7 Dynamics

dyn·am·ics: (*n.*) the forces or properties that stimulate growth, development, or change within a system or process.

Four days before the 2016 U.S. election, a conspiracy site posted a false news report alleging that the staff of one of the presidential candidates was engaged in satanic rituals. The fake news spread in viral fashion on Twitter, mainly among supporters of the opposing candidate who accepted as fact fabricated stories that reinforced their beliefs. Automated accounts known as social bots also contributed to the spread by amplifying its reach. This was just one example of thousands of false news circulating during the electoral campaign — a real epidemic that influenced opinions and, according to some experts, might even have affected the election results.

Scientists are studying the factors that make people and social media platforms vulnerable to this kind of manipulation. Network scientists, in particular, study these phenomena because the structure of online social networks plays a key role in the viral nature of certain messages. For example, Figure 7.1 shows a portion of the diffusion network for the fake report mentioned above. We immediately notice that some nodes, including social bots, were particularly influential.

The spread of misinformation is a special case of information diffusion, one class of dynamic processes that take place on networks. This chapter considers a few other important types of network processes in addition to information diffusion: epidemics, opinion formation, and search. In each case we focus on the *dynamics*, that is, what happens on the network over time — how information and diseases are transmitted across links, how node attributes are affected by their interactions, and how one can search or navigate networks. We present several *models* that capture these dynamics.

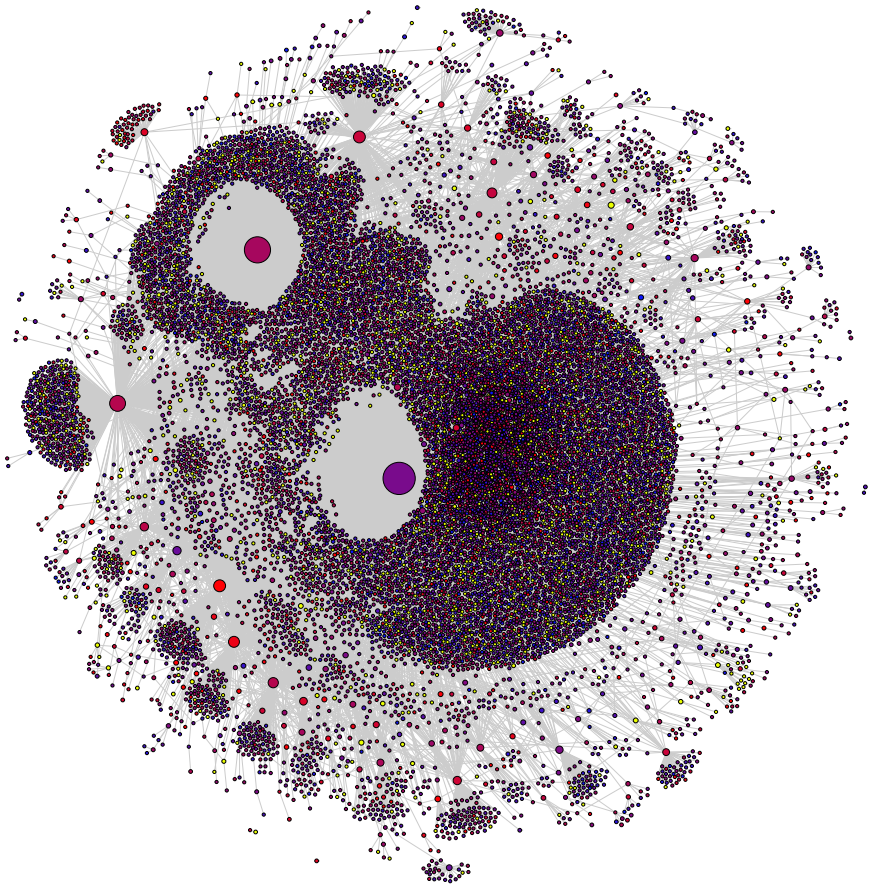


Fig. 7.1 Core of the diffusion network of a viral fabricated news report titled *"Spirit cooking": Clinton campaign chairman practices bizarre occult ritual*, published by the conspiracy site InfoWars four days before the 2016 U.S. election and shared in over 30 thousand tweets. Nodes represent Twitter accounts. A link between two nodes indicates that one of the corresponding accounts has retweeted a post by the other containing the article. Node size indicates account influence, measured by the number of times an account sharing the article was retweeted (out-strength). Node color represents the likelihood that an account is automated, from blue (likely human) to red (likely bot); yellow nodes could not be evaluated because they had been suspended. Recall from Chapter 4 that Twitter does not provide data to reconstruct a retweet tree; all retweets point to the original tweet. The retweet network shown here combines multiple cascades (each a star network originating from a different tweet) that all share the same article. Image courtesy of Shao et al. (2018b); an interactive version of this network is available online (`iunetsci.github.io/HoaxyBots/`).

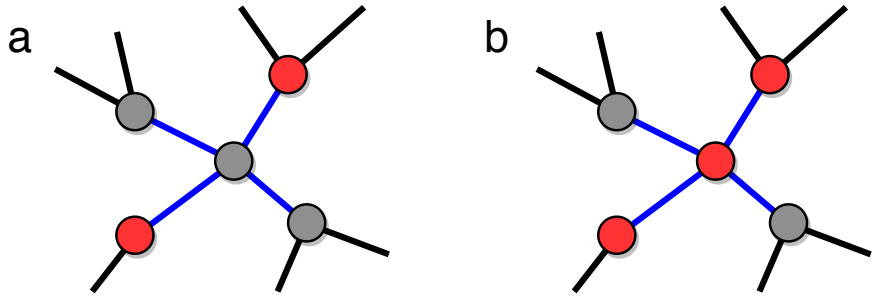


Fig. 7.2 Diffusion of social influence on networks. (a) The central (grey) node is inactive and has two active (red) and two inactive (grey) neighbors. (b) The node becomes active due to the influence of its active neighbors.

7.1 Ideas, Information, Influence

Networks play a central role in the way ideas and information spread in a social community. We are often exposed to new things via friends: for instance, we can find out about a new smartphone model because our best friend just bought it, or discover the latest news on US foreign policy because a friend tells us about it or forwards an article she just read.

Indeed, a lot of what we do is directly or indirectly determined by our social contacts. Social influence is a critical factor when we adopt a behavior, make a decision, embrace an innovation, and shape our cultural, political and religious views. Therefore, modeling how influence, ideas, and information spread in social networks is a key application of network science. These spreading processes are also called *social contagion*, because they resemble a disease that is transmitted via contacts between individuals. In fact, as we shall see in Section 7.2.2, social contagion is often modeled as the spreading of an epidemic.

In any model of influence spreading, we assume that a certain number of nodes (*influencers*) are initially activated, representing that they adopted a new idea, innovation, behavior, etc. Then each inactive node is activated (or not) according to some rule that depends on the presence of active neighbors and on other conditions and parameters, as illustrated in Figure 7.2. The outcome of this process is the generation of *influence cascades*, the activation in sequence of a subset of the nodes in the network. Cascades can range from a handful of nodes to *global cascades* involving a substantial proportion of the network. Sometimes a few nodes end up influencing the whole network. In Section 4.5 we discussed the structure of cascade networks; to see *how* these cascades unfold over time, let us discuss two main classes of social contagion models based on *thresholds* and *independent cascades*.

7.1.1 Threshold Models

The principle of threshold models is very simple: a node can be activated only if the influence exerted on it by its active neighbors exceeds a value. In the most basic version, the *linear threshold model*, the influence on a node is defined as a sum over its active neighbors, in which the contribution of each neighbor is given by the weight of the link joining it to the node: the stronger the connection, the higher the influence of the neighbor. If the influence surpasses a node-specific threshold, the node becomes active, meaning that it adopts the idea, information, or behavior.

In the linear threshold model, the influence on a node i is expressed by:

$$I(i) = \sum_{j:\text{active}} w_{ji}. \tag{7.1}$$

In Eq. 7.1 the sum includes only active neighbors of i; if a node j is not a neighbor, there is no link joining it to i and $w_{ji} = 0$. The condition for the activation of i is

$$I(i) \geq \theta_i, \tag{7.2}$$

where θ_i is the specific threshold of node i, which is assigned to the node before the process starts. Such a threshold indicates the tendency of an individual to be influenced, which usually varies from one individual to another. If the graph is unweighted, Eq. 7.2 reduces to

$$n_i^{on} \geq \theta_i, \tag{7.3}$$

where n_i^{on} is the number of active neighbors of i. In this case, if the number of active neighbors is above the node's threshold, the node is activated, otherwise it remains inactive. If all nodes have equal threshold θ, Eq. 7.3 turns into the simple condition that any inactive node must have at least θ active neighbors to become active.

The model works as follows. First, we choose our network, which could originate from real data or from a graph generation model like the ones introduced in Chapter 5. For simplicity, let us assume that the graph is not weighted. Next we assign a threshold to all nodes, for instance by generating random numbers in some interval. Then a given number of nodes is activated; again, they can be selected at random. Finally, we go through iterative steps in which inactive nodes can become active based on the activation of their neighbors.

Each iteration of the model dynamics consists of the following operations:

1 All active nodes remain active.
2 Each inactive node is activated if the number of active neighbors is at or above its threshold.

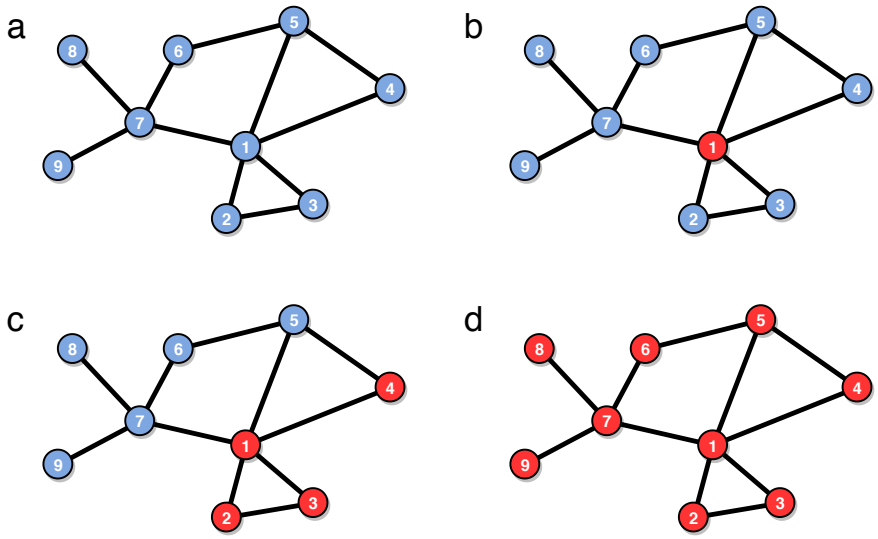


Fig. 7.3 Fractional threshold model of influence diffusion. The activation threshold is $1/2$ for all nodes. (a) Initially, all nodes are inactive. (b) Node **1** is activated. (c) Nodes **2**, **3** and **4** have two neighbors and one of them is **1**, which is active, so they get activated. (d) After the activation of **4**, node **5** has two active neighbors out of three and becomes active (since $2/3 \geq 1/2$). Likewise, nodes **6**, **7**, **8** and **9** are activated subsequently.

The steps are repeated until no further nodes can be activated.

The order in which nodes are considered should not affect the outcome in models of network dynamics. There are two ways to ensure this when implementing the node update rules. In *asynchronous* implementations, nodes are evaluated in a different random sequence at each iteration. This is to avoid biases that may result from always following the same sequence. In *synchronous* implementations, the new activation state of each node in each iteration is determined using the activation values of the other nodes from the previous iteration; all of the nodes are then updated at the end of the iteration. The order is irrelevant in this case.

Many variations of the linear threshold model have been proposed. In the *fractional threshold* model, we consider the fraction rather than the number of active neighbors. So, in this model, in order to activate a node with threshold $1/2$, say, at least half of its neighbors must be active. Figure 7.3 shows how the dynamics of the model unfold on a simple network: the activation of one node triggers a cascade that eventually leads to the activation of all other nodes.

In the fractional threshold model, the activation condition is

$$\frac{n_i^{on}}{k_i} \geq \theta_i, \tag{7.4}$$

where k_i is the degree of node i. The ratio on the left-hand side of Eq. 7.4 is the fraction of active neighbors of i. If all nodes have equal threshold θ, the condition is that an inactive node needs to have at least a fraction θ of active neighbors to be activated.

If the network is sparse, whether or not a global cascade is triggered depends on its structure. The key drivers are the *vulnerable nodes*, *i.e.*, those who can be activated by a single active neighbor.

From Eq. 7.4 we see that a node is vulnerable if $k_i \leq 1/\theta_i$, that is, if its degree is below or at the inverse of its threshold.

To have global cascades, the number of vulnerable nodes has to be sufficiently large. Hubs are usually very effective influencers: the higher the number of neighbors, the more likely it is that some of them have sufficiently low degree to be vulnerable. However, being a hub is not always a sufficient condition for influence. The position of the influencer in the network is also important: a cascade in the periphery of the network will hardly manage to work its way through the core.

Another aspect of the network structure that plays an important role in the size of a cascade is the density and separation between communities. The spread is facilitated within dense communities, but hindered across communities. Cluster boundaries act like walls because a node is unlikely to have multiple active neighbors in different communities.

Knowing the structure of the network enables us to control the size of cascades. In the example of Figure 7.3, if the initial influencer is node **7**, its neighbors **6**, **8** and **9** will become active, but the cascade stops there, because the fractions of active neighbors of nodes **1** and **5** are 1/5 and 1/3, respectively, both below 1/2. However, if we also manage to successively activate node **2**, say, node **3** would also become active and **2**, **3** and **7** would activate **1** allowing the cascade to propagate to the whole network. So, in this case, influencing node **2** "unblocks" the cascade. Indeed, the success of a product or idea often depends on the identification of key individuals that need to be persuaded to buy it. This issue is central in viral marketing, where social networks are used to promote products. Appendix B.6 presents a demonstration of the fractional threshold model.

7.1.2 Independent Cascade Models

Threshold models are based on the concept of *peer pressure*: the more of our contacts share an idea or own a product, the more likely it is that we adopt it ourselves. It is as if our active social neighbors work together to persuade us. But social influence

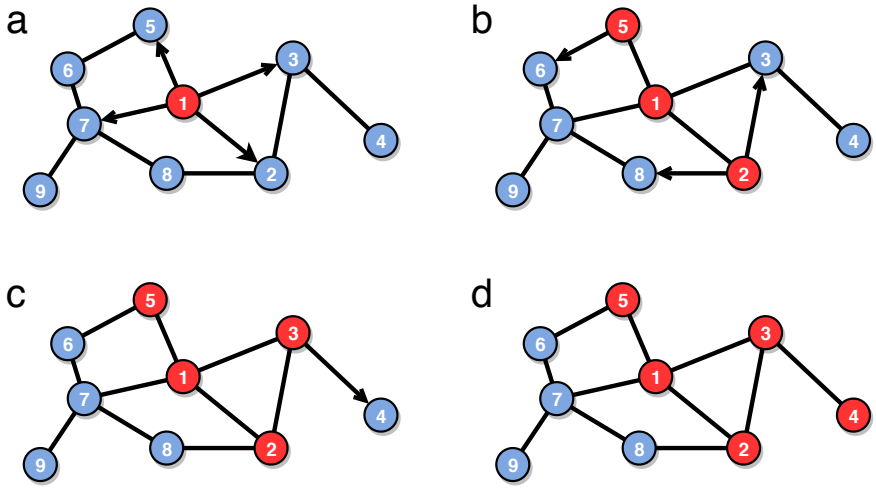


Fig. 7.4 Independent cascade model. The influence probability is set to $1/2$ for all pairs of nodes, so the success of every interaction is decided by flipping a fair coin. The arrows indicate who is trying to influence whom. (a) Node **1** is activated and tries to influence its inactive neighbors **2**, **3**, **5**, and **7**. (b) Nodes **2** and **5** become active and exert their influence on **3**, **6**, and **8**. (c) Node **3** is activated and attempts to convince **4**. (d) Node **4** turns active, the cascade stops.

is often one-to-one: we may be convinced to adopt a product or belief if a single friend speaks enthusiastically about it. Each of our other contacts will have their own influence, unless we have already bought the product or the idea. *Independent cascade models* focus on such node-node interactions.

The setup is the same as for threshold models, in that a network is chosen, or built, and some of the nodes are activated. As soon as a node becomes active, it has one chance to "convince" each of its inactive neighbors; each neighbor is activated with some *influence probability*. If a node fails to activate its friend, it cannot try again. However, the friend can still be persuaded by another active neighbor. The process, illustrated in Figure 7.4, goes on until no further activations occur.

In the simplest version of independent cascade models, an active node i has a probability p_{ij} to convince its inactive neighbor j. Such probability generally depends only on the specific influencer-neighbor pair, so the outcome of each interaction is not affected by what happens to the other pairs. In asynchronous implementations, if j has multiple active neighbors, their activation attempts are sequenced in an arbitrary order to avoid bias. The influence probabilities p_{ij} and p_{ji} may differ, because each node has its own ability to persuade and susceptibility to be persuaded, in general. So it may be easier for i to influence j

than vice versa. The probability p_{ij} can be interpreted as the weight of the link from i to j.

Clearly, the higher the number of active neighbors of an inactive target node, the larger the number of attempts to influence the node and the more likely that it will get activated. Consequently, threshold models and independent cascade models are related, but there are important differences. Threshold models are centered on the target, who is activated if the threshold condition is satisfied. Independent cascade models are centered on the influencer, who persuades its inactive neighbors with given probabilities. In addition, threshold models are usually *deterministic*. The activation of any node depends on whether the threshold condition is satisfied or not; chance plays no role. This means that, if we start from the same initial set of active nodes and activate nodes synchronously, there can only be one outcome. Independent cascade models are instead *probabilistic*: the unfolding of the dynamics depends on chance. In the example of Figure 7.4, different cascades could be triggered by the initial activation of node **1**. In an independent cascade model we can "unblock" a cascade by activating further nodes, suitably chosen, as we have seen in Section 7.1.1 for the linear threshold model. However, due to model's probabilistic character, it is hard to make predictions about the future progress of the cascade, even when the network structure is known.

The very simple models we have described cannot be expected to reproduce real social contagion dynamics. However, more sophisticated variations of these models are capable of capturing important features of many real-world phenomena. One example is a probabilistic version of the threshold model, in which the chances of activation grow with the number of active neighbors. This is similar to the independent cascade model, but contacts with active neighbors are not independent of each other. Such a mechanism models so-called *complex contagion* processes: each new person that exposes us to a product or idea has greater influence than the previous ones in getting us to adopt it or believe it.

7.2 Epidemic Spreading

In the middle of the fourteenth century, humankind suffered one of the greatest calamities in history: the Black Death. Also known as the Great Plague, it is believed to have been caused by the bacterium *Yersinia pestis*, carried by fleas living on black rats that were regularly traveling on board of merchant ships. It probably started in Central Asia and spread throughout all of Europe between from 1346 and 1353 (Figure 7.5). The Black Death is estimated to have killed 30–60% of Europe's population.

While the potentially devastating effects of infectious diseases have been effectively mitigated by great improvements in human living conditions and progress in

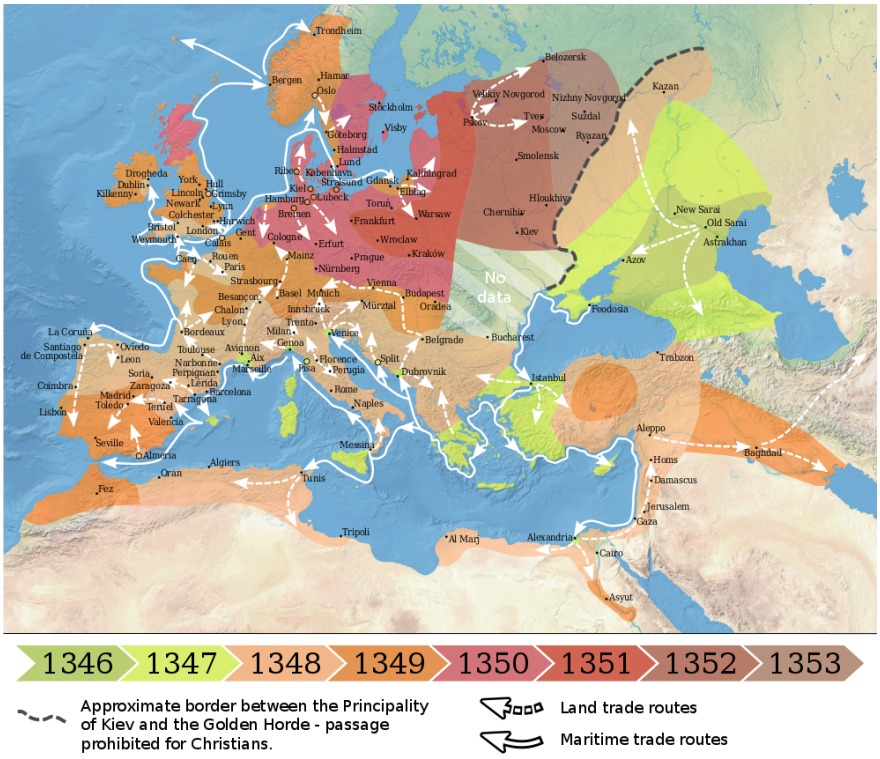


Fig. 7.5 The Black Death reached Europe in 1346 and spread over the whole continent within a few years. The map shows the regions hit by the disease over time, as well as the likely routes of its migration. Image by Flappiefh licensed under CC-BY-SA 4.0 (`commons.wikimedia.org/wiki/File:1346-1353_spread_of_the_Black_Death_in_Europe_map.svg`).

medicine and biology, especially over the past century, the speed of their spread has been strongly enhanced by technological advances in human transportation. In the Middle Ages, the most effective means of travel were horses on land and ships on the sea, and it would take months to reach a remote destination. Nowadays, it takes just a few hours to fly across continents. A person contracting Ebola in Africa could easily travel to Europe, Asia, or America and spread the disease there while still being unaware of it. The world has been facing this kind of emergency repeatedly in recent years.

Technology has also created new forms of epidemics. Computer viruses and other malware spread through the Internet, compromising the function of millions of devices. Mobile phone viruses can easily be transmitted via Bluetooth or Multimedia Messaging Services. Online social media have become fertile ground for the diffusion

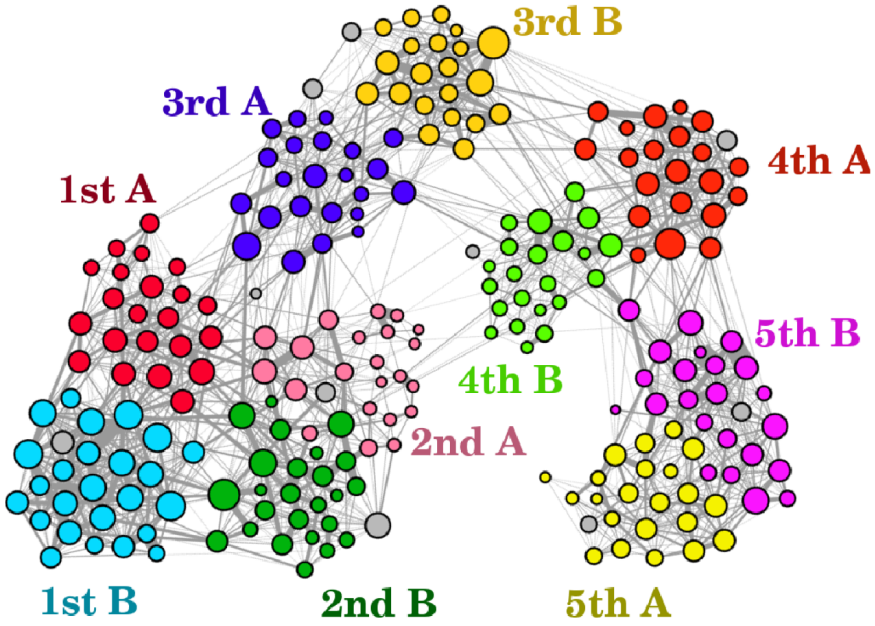


Fig. 7.6 Contact network in a primary school. The links indicate face-to-face proximity among children and teachers in a French school, tracked by Radio-Frequency Identification Devices. The colors label children in the same class and grade; teachers are shown in grey. Nodes with higher degree have larger size, contacts of longer duration are represented by thicker links. While every child eventually interacts with all their classmates after a sufficiently long time, some of them engage with children of other classes as well. This type of network may suggest interventions aimed at containing or mitigating the propagation of infectious diseases in schools. Image reprinted from Stehlé et al. (2011) under CC-BY-4.0 license.

of rumors, hoaxes, fake news, conspiracies, and junk science. Information spreading processes bear many similarities with the epidemics of infectious diseases.

Epidemics spread on *contact networks*, such as networks of physical contacts (Figure 7.6), transportation (Figure 0.7), the Internet (Figure 0.6), email (Figure 0.4), online social networks (Figure 0.1 and 0.3), and mobile phone communication. Many such networks are characterized by the presence of hubs (discussed in Chapter 3), which play a central role in the process. In the remainder of this section we review classic models of epidemic spreading and point out the key differences in the dynamics when they unfold in networks.

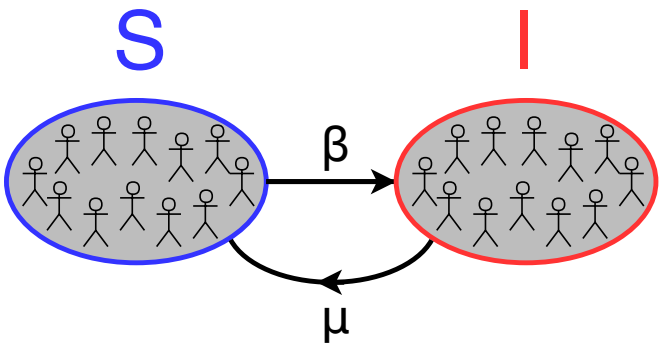


Fig. 7.7 Compartments and transitions in the SIS model. Each susceptible individual gets the disease with probability β after each contact with an infected individual. At each time step, each infected individual has a probability μ to recover from the disease and become susceptible again. Individuals can be infected multiple times.

7.2.1 SIS & SIR Models

Classic epidemic models divide the population into different compartments, corresponding to different stages of the disease. The two key compartments are *susceptible* (S) and *infected* (I). Susceptible individuals can contract the disease, infected individuals have contracted it already and can transmit it to susceptible individuals. Depending on what kind of disease we consider, additional compartments may be needed. In the *susceptible-infected-susceptible model*, or *SIS model*, infected individuals become susceptible again when they recover from the disease, so they can contract it again (Figure 7.7). The model applies to diseases that do not confer long-lasting immunity, like the common cold.

The SIS model starts with either a real-world contact network, reconstructed from empirical data, or from an artificial network generated by some model, like those presented in Chapter 5. Next, we assume that some of the nodes are infected according to some criterion, *e.g.*, at random. All the other nodes are susceptible. During the model dynamics, susceptible individuals contract the disease with a certain probability called *infection rate* at each encounter with an infected individual. Infected people recover from the disease, turning to susceptible, with some probability called *recovery rate* at each time step.

In each iteration of the SIS dynamics, we visit all nodes. For each node i:

1 If i is susceptible, loop over its neighbors: for each infected neighbor, i becomes infected with probability β.
2 If i is infected, i becomes susceptible with probability μ.

As in other spreading models, nodes can be visited asynchronously in random

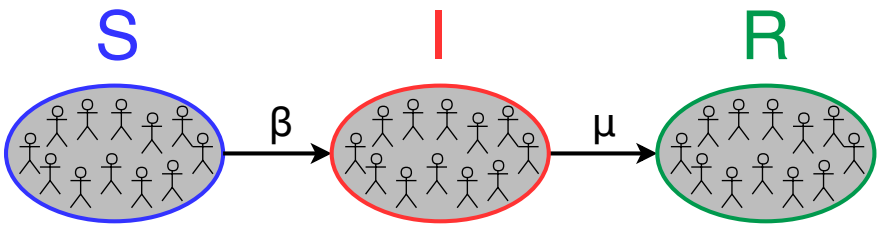


Fig. 7.8 Compartments and transitions in the SIR model. Each susceptible individual gets the disease with probability β after each contact with an infected individual. Each infected individual has probability μ to recover (or die) from the disease at each time step.

order, or synchronously. The *infection rate* β and the *recovery rate* μ are the key parameters of the model.

The dynamics produce a number of transitions from S to I and from I to S that can be sustained indefinitely, under suitable conditions.

Another classic model is the *susceptible-infected-recovered model*, or *SIR model*. When infected individuals recover from the disease, they move to a third compartment of *recovered* (R) people, and they cannot be infected anymore (Figure 7.8). The model applies to diseases that confer long-lasting immunity, like measles, mumps, rubella, and so on. Note that death is a special case of recovered state for deadly diseases, because deceased people do not infect others. The dynamics of infection and recovery follow closely that of the SIS model above, with the same infection rate and recovery rate parameters. The only difference is that, when an infected individual recovers, it is moved to the R state rather than back to the I state; it will not play any further role in the dynamics. Eventually, the SIR model spreading stops, when there are no more infected individuals.

In Figure 7.9 we see the characteristic evolution of both SIS and SIR models, plotted by the fraction of the population that has contracted the disease as a function of time. Initially, just a few people are infected, and the diffusion of the epidemic is irregular and slow. This is followed by a ramp-up phase of exponential growth, that can quickly affect a large number of people. Finally, the process reaches a stationary state, in which the disease is either *endemic*, *i.e.*, it affects a stable fraction of the population over time, or eradicated.

Classic epidemiology models can be simplified by making the *homogeneous mixing approximation*, which consists in assuming that each individual can be in contact with any other. This way, all individuals in the same compartment have identical behavior and only the relative proportions of people in the various compartments matter for the model dynamics. This is equivalent to assuming that the individuals are nodes of a complete graph, where everybody is linked to everyone else. Such a

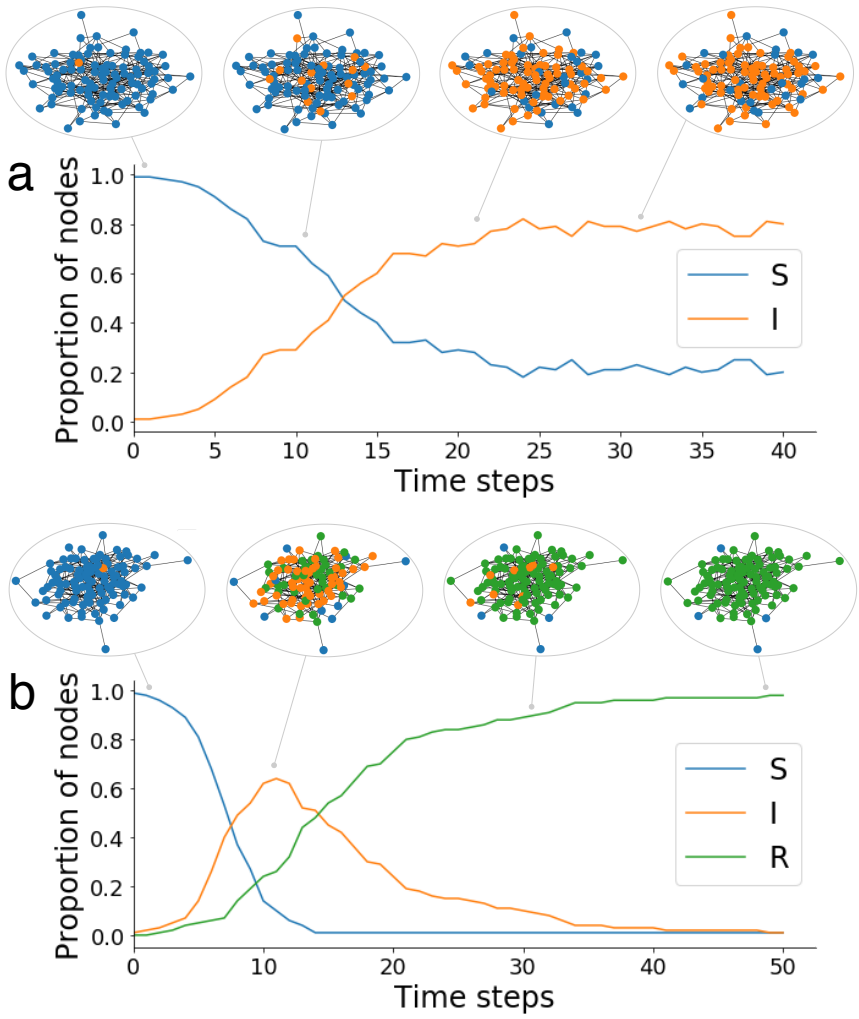


Fig. 7.9 Schematic evolution of (a) SIS and (b) SIR model dynamics. The fraction of infected individuals is plotted versus time, following an epidemic outbreak. After an initial phase, characterized by a low proportion of infected people, the epidemic grows fast until a fraction of the population is hit by the disease. The final phase depends on the model: for the SIS model, the infected stabilize around a constant fraction (which can also be very small or even zero), signaling an endemic state. For the SIR model, the infected fraction always goes down to zero as individuals recover.

simplifying assumption could be justified for a small population, like the inhabitants of a little village where all people are in touch with each other. But in real large-scale epidemics, individuals can only be infected by the people they come in contact with. It is therefore critical to reconstruct the actual network of contacts to the extent possible.

At each iteration of the model, there are newly infected individuals, called *secondary infections*, along with sick individuals who recover from the disease. For the epidemic to spread, there must be more secondary infections than recovered people, because only this way the number of infected people can grow. On homogeneous networks, where all nodes have similar degree, meaning that every individual comes in contact with roughly the same number of people, this condition leads to a *threshold effect*. We can define the *basic reproduction number* as the average number of new infected people generated by an infected individual over the course of its infectious period. This quantity depends on the infection rate, the recovery rate, and the average degree. If it is larger than a threshold, then the epidemic can hit a significant fraction of the population; otherwise it will be absorbed quickly, without major effects.

Assume a homogeneous contact network, with all nodes having degree approximately equal to the average $\langle k \rangle$. According to the SIS and the SIR model dynamics, each sick person infects a susceptible neighbor with probability β. In the early stages of the epidemic only few people are infected, so we can assume that each of them is in contact with mostly susceptible individuals. Each infected person can transmit the disease to about $\langle k \rangle$ individuals at each iteration. Therefore, the average number of infections caused by a single person after one iteration in the early stage of the spreading process is $\beta\langle k \rangle$. On the other hand, at each iteration every sick individual recovers with probability μ. So, if there are I infected individuals, after one iteration there will be on average $I_{sec} = \beta\langle k \rangle I$ secondary infections, while $I_{rec} = \mu I$ people are expected to recover. For the epidemic to spread we must have $I_{sec} > I_{rec}$, which leads to the *epidemic threshold* condition:

$$\beta\langle k \rangle I > \mu I \implies R_0 = \frac{\beta}{\mu}\langle k \rangle > 1. \tag{7.5}$$

The variable $R_0 = \beta\langle k \rangle / \mu$ is the *basic reproduction number*. Equation 7.5 states that if $R_0 < 1$, the initial outbreak dies out in a short time, affecting only a few individuals. If $R_0 > 1$, the epidemic can keep spreading.

For the epidemic to affect a significant portion of the population, each infected person must transmit the disease to more than one other individual. This condition is necessary but not sufficient: in certain situations, the epidemic may not have major consequences even if the basic reproduction number is above one; factors such as quarantine policies or the network community structure might prevent the epidemic from spreading. In general, the higher the basic reproduction number, the

more infectious the disease. For example, the number is above ten for measles and around two for Ebola.

Appendix B.5 presents a demonstration of both SIS and SIR models on homogeneous networks. But as we have seen, real contact networks are not homogeneous. The presence of hubs changes the scenario significantly. If there are nodes with very large degree, *there is effectively no threshold*: even diseases with low infection rate and/or high recovery rate may end up affecting a sizable fraction of the population! In fact, even if the probability to contract the disease is low, it is fairly easy to infect one or more hubs, who are very exposed due to their high number of contacts. Once infected, the hubs are dangerous spreaders among their many susceptible contacts, who will propagate the infection further to their contacts, and so on.

Because of the role of hubs, when facing real epidemic emergencies, effective containment strategies should aim to vaccinate or isolate people with many contacts. For example, sex workers are primary targets of vaccination campaigns for sexually transmitted infections. In many cases, it is not obvious how to identify contact network hubs. Section 3.3 suggests a way. By following the links of a network we increase the chance to bump into hubs. So, instead of vaccinating a random sample of the population, one should vaccinate their friends!

7.2.2 Rumor Spreading

Social contagion can be naturally described as the spreading of an epidemic. In fact, the models of social contagion we have reviewed share similarities with the SIS and the SIR models, especially the independent cascade models of Section 7.1.2.

A variant of the SIR model can be used to describe the spreading of rumors in a community. Like the SIR model, this *rumor spreading model* has three compartments: ignorant (S), spreaders (I) and stiflers (R). The latter are people who know about the rumor but do not contribute to spreading it. The basic idea is that people are engaged in the diffusion of the rumor as long as they find people who are unaware of it, otherwise they lose interest and stop spreading the rumor.

The rumor spreading model starts with a network of contacts, which could be a real network or an artificial one generated by some model, like those we have seen in Chapter 5. All the nodes are ignorant except for some that are turned into spreaders of the rumor, according to some criterion; they can be selected at random. In the dynamics of the model, when a spreader approaches an ignorant, the rumor is told and the ignorant becomes a spreader with a *transmission probability*. When a spreader meets a stifler, the spreader becomes a stifler with a *stop probability*. When two spreaders meet, they both turn to stiflers with the same stop probability. Figure 7.10 illustrates these transitions. Nothing happens if an ignorant meets a stifler.

In each iteration of the rumor spreading model dynamics, all nodes are visited synchronously or asynchronously in random order. For each node i:

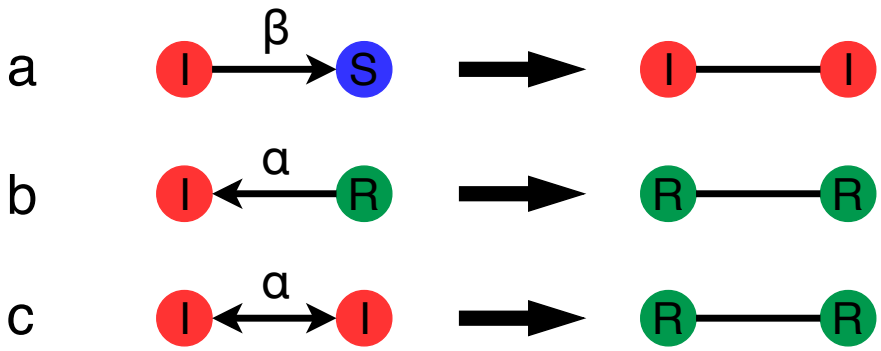


Fig. 7.10 Rumor spreading model. (a) The rumor circulates only if a spreader (I) meets an ignorant (S). In this case, the ignorant becomes a spreader with probability β. (b) If a spreader meets a stifler (R), the spreader becomes a stifler with probability α. (c) If two spreaders meet they both become stiflers with probability α.

1 If i is ignorant, loop over its neighbors: for each spreader neighbor, i becomes a spreader with probability β.
2 If i is a spreader, loop over its neighbors:

(i) For each stifler neighbor, i becomes a stifler with probability α.
(ii) For each spreader neighbor, i and the neighbor both become stiflers with probability α.

The transmission probability β and stop probability α are the two key parameters of the model.

An important difference with the SIR model is that here the transition from I to R does not occur spontaneously (in that a sick person recovers from the disease), but depends on the interaction between individuals. As in the SIR model, starting from a few spreaders, eventually all individuals will be either ignorant or stiflers, as in this case the dynamics cannot produce any change. The number of stiflers in the final state is also the number of people who found out about the rumor.

The rumor spreading model does not have a threshold effect, even on homogeneous networks. The rumor can reach a large number of people even if the transmission probability is low. On heterogeneous networks, there is still no threshold, and the final number of people aware of the rumor is lower than on homogeneous networks with equal numbers of nodes and links. This occurs because the rumor reaches the hubs in the early stages of the process and they quickly become stiflers due to their multiple interactions with other individuals, some of whom may be aware of the rumor. Once the hubs turn into stiflers, the diffusion process slows down.

7.3 Opinion Dynamics

We have opinions about everybody and everything. Opinions drive our behavior, affect our choices, influence our plans. Policies implemented by governments worldwide are dictated by opinions about trade, conflicts, immigration, pandemics, the environment, and so on. Opinion dynamics are the processes that determine how opinions form and diffuse in society. With the introduction of the Internet and social media, humankind has endowed itself with incredibly powerful tools to circulate and even manipulate opinions. Opinions spread on networks like those of Facebook friends and Twitter followers. Therefore, network models can help us understand how opinions propagate.

Models of opinion dynamics are similar to models of influence spreading seen in the previous section, but they have some distinctive features. We can represent an opinion as a number or a set of numbers. Models are usually divided in two categories, based on whether they use *discrete* (integer) or *continuous* (real-valued) opinions. Next we introduce simple models in both classes. We will also discuss the interplay between network structure and dynamics, as in several realistic scenarios the structure of the network affects the processes that take place on it, but the dynamics may in turn change the structure as well.

7.3.1 Discrete Opinions

People are sometimes confronted with a limited number of positions on a specific issue, often just two positions: right/left, Android/iPhone, buy/sell, and so on. In such cases, the opinion is represented by an integer attribute or *state* of each node. For simplicity let us consider just the case of binary opinions.

A model is characterized by the set of rules that determine how the opinion of a node changes due to the opinions of its neighbors. The dynamics usually follows these steps:

1 In the initial configuration, opinions are randomly assigned among the nodes of the network. This means that initially there is about the same number of people holding either opinion (*disagreement*).
2 The opinion update rule is applied over and over to all nodes. An iteration consists in running a loop over all nodes. Typically, nodes are update asynchronously in random order to facilitate convergence.
3 There are two possible outcomes:

(i) The system reaches a steady state, where no node changes its opinion anymore. Such state can be a *consensus*, with all nodes having the same opinion, or *polarization*, with some nodes holding an opinion and the rest holding the other.
(ii) The system does not reach a stationary state, in that some nodes (or all) keep changing their opinions at each iteration. Still, some features of the opinion

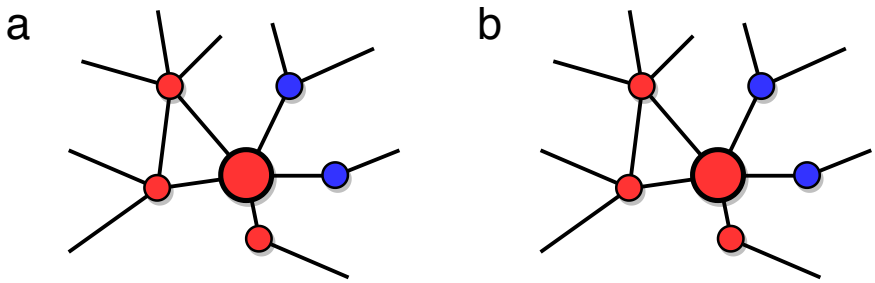


Fig. 7.11 Majority model of opinion dynamics. (a) The node to be updated (big circle) has opinion one (red). The node has five neighbors: three have opinion one, the other two have opinion zero (blue). (b) The node takes the opinion of the majority, so it stays red.

configuration, for example the averages of some variables, may stabilize in the long run.

A few standard variables can be computed and monitored in these models:

- The *average opinion* is the arithmetic average of the opinions across the nodes. If we start from a random distribution of two opinions, zero and one, the average is around 0.5, as half of the nodes will have opinion zero and the other half opinion one. The average opinion usually changes during the dynamics and one can keep track of its value after each iteration. If the system reaches a stationary state, the average converges to a precise value. If the stationary state is consensus, it equals either zero or one, depending on which opinion dominates.
- The *exit probability* estimates how likely the network is to reach consensus to opinion one, as a function of the fraction of nodes with opinion one in the initial configuration. As an illustration, suppose that we run the model dynamics 100 times, starting from 100 different random configurations. In each initial configuration we assign opinion one to every node with probability 0.4, so that approximately 40% of the nodes will have opinion one. Imagine that all runs lead to consensus, 30 of them to consensus opinion one. The value of the exit probability for initial probability 0.4 of opinion one is then $30/100 = 0.3$.

Two simple discrete opinion dynamics models are borrowed from statistical physics: the *majority model* and the *voter model*. In the former, the dynamics are based on a majority rule: each node takes the opinion of the majority of its neighbors, as shown in Figure 7.11. If the number of neighbors is even and there is an equal number of them with either opinion, then we flip a coin to decide which opinion will be taken by the node. This is basically equivalent to the fractional threshold model presented in Section 7.1.1, with a threshold of 1/2. The difference is one of interpretation: here we think of two opinions in competition rather than one idea spreading.

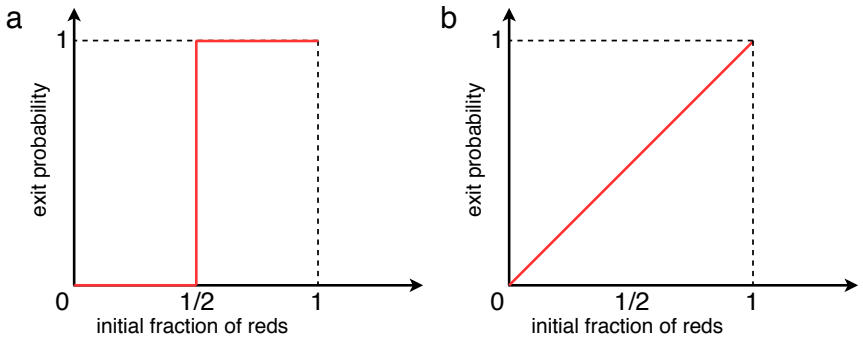


Fig. 7.12 Exit probability. (a) Majority model on a grid network. The step function indicates that the initial proportion of either opinion will determine whether or not the system will reach consensus on that opinion: if the dynamics lead to consensus and more than half of the nodes have opinion one (zero) in the starting configuration, then the consensus is on opinion one (zero). This diagram can be drawn only for one- or two-dimensional grids, as the dynamics never lead to consensus otherwise. (b) Voter model. The diagonal function indicates that the initial proportion of opinions one is also the probability to reach consensus on opinion one. In contrast with majority dynamics, in the voter model it is possible to reach consensus on any opinion even if less than half of the nodes have that opinion initially.

Consensus is the stable state in which all nodes have the same opinion and nothing can change. But there are other stable states: if a node has the opinion of the majority of its neighbors, as in Figure 7.11, its opinion will not change. Such a local majority condition is often reached by all nodes in the network, giving rise to stable configurations in which both opinions coexist. On most networks we have seen in this book, like all model networks of Chapter 5, majority dynamics never reaches consensus; the network gets trapped in states with opinion coexistence. Consensus is reached only on one- and two-dimensional grids. In fact, on the two-dimensional square grid, consensus is reached in about 2/3 of the runs. If we compute the exit probability for the runs that lead to consensus, we obtain the characteristic step-like profile shown in Figure 7.12(a): in order to reach consensus on any opinion, that opinion must have the majority in the initial configuration.

In the voter model, illustrated in Figure 7.13, each node takes the opinion of a randomly chosen neighbor, whatever it may be. A demonstration of both the majority and voter models is presented in Appendix B.6. Consensus is the only stable state of the voter model dynamics, so it is the inevitable final configuration of the system, on any connected network. In fact, as long as different opinions coexist, neighbors with different opinions can always influence each other. The exit probability of the voter model coincides with the fraction of initial nodes with opinion one, so it is the diagonal function in Figure 7.12(b). In contrast to the majority model, here the outcome of the dynamics is uncertain. For instance, suppose that

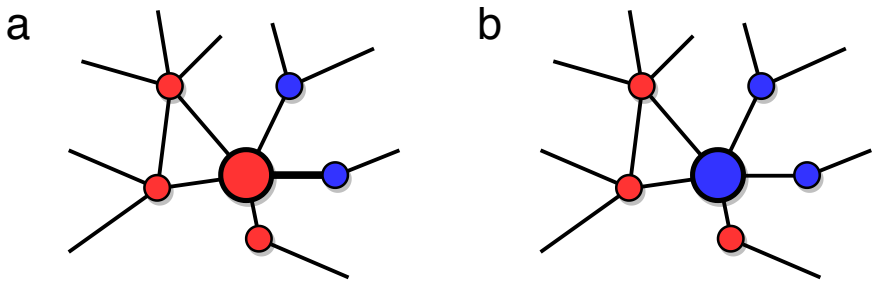


Fig. 7.13 Voter model. The neighborhood of the node to be updated (big circle) is the same as in Figure 7.11. (a) A random neighbor is chosen (the blue node attached to the thick link). (b) The central node takes the opinion of its neighbor.

30% of the nodes have opinion one in the initial configuration. Then we expect that in 30% of the runs all nodes will end up having opinion one, but we cannot tell in advance whether a specific run will lead to consensus on opinion one or zero.

Many variations of the voter model exist. Common modifications are:

- The presence of *zealots*, nodes who never change their opinion. If they all have the same opinion, they will favor consensus around that opinion, otherwise consensus is never reached.
- Considering more than two opinion states. In this case, the interactions may be constrained to occur only among nodes with sufficiently close opinions. For example one could have three opinions (1, 2 and 3) such that only neighboring opinions can interact (1 and 2, 2 and 3, but not 1 and 3). We discuss such a principle in detail in Section 7.3.2. Non-consensus configurations with non-interacting opinions are stable in any network.
- The possibility for nodes to change their opinion spontaneously, for example with a certain probability at each iteration, on top of the voter dynamics.

Similar modifications can also be applied to other discrete opinion dynamics models.

7.3.2 Continuous Opinions

There are situations in which the opinion of an individual can vary smoothly from one extreme to the other of a range of possible choices. For example, it may express the appreciation for an artwork, which could continuously vary from dislike (0) to enthusiasm (10). Or we might wish to model political alignment on a spectrum from very progressive (–1) to very conservative (+1). In such cases, opinions are better represented by real, continuous numbers.

As in discrete opinion models, random opinions are usually assigned to network nodes in the initial configuration. This can be accomplished by generating random numbers in the desired range. Then the opinion values change as they are updated over and over. If at some point the largest variation of any opinion is smaller than a

predefined threshold, we can stop the simulation because the system will eventually reach a stationary state. Typical stationary states are *consensus*, *polarization*, or *fragmentation* depending on whether opinions are concentrated around one, two, or more values, respectively. In the limit of infinite simulation time, each node will have exactly one of the few surviving opinions.

We imagine people having a constructive debate about a topic, with the chance of affecting each other's opinion especially when their positions are sufficiently close to each other. An individual can hardly convince another if the latter has an opposite point of view. This simple observation has inspired the *principle of bounded confidence*: two opinions can affect each other only if their difference is smaller than a given amount, which is called *confidence bound*, or *tolerance*.

The original *bounded-confidence model* has an update rule that consists in choosing a node and one of its neighbors. If their opinions differ by less than the confidence bound, they both "move" towards each other, by some relative amount determined by a convergence parameter. Otherwise, the opinions do not change.

In the bounded-confidence model, at iteration t, each node i has opinion $o_i(t)$, which is a real number between, say, zero and one. An iteration consists of a sweep over all nodes, synchronously or in random order. At iteration $t+1$, when it comes to node i, we pick one of its neighbors at random, say j. If

$$|o_i(t) - o_j(t)| < \epsilon, \tag{7.6}$$

where ϵ is the confidence bound, the values of the opinions are updated to

$$o_i(t+1) = o_i(t) + \mu[o_j(t) - o_i(t)] \tag{7.7}$$

$$o_j(t+1) = o_j(t) + \mu[o_i(t) - o_j(t)], \tag{7.8}$$

where $\mu > 0$ is the *convergence parameter*. If $\mu = 1/2$, the opinions converge to their average, meaning that both individuals adopt a common intermediate position. If $\mu = 1$ the opinions switch, in that i adopts j's opinion and vice versa. Usually μ varies in the range $(0, 1/2]$.
If we sum Equations 7.7 and 7.8 side by side and divide by two, we see that the second terms on the right-hand sides cancel each other out. We conclude that the average opinion of i and j is the same before and after the update: *the average opinion of the system is preserved* in the bounded-confidence dynamics! If the initial opinions are taken at random from the range $[0, 1]$, their average is $1/2$ (with possible small deviations). So, if the system eventually reaches consensus, the opinions of all nodes will cluster around $1/2$.

Starting from a random initial opinion configuration, the dynamics always lead to a stationary state, on any network. The convergence parameter only affects the number of iterations needed to reach convergence. The number of clusters of opinions in the stationary state depends on the confidence bound and on the structure

of the network. The lower the confidence bound, the larger the number of final opinion clusters.

For $\epsilon > 1/2$, the system always reaches consensus, on any network, with the opinions centered around 1/2.

There are many variations of the bounded-confidence model. Common modifications include:

- Using individual values of the confidence bound, to account for the fact that not everybody can be convinced as easily as everybody else. In some extensions, the confidence bound of a node is coupled with the individual's opinion. For instance, if the opinion is close to the extremes of the range, the confidence bound is small because extremists are more difficult to persuade than most people.
- The possibility for individuals to change their opinion spontaneously. As in the voter and other models, this can be implemented by letting nodes change their opinion with some probability at each iteration.

7.3.3 Coevolution of Networks and Dynamics

In Section 2.1 we have seen that assortativity is found in many real graphs, particularly social networks: the nodes are similar to their neighbors. We have also discussed the two possible mechanisms responsible for this: *social influence* (neighbors becoming more similar) and *selection* or *homophily* (similar nodes becoming neighbors). It is plausible that both mechanisms are responsible for the observed assortativity. For instance, if we are constantly debating about an issue with one of our acquaintances, we might either try to find a compromise, or else be better off if we hang around with someone else who shares our view. This happens a lot on social media, where people "unfriend" or "unfollow" contacts with different views. In the models of opinion dynamics discussed so far, the network is fixed. So we are not allowing for selection, because nodes with very similar opinions do not have the option to become neighbors, unless they already are. Similarly, neighbors with very dissimilar opinions cannot become disconnected. Nodes can only influence each other's opinions. A realistic model should allow for the interplay of both influence and selection. This has led to the development of *coevolution models*, in which opinion changes may induce modifications in the network structure, which could in turn affect the opinions, and so on. Basically, opinions and networks *adapt* to each other.

In one coevolution model, opinions are discrete and can take two or more values. At the beginning, opinions are randomly assigned to the nodes. The dynamics consist of alternating selection and influence steps, with a relative frequency determined by a parameter. By selection, nodes set links to other nodes with the same opinion. By influence, nodes take the opinion of their neighbors. Figure 7.14 illustrates the selection and influence steps of the model.

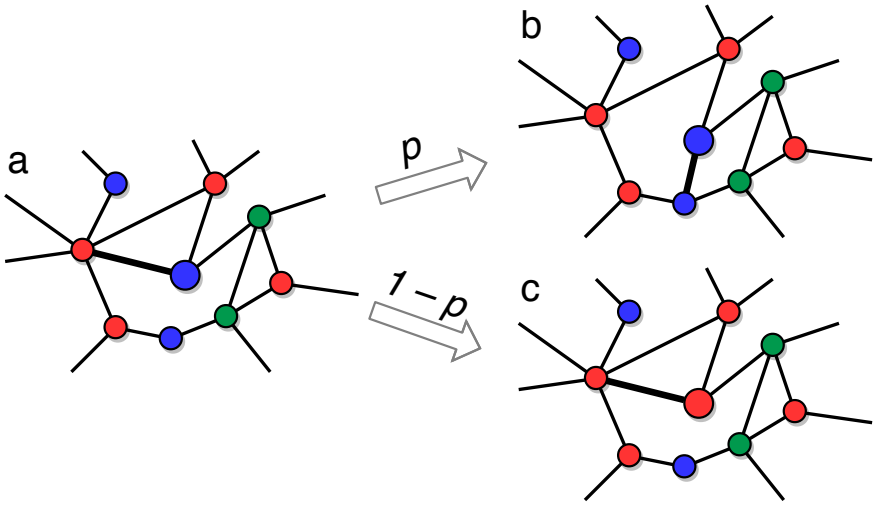


Fig. 7.14 Coevolution of opinions and networks. Opinions are indicated by the colors. (a) A node is chosen (large blue circle in the middle), along with one of its neighbors (red node attached to the thick link). (b) With probability p, the node replaces its neighbor with a node having the same opinion. (c) With probability $1-p$, the node takes the opinion of the neighbor.

Each iteration of the coevolution model requires a sweep over the nodes, synchronously or in random order. When we examine node i, we select a random neighbor j with different opinion from i:

1 With probability p, the link between i and j is rewired from i to a randomly selected non-neighbor holding the same opinion as i (*selection*).
2 Else (with probability $1-p$), i takes the opinion of j (*influence*).

The *selection probability* p is the single parameter of the model.

Since both selection and influence tend to decrease the number of neighboring node pairs with different opinions, the network eventually reaches a state in which all pairs of neighbors have the same opinion. This means that the network will be divided into a set of separate components, disconnected from one another, with all members of each component holding the same opinion, which may differ across components. We will thus observe a segregation into homogeneous opinion communities, as illustrated in Figure 7.15. Such a scenario is a stable state: no more changes in opinions or network structure take place and the dynamics stop.

When the selection probability is close to zero, influence dominates and the network structure barely changes. The system will basically homogenize the opinions within the connected components of the initial network. When the selection proba-

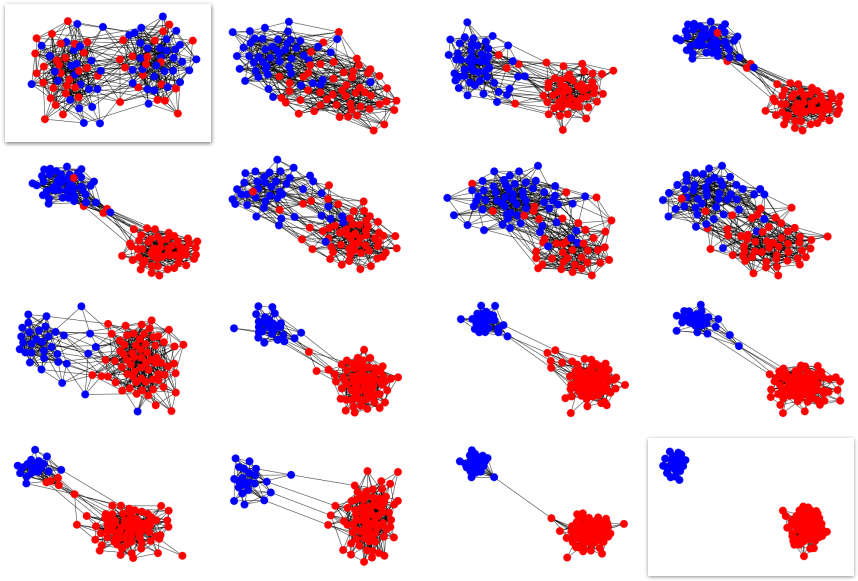


Fig. 7.15 Dynamics of the coevolution model on a network with two communities. Initially (top left), two opinions are randomly distributed among the nodes. The selection probability is $p = 0.7$. Eventually (bottom right), the network becomes segregated into two disconnected components with homogeneous opinions.

bility is close to one, selection dominates and opinions hardly influence each other. Here the final components of the system are the groups of nodes with the same opinion as in the initial configuration.

Let us see what happens when the number of opinions is large. If we start from a random network with average degree larger than one, we know that it has a giant component (Section 5.1), so for selection probability near zero in the long run there will be a giant community holding the majority opinion, and many small communities with different opinions. For selection probability near one, instead, the link dynamics will break the network into many small components, each made mostly of nodes that were initially assigned one of the distinct opinions. It turns out that there is an abrupt transition between the scenario with a large majority opinion and the scenario with many smaller opinion communities of comparable size. This transition takes place at a threshold value of the selection probability.

Models in which people with similar opinions tend to come together can help us study the emergence of echo chambers in social media, as discussed in Section 4.5 and illustrated in Figure 6.2.

7.4 Search

One of the most common activities we perform when interacting with networks is *search*. Suppose you wish to find some resource that is located on some node of a network. It could be a website with information about a topic of interest, a movie stored on a peer network, or a business contact on a social network — not unlike the target person in Milgram's small-world experiment (Section 2.7). To solve these problems, we need to devise strategies to efficiently explore the network, until the right node is reached. One typically starts from a node of origin and proceeds by visiting neighbors, neighbors of neighbors, and so on. The more effective the strategy, the sooner you can reach the target. This section presents a few prevalent search approaches. In particular, we will emphasize how the peculiar properties of real-world networks can be exploited to expedite the search process.

7.4.1 Local Search

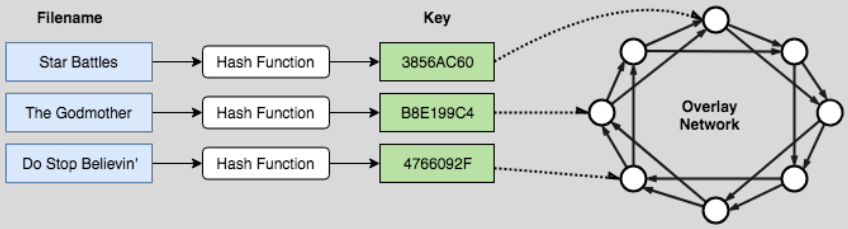
Breadth-first search, presented in Chapter 2, is an attempt to search the whole network by visiting every node, at least in connected components where some seed nodes are known. This type of *exhaustive search* approach solves the problem in some cases, especially when the network is small, or when huge computing and storage resources are available, as demonstrated by Web crawlers that support search engine queries. But often it is more effective, or even necessary, to perform a *local search* of the network, *i.e.*, to perform focused crawls for specific search queries, exploring only a small portion of the network. For instance, you may be interested in a very specific or new piece of Web content that is not listed in a search engine's index. In these cases the search process must employ some heuristics, sorting out the network nodes most likely to contain the desired information.

Another scenario in which local search is necessary is when you wish to download a just-released song from a *peer-to-peer* (or simply *peer*) network, which is a set of personal computers directly connected with each other to share files. Such systems lack a central server that can store the location of every file. This is advantageous because the function of the whole system cannot be compromised by the failure of any single node — say, due to a lawsuit or a denial-of-service attack. The disadvantage is that the location of the desired file is unknown. So, whenever users look for a file, queries are sent to the computers of other users that are connected in the peer network. If a computer does not have the requested file, the query is forwarded to one or more neighbors, and so on.

Breadth-first search can also be used for local search, in principle. Starting from the source, we could visit all the nodes of the first layer, and check if any of them is the target node. If not, each of them forwards the query to all their neighbors, and so on, until the right node is reached. Queries already received from other neighbors are ignored. One of the earliest peer networks, called *Gnutella*, used this approach. But breadth-first search is not an efficient strategy. Most of all, it does not

Box 7.1 Search in peer networks

Peer networks are used for file sharing and have a structure designed to make the search for shared files efficient. This is achieved by a combination of a *distributed hash table* that maps files to peer computers and an *overlay network* that connects these peer nodes.



When a file needs to be stored, a unique *key* is generated for the file. This is done by a *hash function*, an algorithm that produces a unique signature from arbitrary data. A key maps to a specific node in the network, so that the file can be routed to that peer. Similarly, when searching for a file, the key is used to forward the query through the network until it reaches the node that has the file with that key. Each node maintains a set of links to its neighbors — a routing table — that is used to forward messages through the overlay network. The distributed hash table of a particular peer network design encodes rules for maintaining the network structure in such a way that search is fast. In particular, for any key, each node either knows the target node that owns that key or has a link to a node that is closer to the target. Thanks to this property, a simple greedy routing algorithm can be employed to forward a message to the neighbor that is closest to the target. Another important property of the peer network is that any computer can join or leave at any time. When a peer leaves or a new peer joins, only the neighbor peers need to be updated; the rest of the network is unaffected.

take advantage of the structure of the network. In fact, computers on the Gnutella network were flooded with requests and spent all of their bandwidth managing this traffic. That is why Gnutella was eventually replaced by modern peer networks such as *BitTorrent*, that use special network structures designed for efficient search algorithms (Box 7.1).

One way to exploit network structure is to rely on the presence of hubs. A local search algorithm based on this idea assumes that each node knows the degree of all of its neighbors as well as the data stored in them, so all information available to nodes is local. When a neighbor of the target node receives the request, it will reply: "I am not the node you are looking for, but my neighbor is!" and send the address

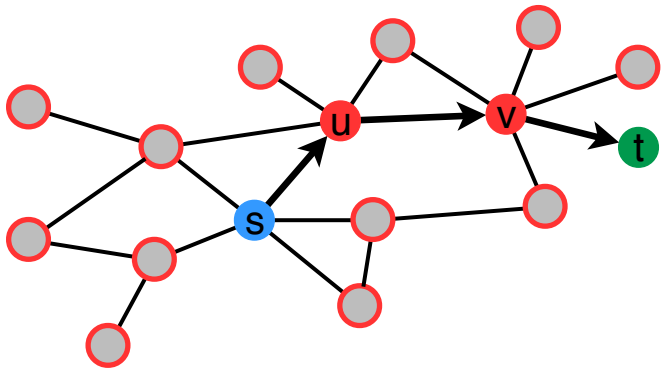


Fig. 7.16 Model of local search on networks. The source is **s**, the target **t**. The source passes the query to its neighbor with highest degree (**u**), which forwards it to its neighbor of highest degree (**v**). Since the target is a neighbor of **v**, the search ends.

of the target node. Each node that is queried, starting from the source, forwards the request to its neighbor with largest degree, unless it or any of its neighbors is the target. The process is repeated until the message is received by a neighbor of the target (Figure 7.16). Since nodes may be visited multiple times during the procedure, those which have passed the request are marked, so that none of them is queried more than once.

In Section 3.3 we have seen that the neighbors of a randomly selected node are more likely to be hubs that the node itself, on average. In particular, by exploring neighbors with large degree, the chance that any of their neighbors is a major hub is higher. Consequently the algorithm quickly reaches the node with largest degree. After the top hub is checked, it is marked and will be avoided in the future. The next hub is then likely to be the one with the second-largest degree, and so on. Basically, after a rapid transient phase, in which the visited nodes have progressively larger degree, the exploration follows the inverse order of the network's degree sequence, from the node with largest degree downwards. The number of queried nodes, which are the neighbors of the hubs, grows very fast and the target is reached in a small number of steps.

While hub-driven local search brings a gain in the number of steps needed to complete the search, the number of nodes that have to be queried is about the same as when using breadth-first search, on average. This is because the target node can be anywhere, in principle, so many checks are necessary in both cases. The fewer steps of the local search algorithm are compensated by the fact that more neighbors are checked at each step, since the nodes traversed during the procedure have large degree. However, if each node knows the information contents of its neighbors, it does not really need to query any of them, which significantly reduces the communication overhead between nodes. This requires hubs to store a huge amount of data, which is unfeasible in very large networks.

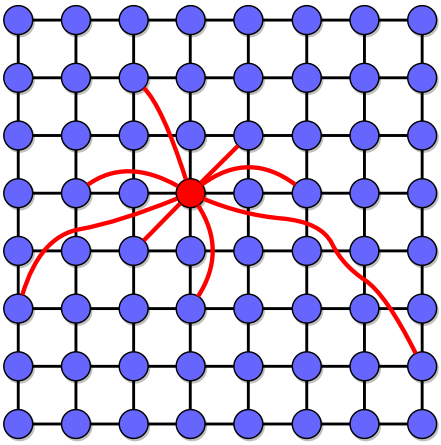


Fig. 7.17 Geographic social network. The square grid represents the geographic area where the people (nodes) live. Each node is linked to its four nearest neighbors. Shortcuts between the nodes are added favoring pairs of individuals living close to each other. The figure only shows the shortcuts of the red node.

7.4.2 Searchability

We have seen a couple of strategies for searching networks. But are all networks "searchable"? Can we search through any graph and expect results in a reasonably short time? The short answer is no, but there are some important exceptions that we discuss next.

To explore the *searchability* properties of a network, recall the small-world experiment by Milgram, presented in Section 2.7. The experiment teaches us two lessons. One is the familiar observation that most pairs of people in a social network are connected via short chains of acquaintances, as we have seen. Lesson number two is that people are surprisingly effective at finding those chains. This is not straightforward: participants knew only their contacts and the name and location of the target person. They had to trust their instinct in the choice of the friend to whom to forward the letter, hoping to get it closer to the target. Most participants tried to send the letter such that it could reach as quickly as possible the Boston area, where the target person lived. This exploits the homophily of the network (discussed in Section 2.1), in particular *geographic homophily*: two people are more likely to know each other if they live nearby. Still, in principle, the letter could have lingered in Boston for a long time once there, being passed among many people before finally reaching the target. Successful participants used some additional intuition about the network structure to find the target in a few steps. They exploited different kinds of homophily based on occupations, say: a lawyer is likely to know another lawyer. This is closely related to topical locality on the Web (Section 4.2.5).

It is possible to analyze the conditions that a network must satisfy in order to

be searchable using heuristics based on the types of homophily described above — connecting to a node that is geographically or topically close to the target. Let us first focus on *geographic searchability*. It turns out that there are narrow conditions that make a network geographically searchable. To illustrate this, consider a special structure resembling small-world networks generated by the model discussed in Section 5.2. We start from a square grid, which serves the purpose of embedding the social network in geographic space, like placing people on a map. Each node is connected to its nearest neighbors, forming a grid network. We then add shortcuts between pairs of nodes of the grid (Figure 7.17). At variance with the small-world model (Figure 5.4(b)), the shortcuts do not connect pairs of nodes with equal probability; rather, the link probability decreases with the geographic distance between the nodes in the grid. This is designed to account for geographic homophily, the empirical observation that most relationships in real social networks occur among people in geographic proximity of each other.

Let us assume that each individual knows exactly the geographic position of their neighbors, as well as the position of the target. Therefore each individual can precisely determine which neighbor is geographically closest to the target. For sake of simplicity, let us further assume that the source and target nodes are chosen at random, and that people follow the *greedy search algorithm* inspired by Milgram's experiment: each node forwards the message across a link that brings it as close as possible to the target. We can define the *delivery time* as the number of times the message is passed between nodes until it reaches the target. As it turns out, the delivery time is very short only if the shortcut probability falls off in just the right way as a function of the geographic distance between nodes. In the case of a two-dimensional grid as shown in Figure 7.17, the probability of a shortcut must decay as the inverse of the square of the distance. For example, a link between two nodes lying two steps apart from each other should be four times more likely than a link connecting two nodes that are twice as far (four steps).

If the shortcut probability falls more rapidly with the distance between the nodes, there are not enough long-range links, so that one is doomed to traverse many local links before reaching the target. If the shortcut probability falls more slowly, there are too many long-range links. In this scenario there are many short paths, but they are hard to find, like searching for a needle in a haystack. In both cases, the search process is not very efficient and the greedy search algorithm needs a long time to find the target.

Although the condition for geographic searchability of a network is quite narrow in this scenario, it is not entirely unrealistic. In the Web, if we replace the notion of geographic homophily with that of topical locality, we can empirically measure the probability that two pages are linked as a function of their topical distance. Imagine that the grid of Figure 7.17 represents a topical landscape, and that nearby points represent related Web pages. In practice, we can measure the similarity between two pages by looking at their content (recall Box 4.1). Small similarity values can be mapped to large distances and vice-versa. It turns out that nearby (similar) pages are highly likely to have common neighbors or be linked, while for distant

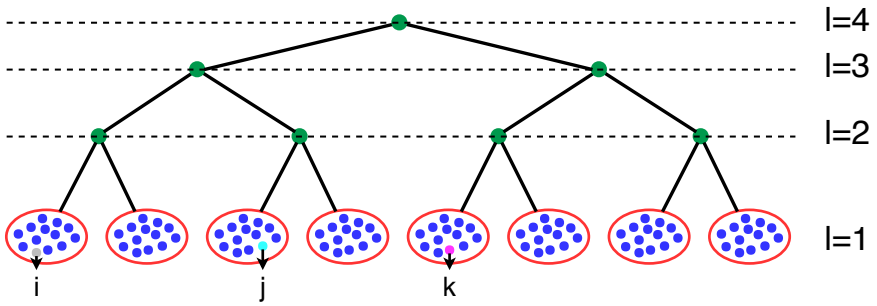


Fig. 7.18 Topical distance tree. The distance between nodes i and j is three, because the nearest common ancestor of the groups to which i and j belong is at level three (green dot on the left of the $l = 3$ dashed line). Likewise, the topical distance between i and k or j and k is $l = 4$ because the nearest common ancestor is the root.

(dissimilar) pages the decay in link probability is compatible with the geographic searchability condition. The Web is therefore a special case of a searchable network, which is reassuring as it means we can find interesting information by clicking on links. If this were not the case, surfing the Web would be hopeless.

The network model used to explore geographic searchability is unrealistic in many ways. People are not located and connected as the nodes of a grid. More importantly, geography is just one of the many possible attributes of the nodes in a network being searched. Two people in a social network may have the same job, practice the same hobby, attend the same school, and so on. Let us generalize the notion of searchability to *topical searchability*, where any attribute of the nodes can be reflected in network homophily and thus facilitate the search process. For instance, as mentioned earlier, the occupation of the target node was a useful piece of information in Milgram's experiment.

We can group the nodes of a network in a hierarchical manner based on their topical attributes: the top of the hierarchy represents the most general category, which is split into smaller, increasingly specific topical categories as we go down, until we reach the smallest groups that we can identify. The resulting hierarchical diagram is a *topical distance tree*, illustrated in Figure 7.18. A topical distance tree could be used to organize Wikipedia articles about science. At the top (below the root) would be the formal sciences, physical sciences, life sciences, social sciences, and applied sciences. In the level below we would find disciplines like math, logic, biology, chemistry, physics, psychology, economics, sociology, engineering, computer science, and so on. More specific fields, such as molecular biology, statistical physics, machine learning, and network science would be placed at lower levels. Similarly, one can use a topical tree to classify people in a social network. At the top would be the whole world population and the lower groups could represent a geographic subdivision of the population into continents, countries, cities, and neighborhoods.

Different social attributes (*e.g.*, occupations, hobbies, schools, religions) lead to different divisions and trees.

A topical distance tree is a mental construct that allows us to estimate the *topical distance* between nodes (Figure 7.18). If two individuals belong to the same smallest identifiable group, their topical distance is one. This would be the case, say, of two professors working in the same department at Indiana University in Bloomington. Otherwise, their groups will eventually merge as we climb up the hierarchical tree. This happens when we bump into their *nearest ancestor* category in the tree, which represents the most specific attribute shared by the nodes. In this case, the topical distance is given by the number of levels in the tree, from the bottom up to the nearest common ancestor. For instance, in the schematic diagram of Figure 7.18, individuals i and j could be two professors working in distinct departments of different universities in Indiana, so their topical distance is three, because working on separate topics and in different locations adds two degrees of separation.

Let us stick with the social network scenario and assume that people can estimate their topical distance from anybody. This is a less stringent hypothesis than in the geographic model, where individuals know each other's exact position. Let us further assume that the topical distance tree captures the social network's homophily, so that the link probability between two nodes decreases as their topical distance increases, according to a decay function. By using the greedy search algorithm, *i.e.*, by letting each person forward the message to the neighbor with the shortest topical distance from the target, it can be shown that there is a special topical decay function that allows for efficient search. In this condition the search takes a small number of steps.

The condition for topical searchability of the network, expressed by the relationship between topical distance and link probability, is quite strict. However, it is sociologically plausible, helping us understand the successful chains in Milgram's experiment. Furthermore, one can measure how the probability that two Web pages are linked decays with their topical distance by analyzing pages classified in a topical Web directory. It turns out that the Web graph meets the topical searchability condition as well, confirming that it is searchable by surfing.

7.5 Summary

Networks are vehicles for the diffusion of ideas, opinions, and influence. By the same token, they facilitate harmful spreading processes, like the diffusion of infections, misinformation, and rumors. Uncovering how these phenomena unfold can help us improve the effectiveness of the former and defend ourselves from the latter. Searching networks is critical for retrieving information, but difficult when the network structure and the content stored by the nodes are unknown.

In this chapter we have reviewed simple models describing these processes and learned the following key lessons:

1 In threshold models of influence diffusion, a node/individual is subject to the combined effect of all of its neighbor influencers: when this effect exceeds a threshold, the node is affected. In independent cascade models, a node/individual is "convinced" by each neighbor influencer with a certain probability. The most effective influencers have large degree and a central position in the network.

2 In the Susceptible-Infected-Susceptible (SIS) model of epidemic spreading, when infected individuals recover they become susceptible again, so they can contract the disease multiple times. In the Susceptible-Recovered-Susceptible (SIR) model, when infected individuals recover they cannot be infected any more, so they play no further role in the dynamics.

3 If the contact networks have hubs, a disease spreading according to both SIR and SIS dynamics can affect an important fraction of the population, even if the probability of infection is low, because the hubs can be easily infected and turn into dangerous spreaders.

4 The rumor spreading model is similar to SIR, but the "recovery" process, corresponding to the decision of not spreading the rumor further, is a consequence of encounters between individuals who know the rumor, instead of happening spontaneously for each individual. The rumor can reach a significant portion of any network even for low transmission probability.

5 In the majority opinion model, a node takes the opinion of the majority of its neighbors. Different opinions coexist in the final state. Consensus is only reached on one- and two-dimensional grids; in these cases, the consensus opinion is the majority opinion in the initial configuration.

6 In the voter model, a node takes the opinion of a randomly selected neighbor. The dynamics lead to consensus on all networks. Consensus on an opinion is reached with a probability that matches the fraction of nodes holding that opinion in the initial configuration.

7 In bounded-confidence models of continuous opinion dynamics, two opinions can affect each other only if their difference is smaller than the confidence bound parameter. The final number of opinion clusters depends on the value of the confidence bound and the network structure. With a sufficiently large confidence bound, the dynamics lead from random initial opinions to consensus on any network.

8 Coevolution models combine the processes of selection and social influence. We presented a model in which a node can either take the opinion of a neighbor or select a new neighbor with the same opinion. In the final state, the

system is segregated into homogeneous opinion communities, disconnected from each other.

9 For an exhaustive search of a network, like those performed by Web crawlers, the standard approach is breadth-first search, the same algorithm used to compute distances and find shortest paths between nodes. This can be unfeasible for large networks, so that local heuristic search becomes necessary. One local search heuristic is to forward the query to the neighbor nodes with largest degree, so that we can quickly reach the biggest hubs and exploit their large numbers of neighbors to find the target in a small number of steps.

10 Some networks are searchable, in that one can find short paths that connect a source to a target. This may be due to a peculiar geographic distribution of links between nodes, or to a hierarchical organization of nodes according to their content or attributes. By estimating the distance between two nodes in the hierarchy, one can identify the neighbor closest to the target.

7.6 Further Readings

Most general books on network science, like the ones recommended in Section 1.12, have ample sections on dynamic processes. The book by Barrat et al. (2008) is dedicated to the topic, and covers in detail most of the models presented in this chapter.

The science behind the spread of misinformation is an emerging area of research (Lazer et al., 2018). The study of information diffusion networks (Shao et al., 2018a) is critical to help us understand how social media can be manipulated, for example via social bots (Shao et al., 2018b).

Threshold models were introduced in a classic paper by Granovetter (1978), while the independent cascade model is more recent (Goldenberg et al., 2001). Watts (2002) proposed to impose a threshold on the fraction of neighbors, instead of their number. Kempe et al. (2003) addressed the problem of identifying the set of influencers who can generate the largest cascades. Kitsak et al. (2010) showed that the hubs are not necessarily the most effective influencers. Centola and Macy (2007) explored complex contagion in the spread of collective behaviors. Weng et al. (2013b) showed that communities affect the viral spread of memes in social media, and that cascade sizes can be predicted based on how many communities are involved in the early stages of diffusion.

The book by Anderson and May (1992) is a good reference for classic epidemic modeling. Pastor-Satorras et al. (2015) published a comprehensive review of epidemic processes on networks. Stehlé et al. (2011) reconstructed the network of face-to-face interactions among children and teachers in a school, by means of radio frequency identification devices. The lack of an epidemic threshold on networks with hubs was first exposed by Pastor-Satorras and Vespignani (2001). Cohen et al.

(2003) suggested that immunizing the acquaintances of randomly selected individuals is an effective strategy if the networks of contacts have heavy-tail degree distributions. Christakis and Fowler (2010) showed that monitoring the friends of randomly selected individuals allows early detection of epidemic outbreaks. The rumor spreading model was first presented by Daley and Kendall (1964).

Castellano et al. (2009) review opinion and other social dynamics models from the point of view of statistical physics. The majority model was originally introduced in the context of spin models in statistical physics (Glauber, 1963). Another model based on the concept of majority, not discussed in this chapter, is called *majority rule model* (Galam, 2002; Krapivsky and Redner, 2003). The voter model was proposed to describe the territorial competition among species (Clifford and Sudbury, 1973). Mobilia et al. (2007) studied the role of zealots in the voter model. Vazquez et al. (2003) developed the *constrained voter model*, in which only similar opinions can interact. The principle of bounded confidence dates back to Festinger's (1954) theory of social comparison. The original bounded-confidence opinion model was introduced by Deffuant et al. (2000). The first models of coevolution of network dynamics and structure were proposed by Holme and Newman (2006) and Gil and Zanette (2006).

Adamic et al. (2001) proposed the local search strategy that relies on the presence of hubs in the network. The geographic network and the relative analysis of network searchability were presented by Kleinberg (2000). Analyses of searchability based on topical hierarchies and distances were put forward independently by Kleinberg (2002) and Watts et al. (2002). Menczer (2002) showed that the Web graph satisfies versions of geographic and topical searchability.

Exercises

7.1 Go through the Chapter 7 Tutorial on the book's Github repository.[1] It provides a class that makes it easy to code and run simulations of network dynamics models.

7.2 Consider the example in Figure 7.19. According to the linear threshold model, will node **1** be activated if its threshold is 4? What if it is 5? Do the answers to these questions change if we vary the weights of the links joining node **1** to its inactive neighbors?

7.3 Someone gives you a network with some of its nodes activated. She claims that you will never succeed in activating all the nodes, no matter which model of influence spreading you use. How can she be so sure?

7.4 Apply the fractional threshold model to the network in Figure 7.20. The threshold is 1/2 for all nodes. Which node should we activate to obtain the

[1] `github.com/CambridgeUniversityPress/FirstCourseNetworkScience`

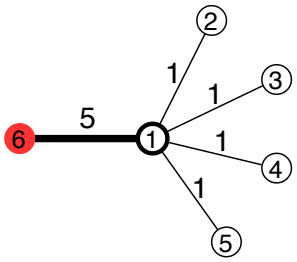


Fig. 7.19 A weighted influence network. Node **1** has only one active neighbor (**6**).

largest cascade? Is the solution unique? What is the minimum number of initial influencers that are needed to activate the whole network?

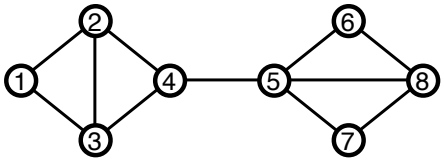


Fig. 7.20 An influence network. Each node has a threshold of 1/2.

7.5 You are considering the independent cascade model on a network. Two active nodes **s** and **t** have degree 4 and 10, respectively. They can convince their neighbors with probability 1/2 (**s**) and 1/5 (**t**). Which node will influence more neighbors on average, **s** or **t**?

7.6 Consider the network in Figure 7.21: the influence probabilities are symmetric, for example the probability that node **1** convinces node **2** is equal to the probability that **2** convinces **1**. Use the independent cascade model to predict how many nodes will be active in the end, on average, by activating node **2** initially.

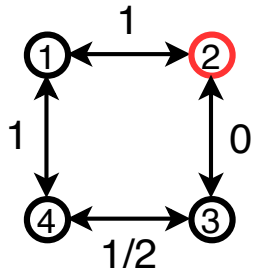


Fig. 7.21 A network with symmetric influence probabilities shown next to the links. Node **2** is active.

7.7 Contagion models such as SIS and SIR come from epidemiology, but as it turns out, they can model other spreading processes on networks quite well. Which of the following processes could best be described by an SIS model on a network?

a. The spread of toxic gas through the air over a geographic region
b. The spread of an oil slick over the surface of a body of water
c. The impact of a power station failure in the US power grid
d. The adoption of a specific smartphone among members of a community

7.8 The game Pandemic II (`pandemic2.org`) is based on an elaborate SIR model. Play the game and write a brief report on how the various aspects of the game correspond to SIR model mechanisms. Discuss key simplifying assumptions made in the game. Describe how various game choices affect the model parameters.

7.9 Consider SIS model dynamics on a population. Suppose that a fraction f of the population never gets sick and that such immune individuals are randomly distributed in a homogeneous contact network (all nodes have similar degree). Is the risk of epidemic spreading bigger or smaller than in the pure SIS model, for which $f = 0$? Would the answer change if we considered SIR instead? Hint: use the condition of Eq. 7.5.

7.10 There is an epidemic outbreak and after a quick verification it turns out that the basic reproduction number is $R_0 = 2.5$, so we are heading towards an epidemic spread (assume that the contact network is homogeneous). The authorities urge the population to limit their contacts with other people, so that, on average, each individual gets in touch with about half the usual number of people. Suppose that doctors are capable of developing medicines that can significantly increase the recovery rate μ. How much does μ have to increase so that the epidemics can be stopped?

7.11 Simulate the SIR dynamics on a random network with $N = 1000$ nodes and link probability $p = 0.01$. Initially ten nodes are infected, chosen at random. The probability of recovery is $\mu = 0.5$. Run the dynamics for these values of the infection probability: $\beta = 0.02, 0.05, 0.1, 0.2$. In each run, save the number of simultaneously infected people after each iteration and calculate the maximum value. Interpret the results. How many iterations are needed to reach the maximum? Do you observe a major outbreak? Why or why not? (Hint: feel free to modify the code in this chapter's tutorial to run the simulations.)

7.12 In a community there are three kinds of people: frustrated (S), aggressive (I), and peaceful (R). When a frustrated individual meets an aggressive one, they become aggressive with probability β. When an aggressive individual meets a peaceful one, they become peaceful with probability α. When two aggressive people meet each other, they start to argue. But with probability α, they realize after a while that fighting is futile, and so they both turn peaceful. Can a major spread in aggressive behaviors be prevented by a small value of β?

7.13 In the network illustrated in Figure 7.22, each node has one of two possible opinions. An *active link* connects nodes in different opinion states. These links are called active because in theory either endpoint has a chance to convince the other to adopt its opinion, depending on the specific rules of the model. How many active links are there?

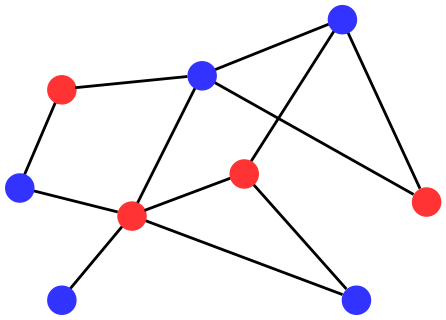


Fig. 7.22 A network with nodes colored red or blue according to their opinions.

7.14 When we simulate a dynamic process on a network, there are several ways to asynchronously pick the next node(s) to update. Typically nodes are selected in random sequence. Another strategy would be to select one endpoint of a randomly selected link. Do you think that this would affect the dynamics in any way? Why or why not?

7.15 Simulate the majority opinion dynamics on a square grid with $N = 20 \times 20 = 400$ nodes. Initially assign each of two opinions to half of the nodes chosen at random. Execute 100 runs with different initial random assignments, until the system gets to a stationary state. How many runs lead to consensus? Create a histogram of the proportion of opinion one in the non-consensus stationary states. Create a histogram of the fraction of active links in those configurations. (Active links are defined in Exercise 7.13. The fraction of active links is the ratio between the number of active links and the total number of links in the network.) Hint: If you use the code in this chapter's tutorial to run the simulations, to guarantee convergence to a stationary state you have to write the `state_transition()` function in such a way that nodes are updated in asynchronous fashion and random order. You also have to specify a stop condition function to end the simulation when the stationary state is reached.

7.16 Compute the exit probability of the majority opinion model on a square grid with $N = 20 \times 20 = 400$ nodes. Let the initial fraction of nodes with opinion one be $p = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$. Execute 20 runs with different initial random assignments for each value of p, until the system gets to a stationary state. Consider only the runs leading to consensus, and for each p compute the fraction of those runs for which consensus is on opinion one, which is the exit probability for that value of p. Plot the result as a function

of p. (Hint: feel free to modify the code in this chapter's tutorial to run the simulations, as explained in the previous exercise.)

7.17 Compute and plot the exit probability of the voter model on the square grid. Use the same parameters as in Exercise 7.16. (Hint: feel free to modify the code in this chapter's tutorial to run the simulations, as explained in the previous exercises.)

7.18 Consider the bounded-confidence model of opinion dynamics on a complete network. Since all nodes are connected to each other, any two nodes can affect each other's opinion if they are sufficiently close. A mathematical argument shows that if the initial opinions are randomly distributed in the interval $[0, 1]$, the number of final opinion clusters in this case is approximately equal to $\frac{1}{2\epsilon}$, where ϵ is the confidence bound. If you are mathematically inclined, can you give an intuition into this argument?

7.19 Simulate the bounded-confidence model of opinion dynamics on a complete network with $N = 1000$ nodes. The initial opinions are random numbers between zero and one. Consider three different values of the confidence bound: $\epsilon = 0.125, 0.25, 0.5$. For each ϵ, use different values for the convergence parameter, say $\mu = 0.1, 0.3, 0.5$. Run every simulation until each opinion varies by less than 1% between consecutive iterations, and plot a histogram of the final opinions. Does the number of final opinion clusters depend on ϵ? Why or why not? Does it depend on μ? Why or why not? (Hint: feel free to modify the code in this chapter's tutorial to run the simulations.)

7.20 Simulate the bounded-confidence model of opinion dynamics on a random network with $N = 1000$ nodes and link probability $p = 0.01$. The initial opinion configuration is generated by assigning to each node a random number between zero and one. Set the parameter $\mu = 1/2$ and explore different values of the confidence bound ϵ. Run every simulation until each opinion varies by less than 1% between consecutive iterations. What is the threshold ϵ_c such that, for $\epsilon > \epsilon_c$, we have a single opinion cluster (consensus) in the final configuration? Now simulate the model on a small-world network with $N = 1000$ nodes, $k = 4$, and rewiring probability $p = 0.01$. What is ϵ_c in this case? (Hint: feel free to modify the code in this chapter's tutorial to run the simulations.)

7.21 Consider the coevolution model with just two opinions, initially distributed randomly among the nodes. How many opinion communities do you expect there will be when selection dominates (p close to 1)? What's their size, approximately? (Hint: You may assume that the network is not too sparse.)

7.22 In the coevolution model, the influence component follows the rule of the voter model, in that the node takes the opinion of a random neighbor. Let us see what happens if we switch to majority dynamics. The new model works like this: given a node, with probability p it rewires one of its links to a non-neighbor node with the same opinion, like before; with probability $1-p$ it takes the majority opinion of its neighborhood. Describe the final configurations you

expect to observe when the system reaches the stable state in the extreme cases of p close to zero and one.

7.23 Build small-world networks with $N = 1000$ nodes, $k = 4$, and these values for the rewiring probability: $p = 0.001, 0.01, 0.1, 1$. Choose a source node $\mathbf{s}$ and a target node $\mathbf{t}$ at random. Apply the greedy search algorithm, where the message is passed to the neighbor that is closest to the target along the ring, and compute the number of steps needed to deliver the message from $\mathbf{s}$ to $\mathbf{t}$ for each value of p. Interpret the results. (Hint: for each p, average your measurement across multiple runs with different random pairs of nodes.)

7.24 The topical distance tree in Figure 7.18 is very stylized and unrealistic. Real topical distance trees are generally asymmetric, like the one in Figure 7.23. What is the topical distance between the two marked individuals?

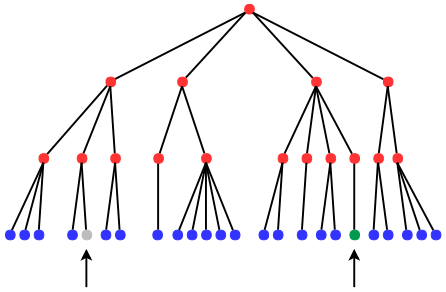


Fig. 7.23 A topical distance tree.