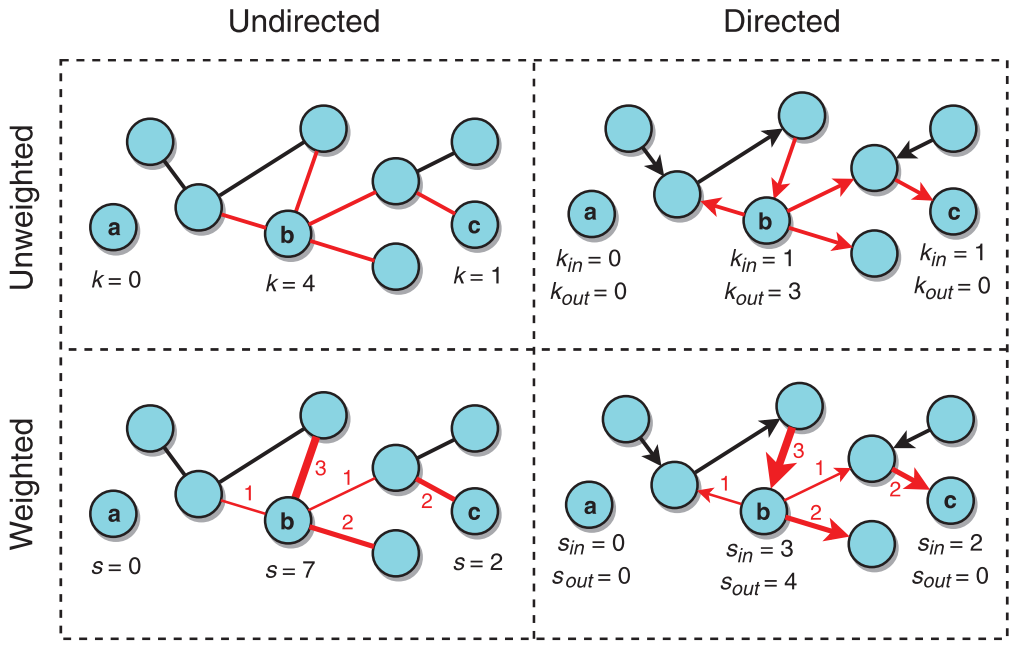


Fig. 1.4 Illustrations of degree and strength in directed, undirected, weighted, and unweighted networks. The links of nodes **a**, **b**, and **c** along with their weights are highlighted in red, and their degrees or strengths are shown.

Using NetworkX we can generate a subnetwork of a given network by specifying a subset of nodes:

```
K5 = nx.complete_graph(5)
clique = nx.subgraph(K5, (0,1,2))
```

1.5 Degree

The *degree* of a node is its number of links, or neighbors. We use k_i to denote the degree of node i. Figure 1.4 illustrates the degree of a few nodes in an undirected network. A node with no neighbors, such as node **a** in the figure, has degree zero ($k = 0$) and is called a *singleton*.

The average degree of a network is denoted by $\langle k \rangle$. It is an important property and is related (directly proportional) to its density.

The average degree of a network is defined as

$$\langle k \rangle = \frac{\sum_i k_i}{N}. \tag{1.5}$$

Since each link contributes to the degree of two nodes in an undirected network, the numerator of Eq. (1.5) can be written as $2L$. From the definition of density for an undirected network [Eq. (1.3)], $2L = dN(N-1)$. Therefore

$$\langle k \rangle = \frac{2L}{N} = \frac{dN(N-1)}{N} = d(N-1) \tag{1.6}$$

and conversely

$$d = \frac{\langle k \rangle}{N-1}. \tag{1.7}$$

This makes sense: the maximum possible degree of a node is $k_{max} = N - 1$, obtained when the node is connected to every other node. Intuitively, the density is the ratio between the average and maximum degree.

Table 1.1 shows the average degree of the network examples illustrated in Chapter 0. NetworkX has a function that returns the degree of a given node. Without arguments, it returns a dictionary with the degree of each node:

```
G.degree(2) # returns the degree of node 2
G.degree()  # returns the degree of all nodes of G
```

In Chapter 3 we will see that the degrees of a network's individual nodes are very important properties to characterize the structure of the network. So far we have defined the degree in undirected networks. Next we extend the definition to directed and weighted networks.

1.6 Directed Networks

In the graphical representation of a network, the directed nature of the links is depicted by means of an arrow, indicating the direction of each link. The main difference between directed and undirected networks is represented in Figure 1.1. In an undirected network, the presence of a link between two nodes connects the adjacent nodes in both directions. In contrast, the presence of a link in a directed network does not necessarily imply the presence of a link in the opposite direction. This fact has important consequences for the connectedness of a directed network, as will be discussed in more detail in Chapter 2.

When we consider the degree of a node in a directed network, we have to think of incoming and outgoing links separately. The number of incoming links, or predecessors, of node i is called the *in-degree* and denoted by k_i^{in}. The number of outgoing links, or successors, of node i is called the *out-degree* and denoted by k_i^{out}. Figure 1.4 illustrates the in- and out-degrees of a few nodes in a directed network.

We already defined the density for a directed network [Eq. (1.4)]. We can define average in-degree and average out-degree similarly to Eq. (1.5).

NetworkX has functions that return the in-degree and out-degree of a given node. If the network is directed, the `degree` function returns the total degree, which is the sum of in-degree and out-degree:

```
D.in_degree(4)
D.out_degree(4)
D.degree(4)
```

1.7 Weighted Networks

In the graphical representation of a network, the weighted nature of the links is depicted by means of lines of different width, indicating the weight of each link. A weight of zero is equivalent to the absence of a link. The main difference between weighted and unweighted networks is represented in Figure 1.1.

A weighted network can be directed or undirected; let us first assume the simpler case of an undirected weighted network. We can measure the degree of a node in a weighted network by disregarding the weights. However, it may be important to consider the weights. We can therefore define the *weighted degree*, or *strength* of a node, as the sum of the weights of its links. Similarly, we can define *in-strength* and *out-strength* in the case of a directed weighted network. Both cases are illustrated in Figure 1.4.

The weighted degree, or *strength*, of node i in an undirected weighted network is denoted by

$$s_i = \sum_j w_{ij}, \tag{1.8}$$

where w_{ij} is the weight of the link between nodes i and j. We assume $w_{ij} = 0$ if there is no link between i and j. We can analogously generalize in-degree and out-degree to in-strength and out-strength in a directed weighted network:

$$s_i^{in} = \sum_j w_{ji}, \tag{1.9}$$

$$s_i^{out} = \sum_j w_{ij}, \tag{1.10}$$

where w_{ij} is the weight of the directed link from i to j.

In NetworkX, both graphs and digraphs can have "weight" attributes attached to links. When adding multiple weighted links, each is specified as a triple where the third element is the weight:

```
W = nx.Graph()
W.add_edge(1,2,weight=6)
W.add_weighted_edges_from([(2,3,3),(2,4,5)])
```

We can get a list of links with their associated weight data, for example if we need to print the links with large weights:

```
for (i,j,w) in W.edges(data='weight'):
    if w > 3:
        print('(%d, %d, %d)' % (i,j,w)) # skip link (2,3)
```

Finally, we can get the strength of a given node using the `degree` function and specifying the weight attribute:

```
W.degree(2, weight='weight') # strength of node 2
                             # is 6 + 3 + 5 = 14
```

1.8 Multilayer and Temporal Networks

In the US air transportation network of Figure 0.7, the links represent direct flights between airports, regardless of which particular airlines operate those flights. But classifying the flights according to their respective airlines is valuable in a number of situations. We may wish to predict the propagation of scheduling delays through an airline's network, or investigate the consequences of such delays on the movement of passengers. In fact, each commercial airline tries to reschedule passengers on its own flights first because it is expensive to rebook them on another company's flights. Therefore the air transportation network of a specific airline has its own identity, even though it is intertwined with the networks of other airlines. In these cases it is beneficial to represent the system as a *multilayer network* (i.e. a combination of layers), where each layer is the air transportation network of a specific airline: the nodes are the airports, the links flights operated by the same company.

If each layer in a multilayer network is built upon the same set of nodes, the network is called a *multiplex*. The air transportation network is an example of a multiplex. Another example is a social network in which the different layers represent different types of social relationships. For example, one layer could represent friendship ties, another layer family ties, another coworker ties, and so on. The nodes in each layer represent the same individuals.

A *temporal network* is a special case of a multiplex. Links are dynamic, in that the respective node–node interactions occur at different times. Nodes may also have a dynamic character, in that they may appear and disappear at different stages of the network evolution. For instance, networks of user activity on Twitter are temporal because posts, retweets,

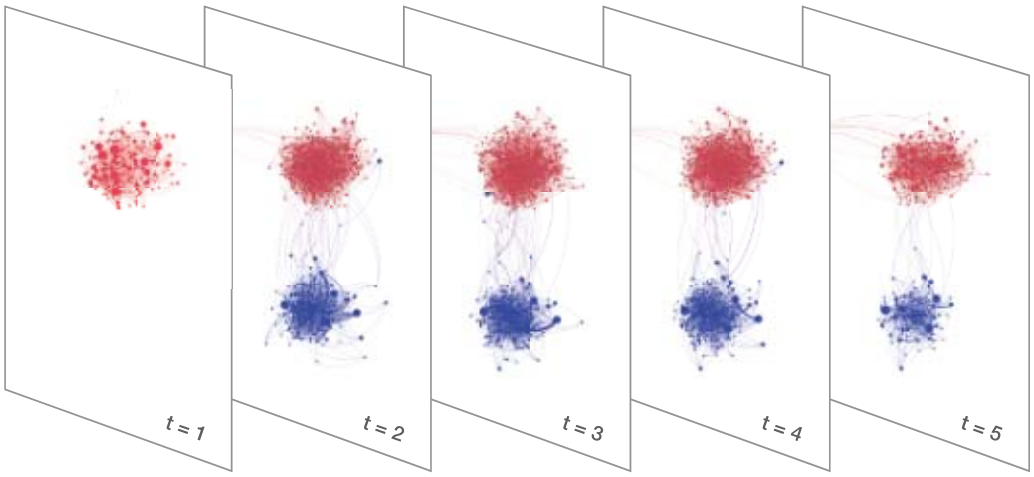


Fig. 1.5 Temporal network of political retweets. Each snapshot includes retweet links with timestamps in a particular time interval. By aggregating these snapshots over time, we obtain the static network shown in Figure 0.3.

and mentions occur at different times, which can be identified by their timestamps. We can divide the time span of a temporal network into consecutive intervals: all nodes and links existing during each interval constitute a *snapshot* of the system. Each snapshot can be interpreted as one layer of a multiplex, as illustrated in Figure 1.5.

In a multilayer network there are *intralayer links*, connecting pairs of nodes in the same layer, and *interlayer links*, connecting pairs of nodes in different layers. In the special case of multiplex networks, interlayer links connect each node of a layer with its counterpart in the other layers. Such links are called *couplings*, because they couple copies of the same node in different layers.

Traditionally, multiplex networks have been analyzed by aggregating data from the different layers and then studying the resulting network. For instance, the networks of Figures 0.3 and 0.7 are aggregations of multiplex networks corresponding to time intervals and different airlines, respectively. The aggregated network is typically weighted, even when the links of the multiplex are not, because there are usually multiple links joining the same pair of nodes in different layers, which turn into a single weighted link in the aggregated system. For example, the links in Figure 0.3 are weighted by the number of times a user retweets another. But aggregation discards a lot of valuable information provided by the original multilayer system. In the air transportation case, merging networks corresponding to different airlines prevents us from studying transitions of passengers between such networks, which may become necessary in case of strikes or technical problems affecting a specific airline.

In general, each layer could be characterized by its own set of nodes and links. Therefore, layers may represent entirely different graphs and the resulting system is a *network of networks*. Here, interlayer links may represent dependency relationships between the nodes of the networks. Consider the electrical power grid, which connects power-generating stations and demand centers through high-voltage transmission lines. The stations are controlled by computers that monitor and manage the production and transmission of electricity. These

computers are connected through the Internet. In turn, Internet routers depend on power stations for their electricity supply. Therefore we have a system with two coupled networks: the power grid and the Internet.

In such a coupled system, one network can affect the other to optimize delivery; the grid can be reconfigured to reroute power when needed. However, this kind of network of networks can also introduce unpredictable vulnerabilities. A software problem or attack can take down one or more nodes in the power grid, and without electricity the Internet in an area could also go down, leading to failures of other nodes and, in an extreme case, a catastrophic domino effect called a *cascading failure* affecting a large portion of the grid. For these reasons, networks of networks are the subject of intense study.

To keep things simple, in this book we focus mainly on networks with a single type of node and a single type of link. In an undirected network, we will assume that there can be at most one link connecting a pair of nodes. (If the network is directed, there can be two links, one in each direction, as shown in Figure 1.1.) In addition, we will not consider *self-loops*, or links connecting a node to itself; we will assume that each link connects two distinct nodes.

1.9 Network Representations

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.

In Figure 1.3 we show the graphical illustrations of different undirected and directed networks and their corresponding adjacency matrices.

For undirected networks the adjacency matrix is symmetric: we can swap rows and columns and the matrix does not change. Therefore, half of the matrix contains redundant information. For directed networks, the adjacency matrix is not symmetric. For unweighted networks, the elements take only values one or zero to indicate the presence or absence of a link, respectively. For weighted networks, matrix elements can take any values corresponding to the link weights. We have already encountered the adjacency matrix elements for weighted networks [w_{ij} in Eqs (1.8)–(1.10)].

In NetworkX, we can get and print adjacency matrices and use the matrix representation to get and set link attributes:

```
print(nx.adjacency_matrix(G)) # graph
G.edge[3][4]
G.edge[3][4]['color']='blue'
```

```
print(nx.adjacency_matrix(D)) # digraph
D.edge[3][4]
D.edge[4][3] # not the same as the previous one
print(nx.adjacency_matrix(W)) # weighted graph
W.edge[2][3]
W.edge[2][3]['weight'] = 2
```

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.

NetworkX provides facilities to loop over a network's adjacency list and retrieve links and their attributes. For example, here is one way to print the neighbors of each node:

```
for n,neighbors in G.adjacency():
    for number,link_attributes in neighbors.items():
        print('(%d, %d)' % (n,number))
```

A third, equally efficient network representation is the *edge list*, which lists each link as a pair of connected nodes. We may also need to list the nodes separately in case of singletons, which would not appear in any of the pairs. In the case of weighted networks, each link is represented as a triple, where the third element is the weight.

In this book we will use the edge list representation to store networks. NetworkX has functions to write and read network files using this representation. You can view the format of an edge list file for yourself:

```
nx.write_edgelist(G, "file.edges")
G2 = nx.read_edgelist("file.edges")        # G2 same as G
nx.write_weighted_edgelist(W, "wf.edges") # store weights
with open("wf.edges") as f:
    for line in f:
        print(line)
W2 = nx.read_weighted_edgelist("wf.edges") # W2 same as W
```

1.10 Drawing Networks

We can learn a lot about a network by drawing it and inspecting its graphical representation. This requires a *network layout algorithm* to place each node on a plane. (There are also

sophisticated 3D layouts, but we do not discuss them in this book.) There are many layout algorithms that are appropriate for representing different kinds of networks; for example, we used a *geographic layout* to draw the air transportation network in Figure 0.7. For relatively small networks, layouts that place nodes along concentric circles or layers can reveal important hierarchical structure. The most popular class of network layout algorithms are *force-directed layout algorithms*, which are used to visualize most of the example networks in Chapter 0. The inset of Figure 0.7 uses a force-directed layout as well.

The goals of a force-directed layout algorithm are to place the nodes so that connected nodes are positioned close to each other, all the links are of similar length, and the number of link crossings is minimized. To get an idea of how force-directed layout works, imagine a force that repels any two nodes from each other, like the force between two particles with the same electrical charge. Further imagine a spring connecting any two linked nodes, generating an attractive force when they are too far from each other. Force-directed layout algorithms simulate such a physical system so that nodes move to minimize the energy of the system: connected nodes will move toward each other and away from nodes not connected to them.

The result is not only an aesthetically pleasing drawing, but also, sometimes, a visualization of the most obvious communities in the network, as we have seen in Chapter 0. For example, in Figure 0.3, because people in a community (progressive or conservative) are densely connected to each other, they end up clustered together in the layout.

NetworkX has a function to draw a network, which uses a rudimentary network layout algorithm:

```
import matplotlib.pyplot
nx.draw(G)
```

Note that drawing requires a plot interface, such as Matplotlib's. This works reasonably well for small networks with, say, less than 100 nodes. For larger networks, there are better visualization tools. The examples in Chapter 0 are visualized with Gephi's *ForceAtlas2* layout algorithm.

1.11 Summary

We have presented some basic definitions and quantities that allow us to describe a network:

1. A network is made up of two sets of elements: the nodes and links connecting pairs of nodes.
2. A subnetwork is a subset of the network including some of its nodes and all of the links among them.
3. In directed networks, links have a direction. There may be a link from node **1** to node **2**, and not necessarily one from node **2** to node **1**. In undirected networks, links are reciprocal.

4. In weighted networks, links have associated weights that represent connection attributes like importance, similarity, distance, traffic, etc. In unweighted networks, all links are the same.
5. Multilayer networks have different types of nodes and links, divided into interconnected layers. If the nodes are the same in each layer, the multilayer network is called a multiplex.
6. The density of a network is the fraction of node pairs that are connected. A network is complete if all pairs of nodes are connected, so that the density is one. Most real networks are sparse, meaning that they have very small density.
7. The degree of a node is the number of neighbors. In directed networks, nodes have in-degree and out-degree measuring the number of incoming and outgoing links, respectively. If the network is weighted, the strength of a node is the sum of the weights of its links. The nodes of weighted directed networks have in-strength and out-strength.
8. Adjacency lists and edge lists are efficient representations to store sparse networks.
9. NetworkX is a popular and convenient programming library to code networks in the Python language.

The definitions in this chapter form a basic vocabulary for network science. More quantities and properties will be introduced in future chapters so that we can describe, analyze, and model real networks and learn what they tell us about the underlying systems and phenomena.

1.12 Further Reading

There are several other excellent textbooks on network science to go beyond the introductory material in this book. Caldarelli and Chessa (2016) dig a bit deeper into the data science of several case studies. If you are interested in branching into physics, consider the textbook by Barabási (2016); if you want to explore the connections to economics and sociology, we recommend the textbook by Easley and Kleinberg (2010). For more advanced physics, math, and social science topics, there are many books to choose from (Wasserman and Faust, 1994; Caldarelli, 2007; Barrat *et al.*, 2008; Cohen and Havlin, 2010; Bollobás, 2012; Dorogovtsev and Mendes, 2013; Latora *et al.*, 2017; Newman, 2018).

Kivelä *et al.* (2014) and Boccaletti *et al.* (2014) have provided influential reviews on multilayer networks. Temporal networks are reviewed by Holme and Saramäki (2012). Gao *et al.* (2012) analyze networks of networks. Catastrophic failure in these networks is discussed by Reis *et al.* (2014) and Radicchi (2015).

For background on network drawing, see Di Battista *et al.* (1998). Force-directed network layout (also known as spring layout) algorithms were introduced by Eades (1984) and improved by Kamada and Kawai (1989) and Fruchterman and Reingold (1991). The ForceAtlas2 layout algorithm, used for many visualizations in this book, was developed by Jacomy *et al.* (2014).

Exercises

1.1 Go through the Chapter 1 Tutorial on the book's GitHub repository.[6]

1.2 Consider a network with N nodes. Given a single link, what is the maximum number of nodes that link can connect? Given a single node, what is the maximum number of links that can connect to that node?

1.3 Consider the road map in Figure 0.9. The grid-like structure of this network means that most nodes have the same degree. What is the most common degree for nodes in this network?

1.4 Consider the road map in Figure 0.9. Manhattan has a lot of one-way streets. This implies that a good network model of traffic flow would probably have directed links. Consider a subgraph of this network with grid-like connectivity and all one-way streets (i.e. each node is a four-way intersection of two one-way streets). What is the most common in-degree of nodes in this subgraph? What is the most common out-degree?

1.5 What network quantity can we use to represent the volume of traffic between each pair of adjacent intersections in the Manhattan road map (Figure 0.9)?

1.6 Consider a directed network of N nodes. Now consider the total in-degree (i.e. the sum of the in-degree over all nodes in the network). Compare this to the analogous total out-degree. Which of the following must hold true for any such network?

a. Total in-degree must be less than total out-degree
b. Total in-degree must be greater than total out-degree
c. Total in-degree must be equal to total out-degree
d. None of these hold true in all instances

1.7 Consider a Twitter retweet network, where users are nodes and we want to show how many times a given user has retweeted another user. What link type best captures this relation?

a. Undirected, unweighted
b. Undirected, weighted
c. Directed, unweighted
d. Directed, weighted

1.8 Consider a hashtag co-occurrence graph from Twitter. In this network, hashtags are the nodes, and a link between two hashtags indicates how often those two hashtags appear in tweets together. What link type would best capture this relation?

a. Undirected, unweighted
b. Undirected, weighted
c. Directed, unweighted
d. Directed, weighted

[6] github.com/CambridgeUniversityPress/FirstCourseNetworkScience

1.9 Consider a network created from characters in a story or play. The nodes are people, and a link exists between two nodes if those characters ever engage in dialogue. Which type of edge could represent this relation? Justify your answer.

a. Undirected, unweighted
b. Undirected, weighted
c. Directed, unweighted
d. Directed, weighted

1.10 Suppose we want to make a more complex version of a dialog network that captures how much each character speaks and to whom. What type of link would best represent this relation?

a. Undirected, unweighted
b. Undirected, weighted
c. Directed, unweighted
d. Directed, weighted

1.11 Imagine that your social network has a subnetwork where you and 24 of your friends (25 people total) are all friends with each other. What is such a subnetwork called? And how many links are contained in the subnetwork?

1.12 Consider an undirected network with N nodes. What is the maximum number of links this network can have?

1.13 Consider a bipartite network of N nodes, N_1 nodes of type 1 and N_2 nodes of type 2 (so that $N_1 + N_2 = N$). What is the maximum number of links in this network?

1.14 Given a complete network A with N nodes, and a bipartite network B also with N nodes, which of the following holds true for any $N > 2$:

a. Network A has more links than network B
b. Network A has the same number of links as network B
c. Network A has fewer links than network B
d. None of these hold true for all such $N > 2$

1.15 Recall that in a complete network there exists a link between each pair of nodes. We know that a complete undirected network of N nodes has $N(N-1)/2$ edges. Must any undirected network of N nodes and $N(N-1)/2$ links be complete? Explain why or why not.

1.16 Consider this adjacency matrix:

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 2 & 1 & 3 & 1 & 1 & 0 \end{pmatrix} \end{array}. \tag{1.11}$$

An entry in the ith row and jth column indicates the weight of the link from node i to node j. For instance, the entry in the second row and third column is 2, meaning the weight of the link from node **B** to node **C** is 2. What kind of network does this matrix represent?

a. Undirected, unweighted
b. Undirected, weighted
c. Directed, unweighted
d. Directed, weighted

1.17 Consider the network defined by the adjacency matrix in Eq. (1.11). How many nodes are in this network? How many links? Are there any self-loops?

1.18 Consider the network defined by the adjacency matrix in Eq. (1.11). Are there any nodes with outgoing links to every other node? If so, which nodes? Are there any nodes with in-links from every other node? If so, which nodes?

1.19 Consider the network defined by the adjacency matrix in Eq. (1.11). A *sink* is defined as a node with in-links but no out-links. Which nodes in the network, if any, have this property?

1.20 Consider the network defined by the adjacency matrix in Eq. (1.11). What is the in-strength of node **C**? What is its out-strength?

1.21 Convert the network defined by the adjacency matrix in Eq. (1.11) to an undirected, unweighted graph. (When converting a directed graph to an undirected one, nodes i and j are connected in the undirected graph if there is a directed link from i to j, or from j to i, or both.) You may want to print out the resulting matrix and/or draw a network diagram for reference. How many nodes are in this converted network? How many links?

1.22 Consider the unweighted, undirected version of the network defined by the adjacency matrix in Eq. (1.11), constructed as explained in Exercise 1.21. What is the minimum degree in this network? What is the maximum degree? What is the mean degree? What is the density?

1.23 Imagine two different undirected networks, each with the same number of nodes and links. Must both networks have the same maximum and minimum degree? Explain why or why not. Must they have the same mean degree? Explain why or why not.

1.24 We have seen that Facebook's network is incredibly sparse. Assume it has approximately 1 billion users, each with 1000 friends on average.

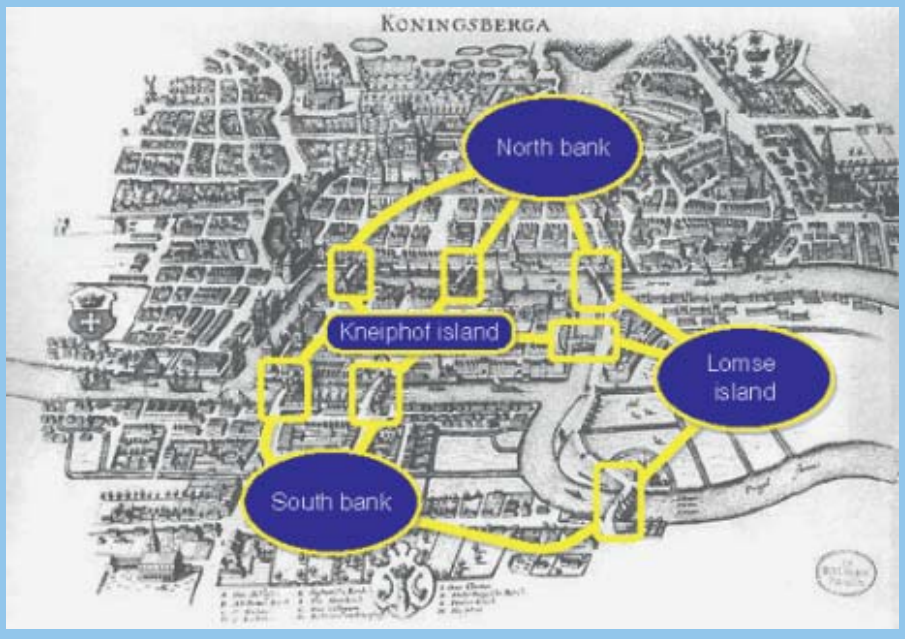
- Suppose Facebook releases its annual report and it shows that while the number of users in the network has stayed the same, the average number of friends per user has increased. Would this imply that the network density increased, decreased, or stayed the same?
- Suppose instead that both the number of users and the average number of friends per user doubled. Would this imply the network density increased, decreased, or stayed the same?

1.25 Netflix keeps data on customer preferences using a big bipartite network connecting users to titles they have watched and/or rated. Netflix's movie library contains approximately 100,000 titles if you count streaming and DVD-by-mail. In the fourth quarter of 2013, Netflix reported having about 33 million users. Assume the average user's degree in this network is 1000. Approximately how many links are in this network? Would you consider this network sparse or dense? Explain.

1.26 Netflix keeps data on customer preferences using a big bipartite network connecting users to titles. Suppose that from 2013 to 2014 Netflix's library has remained the same size, while the number of users has increased. Further suppose that the average user's degree in this network has remained constant. Has the density of this network increased, decreased, or stayed the same?

Box 2.1 The Seven Bridges of Königsberg

In 1736, Leonhard Euler used graph theory to solve a mathematical problem for the first time. The Prussian city of Königsberg was divided by the Pregel river into four land masses (North and South banks, Kneiphof and Lomse islands) connected by seven bridges. The problem was to devise a walk through the city that would cross each bridge once and only once. Euler formulated a generalized version of this problem as finding a path through a network where nodes and links represent the land masses and bridges, respectively, and each link is to be traversed exactly once.



Euler proved that such a path (now called an *Eulerian path* in his honor) exists only if all nodes have even degree, except the source and the target. Nodes must have even degree because for each incoming link to arrive at a node, there must be an outgoing link to depart from the node. The source and the target, if distinct, must have odd degree because when the path starts (ends) it does not "cross" the node. If they coincide (*Eulerian cycle*), there cannot be nodes with odd degree. Since all four nodes in the Königsberg network have odd degree, there was no Eulerian path in this case.

the nodes. Note that there are two paths between nodes **a** and **b**, but the one that passes through node **d** is longer by one link; the shortest path bypasses node **d** by following the link between nodes **e** and **f**. The shortest-path length is $\ell_{ab} = 4$. The directed, unweighted network case is different because directed paths must be consistent with the direction of the links along the path. Therefore there is only one path from source **a** to target **b**, and it goes through **d**. The shortest directed path length is $\ell_{ab} = 5$. Beware that in a directed network there may be no paths between some pairs of nodes. For instance, if a node has only incoming links there are no paths having that node as the source. In the example of

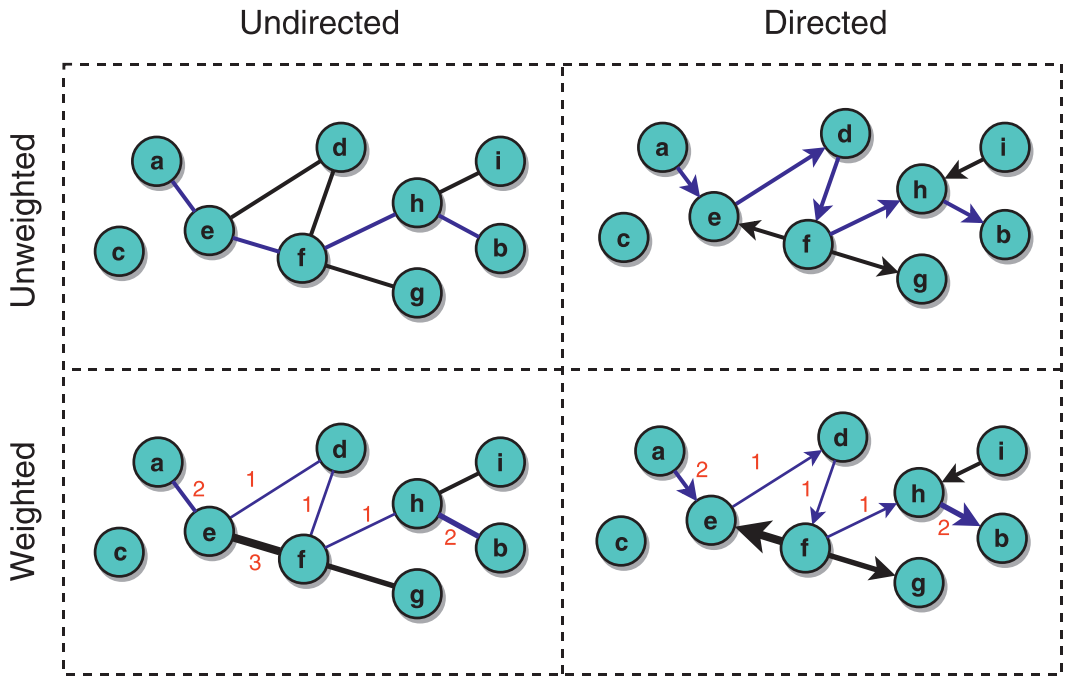


Fig. 2.3 Shortest paths in undirected, directed, unweighted, and weighted networks. Link weights represent distances and are shown in red. In each case the shortest path between nodes **a** and **b**, or from **a** to **b** in directed cases, is highlighted in blue. There is no path between node **c** 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**.

Figure 2.3, there are no paths from **g** to any other node. Similarly, there are no paths to nodes that have only outgoing links, such as **a**.

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.

Formally we define the *average path length* of an undirected, unweighted network as

$$\langle \ell \rangle = \frac{\sum_{i,j} \ell_{ij}}{\binom{N}{2}} = \frac{2 \sum_{i,j} \ell_{ij}}{N(N-1)}, \tag{2.2}$$

where ℓ_{ij} is the shortest-path length between nodes i and j, and N is the number of nodes. The sum is over all pairs of nodes, and we divide by the number of pairs to compute the average. In the directed network case the definition is analogous, but the distance ℓ_{ij} is based on the shortest directed path between i and j, and each pair of nodes is considered twice for paths in both directions:

$$\langle \ell \rangle = \frac{\sum_{i,j} \ell_{ij}}{N(N-1)}. \tag{2.3}$$

The weighted cases are similar, with ℓ_{ij} defined based on link distances. The *diameter* of a network is

$$\ell_{max} = \max_{i,j} \ell_{ij}. \tag{2.4}$$

The definitions of average path length and diameter assume that the shortest-path length is defined for each pair of nodes. If there are any pairs without a path, then the average path length and diameter are not defined. For example, the networks in Figure 2.3 have no path between the singleton node **c** and any other node. We can think of such missing paths as paths with infinite distance. There are a few ways to deal with these cases.

If one wishes to define the average path length in a network where some of the paths do not exist, the following formula can be used for undirected networks:

$$\langle \ell \rangle = \left(\frac{\sum_{i,j} \frac{1}{\ell_{ij}}}{\binom{N}{2}} \right)^{-1}. \tag{2.5}$$

Note that if there is no path between i and j, $\ell_{ij} = \infty$ and therefore $1/\ell_{ij} = 0$ is defined. The same trick can be used for directed networks.

In Section 2.3 we show a couple of different ways to calculate the network distance and diameter when some paths are missing.

Both average path length and diameter can be used to describe the typical distance of a network. In this book we use the former. Although by definition the average cannot exceed the maximum, the two terms are sometimes used interchangeably because the two quantities behave similarly as the network size grows.

NetworkX has functions to determine the existence of paths, find shortest paths, and measure the length of a path or the average path length of a network. In the case of the undirected, unweighted network in Figure 2.3:

Table 2.1 Average path length and clustering coefficient of various network examples. The networks are the same as in Table 1.1, their numbers of nodes and links are listed as well. Link weights are ignored. The average path length is measured only on the giant component; for directed networks we consider directed paths in the giant strongly connected component. To measure the clustering coefficient in directed networks, we ignore link directions

Network	Nodes (N)	Links (L)	Average path length ($\langle \ell \rangle$)	Clustering coefficient (C)
Facebook Northwestern Univ.	10,567	488,337	2.7	0.24
IMDB movies and stars	563,443	921,160	12.1	0
IMDB co-stars	252,999	1,015,187	6.8	0.67
Twitter US politics	18,470	48,365	5.6	0.03
Enron email	87,273	321,918	3.6	0.12
Wikipedia math	15,220	194,103	3.9	0.31
Internet routers	190,914	607,610	7.0	0.16
US air transportation	546	2,781	3.2	0.49
World air transportation	3,179	18,617	4.0	0.49
Yeast protein interactions	1,870	2,277	6.8	0.07
C. elegans brain	297	2,345	4.0	0.29
Everglades ecological food web	69	916	2.2	0.55

2.6 Social Distance

The average path length, defined in Section 2.2, characterizes how close or far we expect nodes to be in a network. Intuitively, in a grid-like network like road networks and power grids, paths can be long. Is this typical of many real-world networks? Let us start by considering a few social networks, in which this question has been explored extensively.

Coauthorship networks are a well-studied kind of social collaboration network because it is relatively easy to gather data about nodes and links. Nodes are scholars, and links can be mined from digital libraries. When we see a publication coauthored by two or more scholars, we can infer links between them in the network.

Paul Erdős was a famous mathematician who made critical contributions to network science, discussed in Chapter 5. (For more background about his life, see Box 5.1.) Mathematicians are fond of studying their distance in the coauthorship network from the particular node corresponding to Erdős. They call this distance their *Erdős number* (Box 2.3). Many mathematicians have a very small Erdős number. Figure 2.8 illustrates the network of collaborations involving Erdős and his over 500 coauthors. In reality, scholars are not just close to Erdős; they are close to everyone. This is typical of collaboration networks: there are short paths among all pairs of nodes. Pick any two scholars and they will not be very far from each other.

Box 2.3 The Erdős Number

Paul Erdős was one of the world's greatest mathematicians. He also stands out among scientists due to his amazing productivity and number of collaborators. Therefore Erdős plays an important role in the connectedness of the scientific collaboration network, in that one can go from many nodes of the graph to many others through him. This is so much so that a special measure has been defined in his honor: the *Erdős number*. Many scientists proudly display their Erdős number on their homepages and CVs. This number is simply defined as the length of the shortest path, in the coauthorship network, from a scholar to Paul Erdős. There is even an online tool to compute the Erdős number for mathematicians (www.ams.org/mathscinet/collaborationDistance.html). For example, Erdős was a collaborator of Fan Chung, who coauthored a report with Alex Vespignani, who has been a coauthor of two of the authors of this book, who therefore have an Erdős number of three. Because of the huge number of Paul Erdős's coauthors, the number of scholars with a small Erdős number is quite large.

It turns out that not only collaboration networks, but pretty much all social networks have very short paths among nodes. You are likely to know someone who knows someone who knows someone...and in a few steps you can get to anyone on the planet! For a demonstration from a more familiar domain, let us turn to the social network connecting movie stars. As we have seen in Chapter 0, nodes are actors and actresses, and two nodes are linked if they have co-starred in a movie. *Six Degrees of Kevin Bacon* is a fun game that originates from such a network. The game, illustrated in Figure 2.9, consists of finding the shortest path connecting an arbitrary actress or actor to Kevin Bacon in the co-star network. For example, a path of length $\ell = 2$ connects Marilyn Monroe to Kevin Bacon. You can play this game online at *The Oracle of Bacon* (oracleofbacon.org). The website pulls data to build the network from the Internet Movie Database (IMDB.com). While Kevin Bacon is often jokingly considered "the" hub of the star network, in reality he is not special; you can enter any pair of actors/actresses and the Oracle shows you the shortest path as a sequence of nodes (stars) and links (movies). Can you find two familiar stars separated by more than four links? Play this game and try!

The Erdős number and the Oracle of Bacon demonstrate that finding long paths in real-world networks is not easy. When we think about it, the concept of short social distance — that we are all only a few steps from each other in the social network — is a familiar one. How many times have you met someone and then been surprised to discover a common friend? The low expectation of running into a friend of a friend is rooted in our intuition of how small our circle of acquaintances is, compared to an entire population. Yet this sort of thing happens often enough, prompting us to exclaim "what a small world!" A *small world* is the popular notion that social distances are short, on average. Consequently, the number of friends of friends out there is much, much larger than we think, and finding short paths in the social network is not so strange after all.

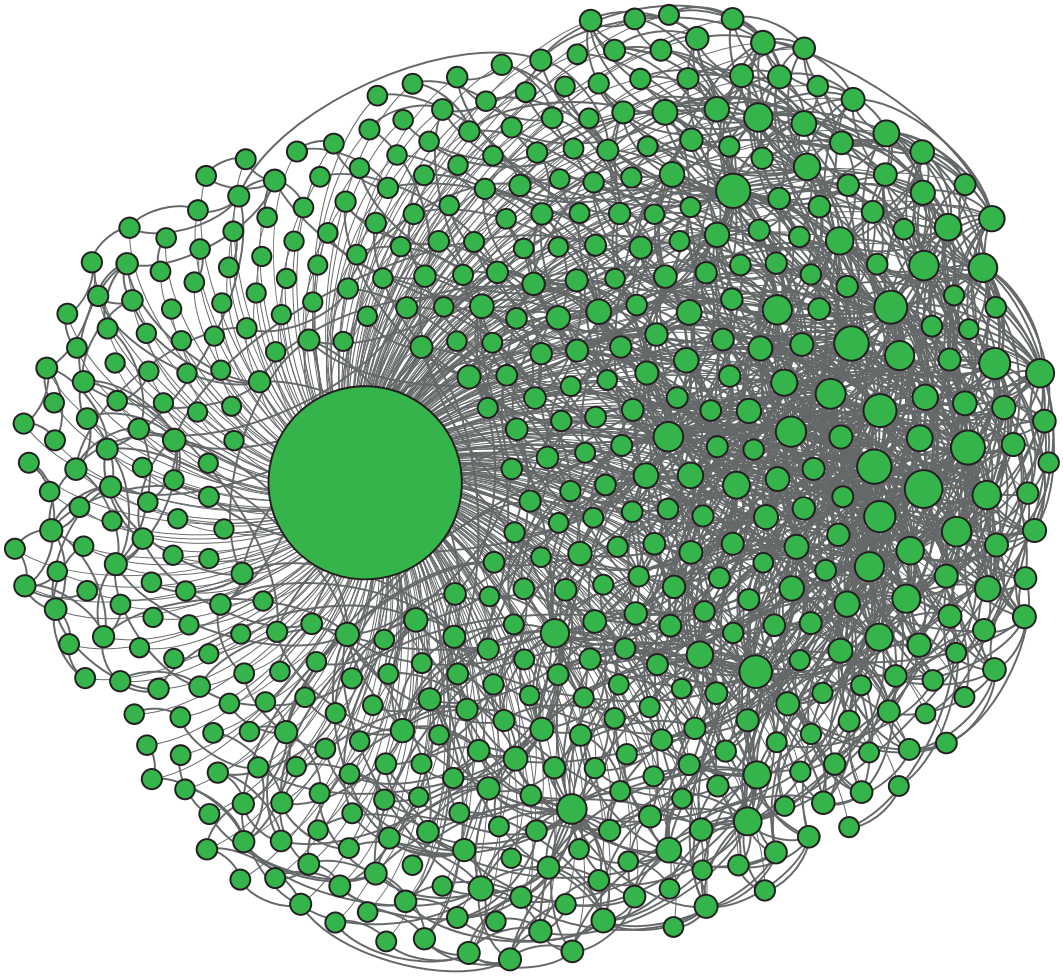


Fig. 2.8 Ego network of Paul Erdős (the large node at the center) within the coauthorship network. Ego networks are defined in Section 1.4.

2.7 Six Degrees of Separation

The name of the game *Six Degrees of Kevin Bacon* is inspired by the concept of *six degrees of separation*. The idea is the same as that of a small world: any two people in the world are connected by a short chain of acquaintances. In other words, social networks have a short diameter and an even shorter average path length. The number "six" in the expression originated from Hungarian author Frigyes Karinthy in the 1920s, and some credit goes to Italian inventor Guglielmo Marconi as well for coming up with the same idea 20 years before, in the early 1900s. However, what made the "six degrees" expression famous was an experiment conducted by psychologist Stanley Milgram[2] in the 1960s, which provided the first empirical evidence of small worlds.

[2] Milgram is famous for another, very controversial experiment, in which subjects were instructed to inflict pain on other people. The goal was to test the degree to which a human being is capable of immoral acts as a result of pressure from authority.

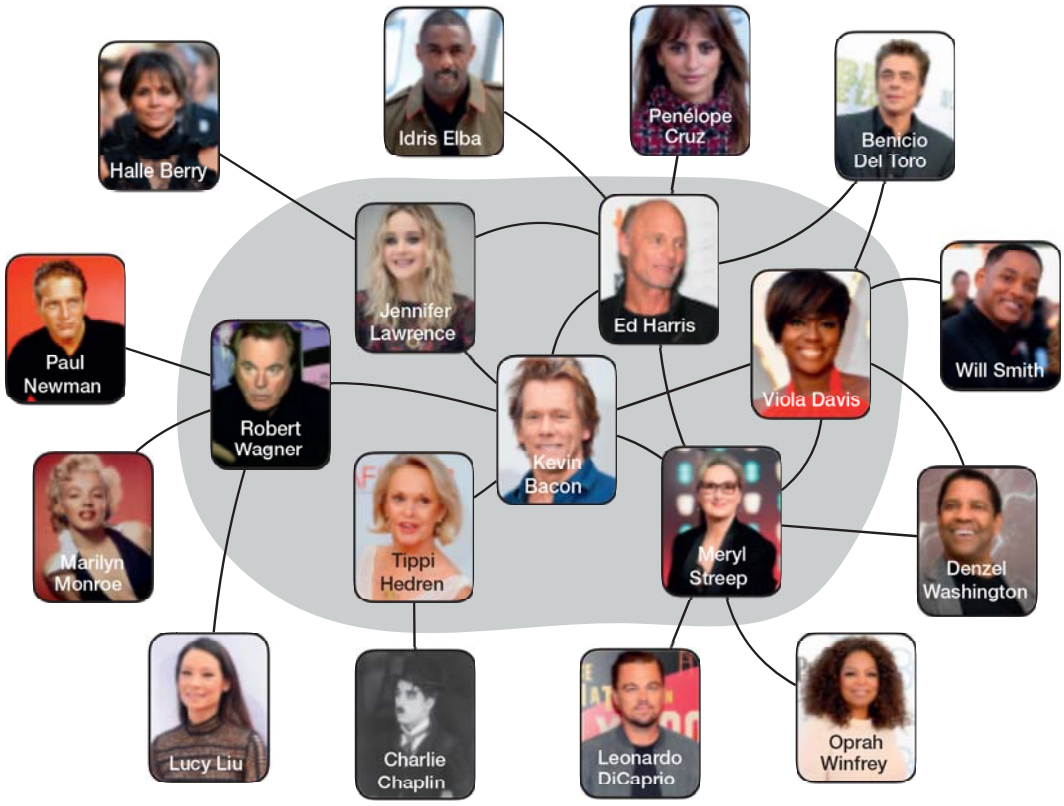


Fig. 2.9 Illustration of the *Six Degrees of Kevin Bacon* game. A few of the nodes connected to Kevin Bacon in the co-star network are shown in the shaded area, along with links among them. A small sample of the nodes at distance $\ell = 2$ is also included. Photo credit: Getty Images.

Milgram wanted to measure the social distance between strangers. He therefore asked 160 subjects in Nebraska and Kansas to forward a letter to an acquaintance, with instructions that the letter should eventually reach a target person in Massachusetts. Each recipient was supposed to forward the letter to someone known, who was likely to know the target. Only 42 of the letters (26%) reached the target. In those cases, however, the path lengths were surprisingly short, ranging between 3 and 12 steps. Figure 2.10 illustrates a typical path of 4 steps. The average path length was a little more than 6 steps, which would eventually inspire a play titled "six degrees of separation" that ultimately popularized the small-world notion. Migram's experiment was replicated in 2003 using emails to recruit a larger number of subjects. There were 18 targets in 13 countries. Of the more than 24,000 chains started, only 384 were completed, with an average length of 4 steps. When accounting for the many broken chains, the authors estimated a median path length of 5–7 steps, in agreement with Milgram's "six degrees." Even more recently, in 2011, researchers at Facebook and the University of Milan examined all 721 million Facebook users who were active at that time (more than 10% of the global population), with 69 billion friendships among them, and found that the average path length was 4.74 steps.

So far we have been referring to the paths we find when playing a game like the *Six Degrees of Kevin Bacon*, or those reported in the studies by Milgram and other researchers,

Fig. 2.10 The path followed by one of the letters in Milgram's experiment. The source subject in Omaha, NE sent the letter to an acquaintance in Santa Fe, NM. From there the letter was forwarded to people in New Orleans, LA and Eugene, OR before reaching the target in Boston, MA.

as "short." But when can we call a path *short*? Compared to what? Would we call a path with 6 steps *short* in a network with only 10 nodes? Clearly we must define what we mean by *short* paths more precisely, and the definition must be relative to the size of the network. In fact, it makes more sense to observe the relationship between the average path length $\langle \ell \rangle$ and the network size N when we consider networks (or subnetworks) of different sizes. We say that the average path length is *short* when it grows very slowly with the size of the network.

We can express slow growth mathematically by saying that the average path length scales *logarithmically* with the size of the network:

$$\langle \ell \rangle \sim \log N.$$

The logarithm of a in base b, $\log_b a$, is the exponent c such that $b^c = a$. Base $b = 10$ is commonly used; $\log_{10} 10 = 1$ because $10^1 = 10$, $\log_{10} 100 = 2$ because $10^2 = 100$, $\log_{10} 1000 = 3$, etc. The logarithm is therefore a function that grows very slowly.

What this means is that the network could have tens of millions of nodes, and yet its average path length would be in the single digits. Furthermore, the network could multiply many times in size while the average path length would only get a few steps longer.

Short paths that obey this kind of relationship are found across social networks, including academic collaborations, actor networks, networks of high-school friends, and online social networks such as Facebook. Short social distances can be useful, say when we are looking for a job. But short paths are not an exclusive feature of social networks. In fact,

searching for paths is something we do routinely in all kinds of networks, for example when we book a long flight and try to minimize the number of intermediate stops. Finding network paths can be fun, too. *Wikiracing* is a hypertext search game designed to work with Wikipedia. A player must navigate from a source article to a target article, both randomly selected, solely by clicking links within each article. The goal is to reach the target in the fewest clicks (i.e. to find a network path with few links). There are variations for teams and with a race against the clock. You can play several versions of this game online, such as *The Wiki Game* (thewikigame.com). You will be amazed at how quickly you can reach any target with a bit of practice. This tells us that Wikipedia has short paths. The same is true for the Web, as we will see in Chapter 4.

As it turns out, short paths are a ubiquitous feature in almost all real-world networks; grid-like networks are among the few exceptions. Table 2.1 reports the average path length of various networks.[3] In all of these examples, the average path length is only a few steps. In the case of the movies and stars network, the paths appear to be longer. However, keep in mind that this is a bipartite network in which a link connects a movie and an actor/actress. If we consider the co-star network, in which two stars are connected if they acted together (as in Figure 2.9), links are associated with movies; in this case the average path length is roughly cut in half. In Chapter 5 we will find short paths even in the simplest of networks, where links are assigned at random.

2.8 Friend of a Friend

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

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

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

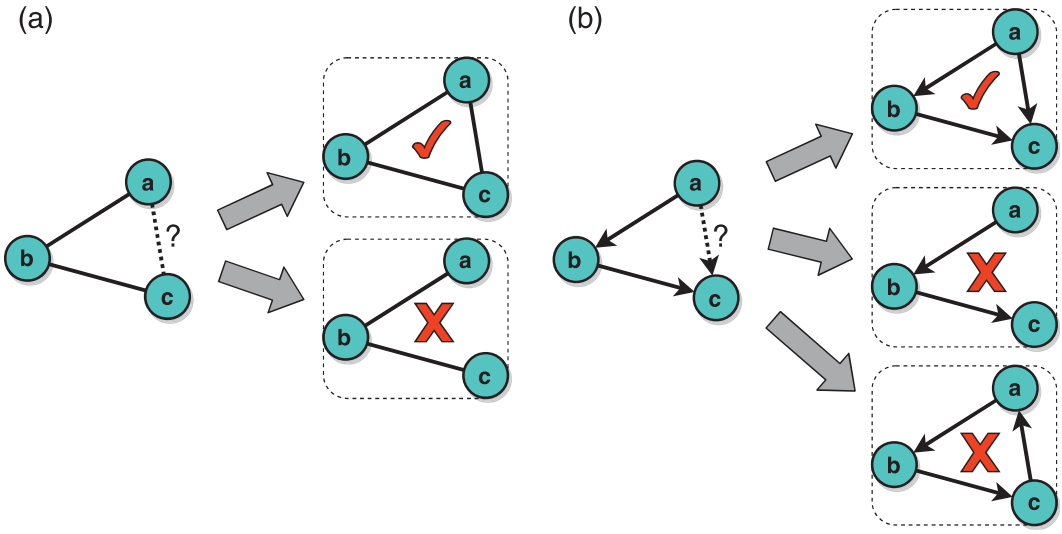


Fig. 2.11 Triads and triangles. (a) In an undirected network, node **b** has neighbors **a** and **c**. They may or may not form a triangle, depending on whether or not **a** and **c** are connected to each other. (b) In a directed network, node **a** links to **b** and node **b** links to **c**. A shortcut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

where $\tau(i)$ is the number of triangles involving i. The maximum possible number of triangles for i is the number of pairs formed by its k_i neighbors. Note that $C(i)$ is only defined if the degree $k_i > 1$ due to the terms k_i and $k_i - 1$ in the denominator: a node must have at least two neighbors for any triangle to be possible.

The clustering coefficient of the entire network is the average of the clustering coefficients of its nodes:

$$C = \frac{\sum_{i:k_i>1} C(i)}{N_{k>1}}. \tag{2.7}$$

Nodes with degree $k < 2$ are excluded when calculating the average clustering coefficient.

Figure 2.12 illustrates how to calculate the clustering coefficient for a few nodes in a network. Node **a** has two neighbors **f** and **g** that are connected to each other, forming a triangle. Therefore its clustering coefficient is $C(a) = 1/1 = 1$. Node **b** has four neighbors. Only two of the six pairs of neighbors are connected: (**e**, **c**) and (**c**, **g**). Therefore, $C(b) = 2/6 = 1/3$. Node **c** has three neighbors that form two triangles via links (**e**, **b**) and (**b**, **g**). The third possible triangle is not realized because the link (**e**, **g**) is missing. Therefore, $C(c) = 2/3$. Finally, node **d** has a single neighbor **e**, therefore $C(d)$ is undefined.

Our definition of a clustering coefficient only applies to undirected networks, because we have only defined undirected triangles. We could extend the definition to directed networks, but it depends on the kinds of triangles that are relevant to a specific case. On Twitter, for example, we might be interested in triangles that shortcut paths along which information

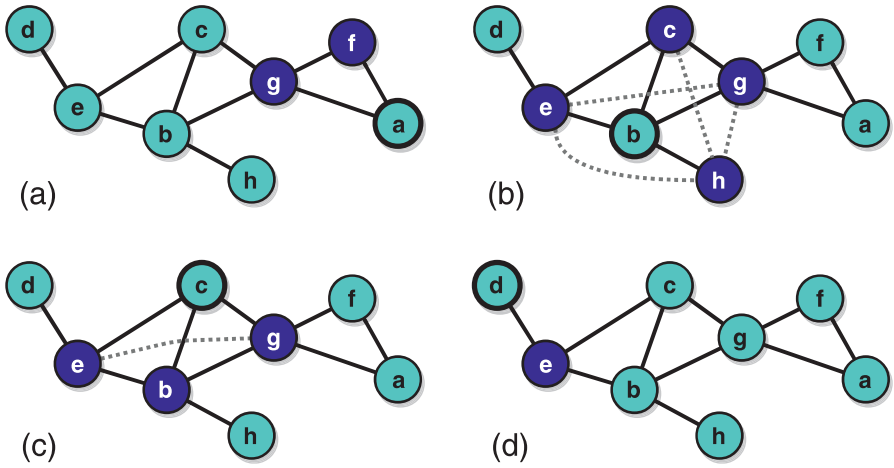


Fig. 2.12 Examples of clustering coefficient. (a) Node **a** has two neighbors **f** and **g** that are connected, forming a triangle. (b) Node **b** has four neighbors **c**, **e**, **g**, and **h**. Two of the six pairs of neighbors are connected, forming two out of six possible triangles. The missing 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.

travels. Consider the scenario in Figure 2.11(b): if **a** follows **b** and **b** follows **c**, then **a** might be interested in following **c** in order to access **c**'s posts directly rather than through **b**'s retweets. In such a scenario, we may want to only count directed triangles that encode these kinds of shortcuts. In this book we only deal with the clustering coefficient in undirected networks; for directed networks we can simply ignore the direction of the links and treat them as if they were undirected when calculating the clustering coefficient.

By averaging the clustering coefficient across the nodes, we can calculate a clustering coefficient for an entire network. A low clustering coefficient (near zero) means that the network has few triangles, while a high clustering coefficient (near one) means that the network has many triangles. Social networks have a large clustering coefficient; a significant portion of all possible triangles are present. For example, coauthorship networks tend to have clustering coefficient above 0.5. A simple mechanism explains the abundance of triangles in social networks: we meet people through shared contacts, thus closing triangles. This mechanism, called *triadic closure*, is discussed further in Chapter 5. Online social networks make suggestions based on triadic closure. For example, Facebook recommends "people you may know" based on common friends, and Twitter recommends accounts followed by your friends (whose accounts you follow). These recommendations result in high clustering.

Table 2.1 reports the clustering coefficient of various networks. We observe high clustering in many, but not all cases. The movies and stars network has $C = 0$. This is because the network is bipartite, therefore there can be no triangles; triangles would require links between pairs of movies or stars, which are not present in the bipartite network. If we instead examine the social network of co-stars, we find a high clustering coefficient. The Twitter retweet network also has a low $C = 0.03$. To understand why, consider that if Bob

retweets Alice and Charlie retweets Bob, Twitter links both Bob and Charlie to the original author, Alice. Therefore each retweet cascade tree looks like a star. The only triangles stem from users participating in multiple stars.

NetworkX has functions to count triangles and calculate the clustering coefficient for nodes and networks. Currently NetworkX sets the clustering coefficient to zero for nodes with degree below two and includes those nodes in the average calculation.

```
nx.triangles(G)           # dict node -> no. triangles
nx.clustering(G, node)    # clustering coefficient of node
nx.clustering(G)          # dict node -> clustering coeff.
nx.average_clustering(G)  # network's clustering coeff.
```

2.9 Summary

In this chapter we have learned about several features of networks: assortativity, connectedness, short paths, and clustering.

1. Assortativity is the correlation between the likelihood that two nodes are connected and their similarity. Similarity can be measured based on degree, content, location, topical interests, or any other node property. Assortativity in social networks can be due to homophily, the tendency of similar people to be connected; or to social influence, the tendency of connected people to be similar.
2. Paths are sequences of links connecting nodes in a network. The natural distance measure between two nodes is defined as the number of links traversed by the shortest connecting path. The simplest way to find a short path is the breadth-first search algorithm. The concepts of paths and distances can be extended to take into consideration link directions and weights.
3. A tree is a connected undirected network with as few links as possible. Trees have no cycles.
4. Connected components are subnetworks such that there exists a path between any two nodes in the same component, but not between two nodes in different components. In directed networks we distinguish between strongly and weakly connected components based on whether or not paths respect link directions.
5. The average path length of a network is found by averaging the shortest-path lengths across all pairs of nodes in a connected network. If a network is not connected, usually only pairs of nodes in the same component are considered.
6. Most real networks have very short paths on average. This is known as the small-world property. The popular notion that social networks have six degrees of separation originated from Milgram's experiment.
7. The local clustering of a network is induced by the presence of triangles, or connected triads. For a node, the clustering coefficient measures the fraction of triangles out of

the maximum possible number. For an entire network we can average the clustering coefficient across nodes. Social networks have high clustering due to friend-of-a-friend triangles.

2.10 Further Reading

The word "homophily" originates from the Greek "homós" (same) and "philia" (friendship). The concept was formulated by Lazarsfeld *et al.* (1954) and the presence of various forms of homophily has been observed in many studies of social networks (McPherson *et al.*, 2001). Aiello *et al.* (2012) found that users with similar interests are more likely to be friends in various online social media platforms, and that similarity among users based on their profile metadata is predictive of social links. The *k*-nearest-neighbors connectivity and assortativity coefficient were introduced by Pastor-Satorras *et al.* (2001) and Newman (2002), respectively.

Researchers are increasingly studying the negative consequences of homophily. Exposure to news and information through the filter of like-minded individuals in online social networks may facilitate the emergence of clustered communities in which our attention is focused toward information that we are already likely to know or agree with. These so-called "echo chambers" (Sunstein, 2001) and "filter bubbles" have been claimed to be pathological consequences of social media recommendation algorithms (Pariser, 2011) and to lead to polarization (Conover *et al.*, 2011b) and viral misinformation (Lazer *et al.*, 2018).

Algorithms for finding shortest paths and connected components in networks have a complicated history. The invention of breadth-first search is attributed to Zuse and Burke in a rejected 1945 Ph.D. thesis, and independently to Moore (1959). There are two famous algorithms for finding shortest paths in weighted networks: one by Dijkstra (1959) and the Bellman–Ford algorithm, published independently by Shimbel (1955), Ford Jr. (1956), Moore (1959), and Bellman (1958).

Milgram's experiment (Travers and Milgram, 1969) was repeated by Dodds *et al.* (2003) using emails. Backstrom *et al.* (2012) found that the average shortest-path length on the network of Facebook friends is lower than five. Newman (2001) first studied the structure of scientific collaboration networks.

An accessible introduction to networks and their small-world and clustered structure is offered by Watts (2004). The existence of triangles in networks is also referred to as *transitivity* (Holland and Leinhardt, 1971). An early definition of the network clustering coefficient was formulated by Luce and Perry (1949), while the local definition used in this book is due to Watts and Strogatz (1998).

The concept of triadic closure was introduced in a seminal paper by Granovetter (1973) and is discussed in Chapter 5. Studying data from a social media platform, Weng *et al.* (2013a) confirmed that triadic closure has a strong effect on link formation, but also found that shortcuts based on traffic are another key factor in explaining new links.

Exercises

2.1 Go through the Chapter 2 Tutorial on the book's GitHub repository.[4]

2.2 Recall that unless otherwise specified, the length of a path is the number of links contained therein. Given two nodes in an arbitrary undirected, connected graph, there must exist some shortest path between them. True or False: There may exist multiple such shortest paths.

2.3 True or False: Given any two nodes in an (undirected) tree, there exists exactly one path between those two nodes.

2.4 Consider an undirected, connected network with N nodes. What is the minimum number of links the network can have? If we do not require the network to be connected, does that minimum number of links change?

2.5 Recall that a tree of N nodes contains $N - 1$ links. True or False: Any connected, undirected network of N nodes and $N - 1$ links must be a tree.

2.6 True or False: Any undirected network of N nodes with at least N links must contain a cycle.

2.7 True or False: Any directed network of N nodes with at least N links must contain a cycle.

2.8 Consider the network defined by the adjacency matrix in Eq. (1.11). Are there any cycles in this network? Is it strongly connected? Weakly connected?

2.9 Consider the unweighted, undirected version of the network defined by the adjacency matrix in Eq. (1.11). Is this network a tree?

2.10 Consider the unweighted, undirected version of the network defined by the adjacency matrix in Eq. (1.11). What is this network's diameter?

2.11 If you convert a weakly connected directed network to an undirected network, will the resulting network be connected? Explain why or why not.

2.12 Consider an arbitrary non-complete undirected network. Now add a single link. How has the number of nodes in this network's giant component changed as a result of this addition?

a. It has strictly decreased
b. It has decreased or stayed the same
c. It has increased or stayed the same
d. It has strictly increased

2.13 Consider the weighted directed network in Figure 2.13. Which of the following most accurately describes the connectedness of this network?

[4] github.com/CambridgeUniversityPress/FirstCourseNetworkScience

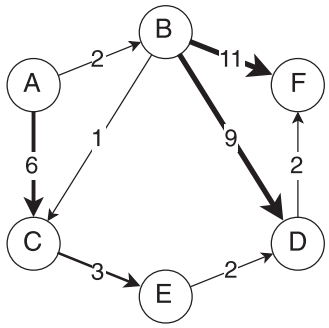


Fig. 2.13 A weighted, directed network. The numbers give the link weights.

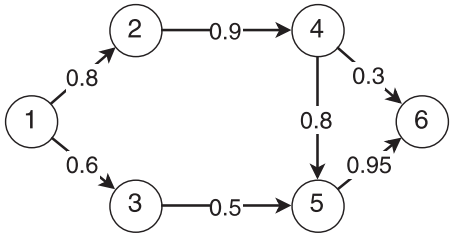


Fig. 2.14 A weighted, directed network. The numbers give the link weights.

a. Strongly connected
b. Weakly connected
c. Disconnected
d. None of the above

2.14 Consider the weighted directed network in Figure 2.13. What is the in-strength of node **D**? What is the out-strength of node **C**? (Recall the definitions from Chapter 1.)

2.15 How many nodes are in the largest strongly connected component of the network in Figure 2.13?

2.16 Consider the network in Figure 2.14. Which of the following most accurately describes the connectedness of this network?

a. Strongly connected
b. Weakly connected
c. Disconnected
d. None of the above

2.17 Link weights can represent anything about the relationship between the nodes: strength of the relationship, geographic distance, voltage flowing through a link cable, etc. When discussing path lengths on a weighted graph, one must first define how the weights are related to the distances. The length of a path between two nodes is then the sum of the distances of the links in that path. The simplest case occurs when the link weights represent the distance. Consider the network in Figure 2.14

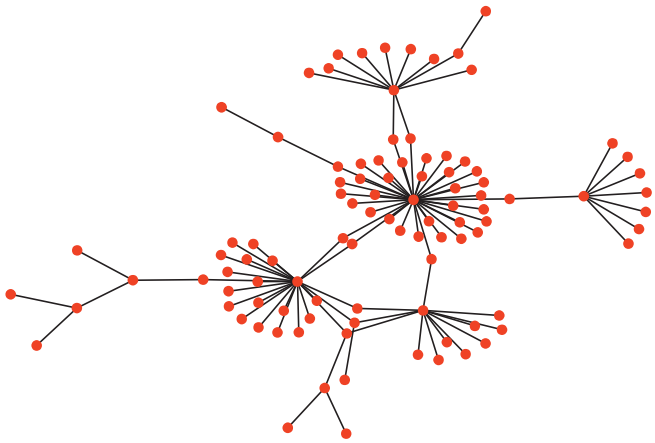


Fig. 2.15 A small subnetwork of the *Drosophila melanogaster* (a.k.a. fruit fly) protein interaction network. Each node represents a protein that interacts with other proteins to perform the essential work of the cell. Experimental evidence has demonstrated that linked proteins form a molecular bond to accomplish some biological function.

and assume that the link weights represent distances. Using this distance metric, what is the shortest path between nodes 1 and 6?

2.18 A common way to define the distance between two nodes is the inverse (or reciprocal) of the link weight. Consider the network in Figure 2.14, and assume that the distance between two adjacent nodes is defined as the reciprocal of the link weight. Using this distance metric, what is the shortest path between nodes 1 and 6?

2.19 Consider the network in Figure 2.15. Which of the following is the best estimate of this network's diameter?

a. 2
b. 4
c. 10
d. 20

2.20 Consider the network in Figure 2.15. Which of the following is the best estimate for the average clustering coefficient of this graph?

a. 0.05
b. 0.5
c. 0.75
d. 0.95

2.21 Would a social network be likely to have the diameter and clustering coefficient of the graph in Figure 2.15?

2.22 Consider the network in Figure 2.16. Which of the following most accurately describes the connectedness of this network?

a. Strongly connected
b. Weakly connected

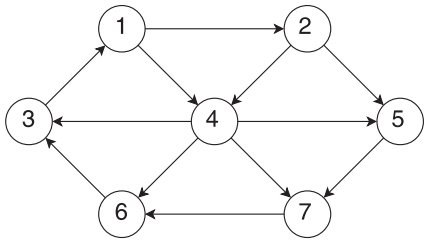


Fig. 2.16 A directed network.

c. Disconnected
d. None of the above

2.23 What is the diameter of the network in Figure 2.16?

2.24 Consider an undirected version of the network in Figure 2.16. What is the diameter of this network?

2.25 Consider any arbitrary directed graph D along with its undirected version G. True or False: If the average shortest-path length and diameter of the directed graph exist, they can be smaller than those of the undirected version.

2.26 Imagine that you were building a competitor of NetworkX. You have already written a method `shortest_path()` to compute the shortest path between two nodes, and now you want to write a function to compute the diameter of a network. Which of the following best describes how to go about doing this?

a. First compute the shortest-path lengths between each pair of nodes. The diameter is the minimum of these values
b. First compute the shortest-path lengths between each pair of nodes. The diameter is the average of these values
c. First compute the shortest-path lengths between each pair of nodes. The diameter is the maximum of these values
d. First compute the average length of all paths between each pair of nodes. The diameter is the minimum of these values

2.27 True or False: A network's diameter is always greater than or equal to its average path length.

2.28 What is the central idea behind the notion of "six degrees of separation"?

a. Social networks have high clustering coefficients
b. Social networks are sparse
c. Social networks have many high-degree nodes
d. Social networks have small average path length

2.29 The American Mathematical Society has a Web tool to find the *collaboration distance* between two mathematicians (see Box 2.3). Use this tool to calculate the Erdős number for a few mathematicians in your institution, or whom you know by fame.

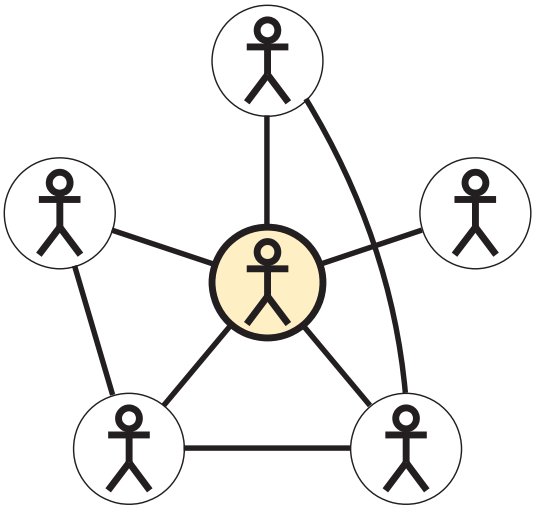


Fig. 2.17 An ego network. The ego is highlighted in yellow.

2.30 Use *The Oracle of Bacon* (oracleofbacon.org) to measure the shortest-path distance in the co-star network among as many pairs of obscure actors and actresses as you can think of. Plot a histogram showing the distribution of the shortest-path lengths, and also estimate the average path length based on your sample. (If you are not familiar with histograms, they are defined in the next chapter.)

2.31 Play *The Wiki Game* (thewikigame.com) until you are able to complete a few rounds successfully. Report the average length (number of clicks) of the discovered paths.

2.32 What is the maximum clustering coefficient for a node in an arbitrary undirected graph?

2.33 What is the maximum clustering coefficient for a node in a tree?

2.34 Recall the definition of an ego network in Section 1.4. Consider the ego network in Figure 2.17: what is the clustering coefficient of the ego?

2.35 Consider the undirected network in Figure 2.4. Compute the shortest-path length for each pair of nodes in the giant component.

2.36 Consider the undirected network in Figure 2.4. Compute the clustering coefficient for each node such that it is defined.

2.37 Consider the network example in Figure 2.12. Compute the shortest-path length for each pair of nodes, and the average shortest-path length for the network.

2.38 Consider the network example in Figure 2.12. Compute the clustering coefficient for each node such that it is defined, and for the network.

2.39 If you use an online social network such as Facebook or LinkedIn, measure your clustering coefficient in the network. (*Hint 1*: If you use a social network with directed links, such as Twitter or Instagram, you can treat the links as undirected.) (*Hint 2*:

This might take a while; it's okay to make an estimate based on a small sample of your friends.)

2.40 Which of the following seemingly conflicting properties are true of social networks?

a. Social networks have short paths, yet large diameter
b. Social networks have small diameter, yet large average path length
c. Social networks have many high-degree nodes, yet are disconnected
d. Social networks are highly clustered, yet are not dense

2.41 The `socfb-Northwestern25` network in the book's GitHub repository is a snapshot of Northwestern University's Facebook network. The nodes are anonymous users and the links are friend relationships. Load this network into a NetworkX graph in order to answer the following questions. Be sure to use the proper graph class for an undirected, unweighted network.

1. How many nodes and links are in this network?
2. Which of the following best describes the connectedness of this network?
   a. Strongly connected
   b. Weakly connected
   c. Connected
   d. Disconnected
3. We want to obtain some idea about the average length of paths in this network, but with large networks like this it is often too computationally expensive to calculate the shortest path between every pair of nodes. If we wanted to compute the shortest path between every pair of nodes in this network, how many shortest-path calculations would be required? In other words, how many pairs of nodes are there in this network? (*Hint*: Remember this network is undirected and we usually ignore self-loops, especially when computing paths.)
4. To save time, let's try a sampling approach. You can obtain a random pair of nodes with

```
random.sample(G.nodes, 2)
```

   Since this sampling is done without replacement, it prevents you from picking the same node twice. Do this 1000 times and for each such pair of nodes, record the length of the shortest path between them. Take the mean of this sample to obtain an estimate for the average path length in this network. Report your estimate to one decimal place.
5. Apply a slight modification to the above procedure to estimate the diameter of the network. Report the approximate diameter.
6. What is the average clustering coefficient for this network? Answer to at least two decimal places.
7. Is this network assortative or disassortative? Answer this question using the two methods shown in the text. Do the answers differ?