8 How do I influence people on Facebook and Twitter?

To study a network, we have to study both its *topology* (the graph) and its *functionalities* (tasks carried out on top of the graph). This chapter on topology-dependent influence models does indeed pursue both, as do the next two chapters.

8.1 A Short Answer

Started in October 2003 and formally founded in February 2004, Facebook has become the largest social network website, with 900 million users worldwide as of spring 2012 at the time of its IPO. Many links have been formed among these nodes, although it is not straightforward to define how many mutual activities on each other's wall constitute a "link."

Founded in July 2006, Twitter attracted more than 500 million users in six years. At the end of 2011, over 250 million tweets were handled by Twitter each day. Twitter combines several functionalities into one platform: microblogging (with no more than 140 characters), group texting, and social networking (with one-way following relationships, i.e., directional links).

Facebook and Twitter are two of the most influential communication modes, especially among young people. For example, in summer 2011's east-coast earthquake in the USA, tweets traveled faster than the earthquake itself from Virginia to New York. They have also become a major mechanism in social organization. In summer 2009, Twitter was a significant force in how the Iranians organized themselves against the totalitarian regime.

There have been all kinds of attempts at figuring out

- (1) how to quantify the statistical properties of opinions on Facebook or Twitter;
- (2) how to measure the influential power of individuals on Facebook or Twitter; and
- (3) how to leverage the knowledge of influential power's distribution to actually influence people online.

For example, on question (1), our recent study of all the tweets about Oscar-nominated movies during the month of February 2012 shows a substantial skew towards positive opinion relative to other online forums, and the need to couple

tweet analysis with data from other venues like the IMDb or Rotten Tomato to get a more accurate reading of viewer reception and prediction of box office success.

Question (2) is an analysis problem and question (3) a synthesis problem. Neither is easy to answer; and there is a significant gap between theory and practice, perhaps the biggest such gap you can find in this book. Later in this chapter, we will visit some of the fundamental models that have yet to make a significant impact on characterizing and optimizing influence over these networks.

But the difficulty did not prevent people from trying out heuristics. Regarding (2), for example, there are many companies charting the influential power of individuals on Twitter, and there are several ways to approximate that influential power: by the number of followers, by the number of retweets (with "RT" or "via" in the tweet), or by the number of repostings of URLs. There are also many companies data-mining the friendship network topology of Facebook.

As to (3), Facebook uses simple methods to recommend friends, which are often based on email contact lists or common backgrounds. Marketing firms also use Facebook and Twitter to stage marketing campaigns. Some "buy off" a few influential individuals on these networks, while others buy off a large number of randomly chosen, reasonably influential individuals.

It is important to figure out who the influential people are. An often-quoted historical anecdote concerns the night rides by Paul Revere and by William Dawes on 18-19 April in 1775. Dawes left Boston earlier in the evening than did Revere. They took different paths towards Lexington, before riding together from Lexington to Concord. Revere alerted influential militia leaders along his route to Lexington, and was therefore much more effective in spreading the word of the imminent British military action. This in turn lead to the American forces winning on the next day the first battle that started the American Revolutionary War.

How do we quantify which nodes are more important? The question dates back thousands of years, and one particularly interesting example occurred during the Renaissance in Italy. The Medici family was often viewed as the most influential among the 15 prominent families in Florence during the fifteenth and sixteenth centuries. As shown in Figure 8.1, it sat in the "center" of the family social network through strategic marriages. We will see several ideas quantifying the notion of centrality.

How do we quantify which links (and paths) are more important? We will later define strong vs. weak ties. Their effects can be somewhat unexpected. For example, Granovetter's 1973 survey in Amherst, Massachusetts showed the strength of weak ties in spreading information. We will see another surprise of weak ties' roles in social networks, on six-degree separation in Chapter 9.

Furthermore, how do we quantify which subset of nodes (and the associated links) are connected enough among themselves, and yet disconnected enough from the rest of the network, that we can call them a "group"? We save this question for the Advanced Material.

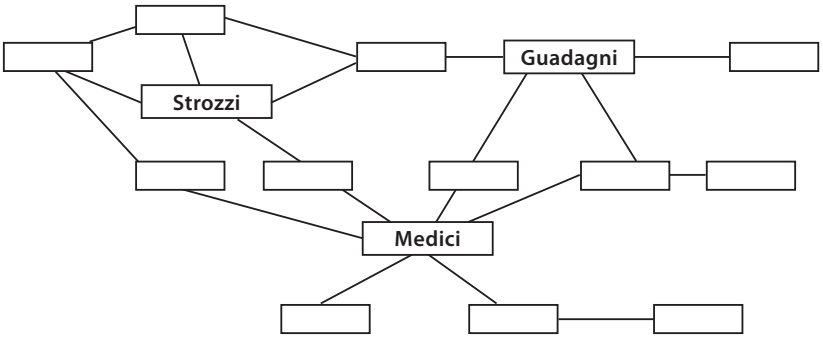

Figure 8.1 Padgett's Florentine-family graph shows the central position of the Medici family in Renaissance Florence. Each node is a family, three of them shown with their names. Each link is a marriage or kinship relationship. The Medici family clearly had the largest degree, but its influential power relative to the other families was much more than the degree distribution would indicate. Other measures of centrality, especially betweenness centrality, reveal just how influential the Medici family was.

8.1.1 Graphs and matrices

Before proceeding further, we first formally introduce two commonly used matrices to describe graphs. We have seen a few different types of graphs in the previous chapters. In general, a graph G is a collection of two sets: V is the set of vertices (nodes) and E is the set of edges (links). Each link is in turn a directed two-tuple: the starting and the ending nodes of that link.

We will construct a few other matrices later as concise and useful representations of graphs. We will see that properties of a graph can often be summarized by linear-algebraic quantities about the corresponding matrices.

The first is the **adjacency matrix A**, of dimension $N \times N$, of a given graph $G = (V, E)$ with N nodes connected through links. For the graphs we deal with in this book, A_{ij} is 1 if there is a link from node i to j, and 0 otherwise. We mostly focus on **undirected graphs** in this chapter, where each link is **bidirectional**. Given an undirected graph, $\mathbf{A}$ is symmetric: $A_{ij} = A_{ji},\ \forall i, j$. If a link can be **unidirectional**, we have a **directed graph**, like the Twitter following relationship graph.

The second, which is less used in this book, is the **incidence matrix $\hat{\mathbf{A}}$**, of dimension $N \times L$, where N is again the number of nodes and L the number of links. For an undirected graph, $\hat{A}_{ij} = 1$ if node i is on link j, and 0 otherwise. For a directed graph, $\hat{A}_{ij} = 1$ if node i starts link j, $\hat{A}_{ij} = -1$ if node i ends link j, and $\hat{A}_{ij} = 0$ otherwise.

Straightforward as the above definitions may be, it is often tricky to define what exactly constitutes a link between two persons: being known to each other by first-name as in Milgram's small-world experiment? Or "friends" on Facebook who have never met or communicated directly? Or only those to whom you text at least one message a day? Some links are also directional: I may have

commented on your wall postings on Facebook but you never bothered reading my wall at all. Or I may be following your tweets, but you do not follow mine.

Even more tricky is to go beyond the simple static graph metrics and into the functionalities and dynamics on a graph. That is a much tougher subject. So we start with some simple static graph metrics first.

8.2 A Long Answer

8.2.1 Measuring node importance

You may be in many social networks, online as well as offline. How important are you in each of those networks? Well, that depends on how you define the "importance" of a node. It depends on the specific functionalities we are looking at, and it evolves over time. But we shall restrict ourselves to just static graph metrics for now. Neither is it easy to discover the actual topology of the network. But let us say for now that we are given a network of nodes and links.

There are at least four different approaches to measuring the importance, or **centrality**, of a node, say node 1.

The first obvious choice is **degree**: the number of nodes connected to node 1. If it is a directed graph, we can count two degrees: the in-degree: the number of nodes pointing towards node 1, and the out-degree: the number of nodes that node 1 points to. **Dunbar's number**, usually around 150, is often viewed as the number of friends a typical person may have, but the exact number of course depends on the definition of "friends." The communication modes of texting, tweeting, and blogging may have created new shades of definition of "friends." In Google+, you can also create your own customized notions of friends by creating new circles.

We will see there are many issues with using the degree of a node as its centrality measure. One issue is that if you are connected to more-important nodes, you will be more important than you would be if you were connected to less-important nodes. This may remind you of PageRank in Chapter 3. Indeed, we can take PageRank's importance scores as a centrality measure.

A slightly simpler but still useful view of centrality is to just look at the successive multiplication of the centrality vector $\mathbf{x}$ by the adjacency matrix $\mathbf{A}$ that describes the network topology, starting with an initialization vector $\mathbf{x}[0]$:

$$\mathbf{x}[t] = \mathbf{A}^t \mathbf{x}[0].$$

In a homework problem, you will discover a motivation for this successive multiplication.

We can always write a vector as a linear combination of the eigenvectors $\{\mathbf{v}_i\}$ of $\mathbf{A}$, arranged in descending order of the corresponding eigenvalues and indexed by i, for some weight constants $\{c_i\}$. For example, we can write the vector $\mathbf{x}(0)$

as follows:

$$\mathbf{x}[0] = \sum_i c_i \mathbf{v}_i.$$

Now we can write $\mathbf{x}[t]$ at any iteration t as a weighted sum of $\{\mathbf{v}_i\}$:

$$\mathbf{x}[t] = \mathbf{A}^t \sum_i c_i \mathbf{v}_i = \sum_i c_i \mathbf{A}^t \mathbf{v}_i = \sum_i c_i \lambda_i^t \mathbf{v}_i, \tag{8.1}$$

where $\{\lambda_i\}$ are the eigenvalues of $\mathbf{A}$.

As $t \to \infty$, the effect of the largest eigenvalue λ_1 will dominate, so we approximate by looking only at the effect of λ_1. Now the **eigenvector centrality** measures $\{x_i\}$ constitute a vector that solves

$$\mathbf{A}\mathbf{x} = \lambda_1 \mathbf{x},$$

which means

$$x_i = \frac{1}{\lambda_1} \sum_j A_{ij} x_j, \quad \forall i. \tag{8.2}$$

We can also normalize the eigenvector centrality $\mathbf{x}$.

The third notion, **closeness centrality**, takes a "distance" point of view. Take a pair of nodes (i, j) and find the shortest path between them. It is not always easy to compute the shortest path, as we will see in Chapters 9 and 13. But, for now, say we have found the shortest paths between each pair of nodes, and denote their lengths as $\{d_{ij}\}$. The largest d_{ij} across all (i, j) pairs is called the **diameter** of the network. The average of d_{ij} for a given node i across all other $n-1$ nodes is an average distance $\sum_j d_{ij}/(n-1)$. The closeness centrality is the reciprocal of this average:

$$C_i = \frac{n-1}{\sum_j d_{ij}}. \tag{8.3}$$

We have used the arithmetic mean of $\{d_{ij}\}$, but could also have used other "averages," such as the harmonic mean.

Closeness centrality is quite intuitive: the more nodes you know or the closer you are to other nodes, the more central you are in the network. But there is another notion that is just as useful, especially when modeling influence and information exchange: **betweenness centrality**. If you are on the (shortest) paths of many *other* pairs of nodes, then you are important. (We can also extend this definition to incorporate more than just the shortest paths, as in the context of Internet routing in Chapter 13.)

Let g_{st} be the total number of shortest paths between two diffferent nodes, source s and destination t (neither of which is node i itself), and let n_{st}^i be the number of such paths that node i sits on. Then the betweenness centrality of node i is defined as

$$B_i = \sum_s \sum_{t<s} \frac{n_{st}^i}{g_{st}}, \tag{8.4}$$

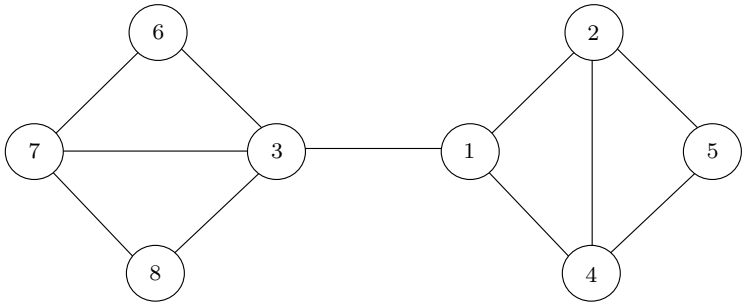

Figure 8.2 An example network to illustrate node-importance metrics, link importance metrics, and group connectedness metrics. For example, how much more important node 1 is than node 2 depends on which centrality metric we use.

where the double summation is indexed such that double counting is avoided.

A node with a large closeness centrality need not have a large betweenness centrality, and vice versa. In fact, many social networks exhibit the small-world phenomenon as explained in Chapter 9, and the closeness centrality values of different nodes tend to be close to each other. Betweenness centrality values tend to have a larger dynamic range.

We can also think of hybrid metrics. For example, first weight each node by eigenvector centrality, then weight each node pair by the product of their eigenvector centrality values, and then calculate the betweenness centrality by weighting each (st) term in (8.4) accordingly.

In the Renaissance Florence family-relationship graph, it is obvious that the Medici family has the largest degree, 6, but the Strozzi and Guadagni families' degrees are not too far behind: 4 for both. But if we look at the betweenness centrality values, Medici family has a value of 50.83, which is five times as central as Strozzi and twice as central as Guadagni, not just a mere factor of 1.5.

Now let us take a look at how different node-importance metrics turn out to be for two of the nodes, 1 and 2, in the small network in Figure 8.2.

For degree centrality, obviously $d_1 = d_2 = 3$. But nodes 1 and 2 cannot be the same in their importance according to most people's intuition that says nodes gluing the graph together are more important.

Let us compute the eigenvector centrality next. From the adjacency matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

we can solve for

$$\mathbf{A}\mathbf{x} = \lambda_1 \mathbf{x}$$

to obtain

$$\begin{aligned}\lambda_1 &= 2.8723,\\ \mathbf{x} &= [0.4063, 0.3455, 0.4760, 0.3455, 0.2406, 0.2949, 0.3711, 0.2949]^T.\end{aligned}$$

So node 1 is slightly more important than node 2: $0.4063 > 0.3415$. But is it really just *slightly* more important?

To compute the closeness centrality, we first write out the pairwise (shortest) distances from node 1 to the other nodes, and from node 2 to the other nodes:

$$\begin{aligned}d_{12} = 1, d_{13} = 1, d_{14} = 1, d_{15} = 2, d_{16} = 2, d_{17} = 2, d_{18} = 2,\\ d_{21} = 1, d_{23} = 2, d_{24} = 1, d_{25} = 1, d_{26} = 3, d_{27} = 3, d_{28} = 3.\end{aligned}$$

Then, we can see that node 1 is again only slightly more important:

$$\begin{aligned}C_1 &= \frac{7}{1+1+1+2+2+2+2} = 0.6364,\\ C_2 &= \frac{7}{1+2+1+1+3+3+3} = 0.5.\end{aligned}$$

Finally, to compute the betweenness centrality, it helps to first write out the quantities of interest. Let $\mathbf{G}$ be the matrix with $G_{ij} = g_{ij}$, $\mathbf{N}^1$ be the matrix with the (i, j) entry being n_{ij}^1 for node 1, and $\mathbf{N}^2$ be the matrix with the (i, j) entry being n_{ij}^2 for node 2. Also let X denote a matrix entry that is not involved in the calculations (a "don't care"). Then, we have

$$\mathbf{G} = \begin{bmatrix} X & 1 & 1 & 1 & 2 & 1 & 1 & 1\\ 1 & X & 1 & 1 & 1 & 1 & 1 & 1\\ 1 & 1 & X & 1 & 2 & 1 & 1 & 1\\ 1 & 1 & 1 & X & 1 & 1 & 1 & 1\\ 2 & 1 & 2 & 1 & X & 2 & 2 & 2\\ 1 & 1 & 1 & 1 & 2 & X & 1 & 2\\ 1 & 1 & 1 & 1 & 2 & 1 & X & 1\\ 1 & 1 & 1 & 1 & 2 & 2 & 1 & X \end{bmatrix},$$

and the following two matrices for nodes 1 and 2, respectively:

$$\mathbf{N}^1 = \begin{bmatrix} X & X & X & X & X & X & X & X \\ X & X & 1 & 0 & 0 & 1 & 1 & 1 \\ X & 1 & X & 1 & 2 & 0 & 0 & 0 \\ X & 0 & 1 & X & 0 & 1 & 1 & 1 \\ X & 0 & 2 & 0 & X & 2 & 2 & 2 \\ X & 1 & 0 & 1 & 2 & X & 0 & 0 \\ X & 1 & 0 & 1 & 2 & 0 & X & 0 \\ X & 1 & 0 & 1 & 2 & 0 & 0 & X \end{bmatrix},$$

$$\mathbf{N}^2 = \begin{bmatrix} X & X & 0 & 0 & 1 & 0 & 0 & 0 \\ X & X & X & X & X & X & X & X \\ 0 & X & X & 0 & 1 & 0 & 0 & 0 \\ 0 & X & 0 & X & 0 & 0 & 0 & 0 \\ 1 & X & 1 & 0 & X & 1 & 1 & 1 \\ 0 & X & 0 & 0 & 1 & X & 0 & 0 \\ 0 & X & 0 & 0 & 1 & 0 & X & 0 \\ 0 & X & 0 & 0 & 1 & 0 & 0 & X \end{bmatrix}.$$

Applying (8.4), we clearly see an intuitive result this time: node 1 is much more important than node 2:

$$B_1 = 12,$$
$$B_2 = 2.5.$$

8.2.2 Measuring link importance

Not all links are equally important. Sometimes there is a natural and operationally meaningful way to assign weights, whether integers or real numbers, to links. For example, the frequency of Alice retweeting Bob's tweets, or of reposting Bob's URL tweets, can be the weight of the link from Bob to Alice. Sometimes there are several categories of link strength. For example, you may have 500 friends on Facebook, but those with whom you have had either one-way or mutual communication might be only 100, and those with whom you have had mutual communication might be only 20. The links to these different types of Facebook friends belong to different strength classes. Weak links might be weak in action, but strong in information exchange.

A link can also be important because it "glues" the network together. For example, if a link's two end points A and B have no common neighbors, or more generally, do not have any other (short) paths of connection, this link is *locally important* in connecting A and B. Links that are important in connecting many node pairs, especially when these nodes are important nodes, are *globally important* in the entire network. These links can be considered *weak*, since they connect nodes that otherwise have no, or very little, overlap. (The opposite is the "triad closure" which we will see in the next chapter.)

But these **weak links** are *strong* precisely for the reason that they open up communication channels across groups that normally do not communicate with each other, as seen in Granovetter's 1973 experiment. One way to quantify this notion is **link betweenness**: this is similar to the betweenness metric defined for nodes, but now we count how many shortest paths a *link* lies on.

Going back to Figure 8.2, we can compute $B_{(i,j)}$, the betweenness of link (i,j) by

$$B_{(i,j)} = \sum_s \sum_{t<s} \frac{n_{st}^{(i,j)}}{g_{st}}, \tag{8.5}$$

where $n_{st}^{(i,j)}$ is the number of shortest paths between two nodes s and t that traverse link (i,j).

Let $\mathbf{N}^{(i,j)}$ be a matrix with the (s,t) entry as $n_{st}^{(i,j)}$. We can compare, for example, the betweenness values of the links $(1,3)$ and $(1,2)$ in Figure 8.2:

$$\mathbf{N}^{(1,3)} = \begin{bmatrix} X & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & X & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & X & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & X & 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 0 & X & 2 & 2 & 2 \\ 1 & 1 & 0 & 1 & 2 & X & 0 & 0 \\ 1 & 1 & 0 & 1 & 2 & 0 & X & 0 \\ 1 & 1 & 0 & 1 & 2 & 0 & 0 & X \end{bmatrix},$$

$$\mathbf{N}^{(1,2)} = \begin{bmatrix} X & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & X & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & X & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & X & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & X & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & X & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & X & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & X \end{bmatrix}.$$

Now we can compute link importance by (8.5). We have the intuition quantified: link (1,3) is much more important than link (1,2) because it glues together the two parts of the graph:

$$B_{(1,3)} = 16,$$
$$B_{(1,2)} = 7.5.$$

8.2.3 Contagion

Now that we have discussed the basic (static) metrics of a graph, we continue with our discussion on influence models with the help of network topology.

Remember the last chapter's section on tipping behavior under the best response strategy? In this two-state model, the initialization has a subset of the

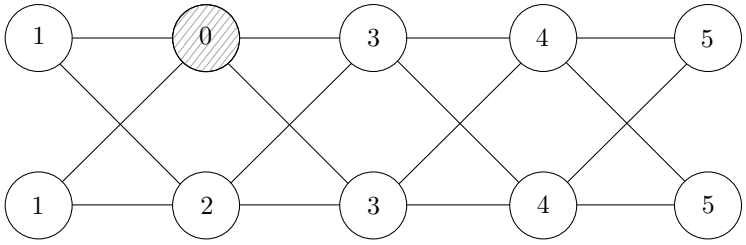

Figure 8.3 An example for the contagion model, with flipping threshold $p = 0.49$. Numbers in nodes indicate the times at which they change from state-0 to state-1: at time 1 the leftmost two nodes flip because one of their two neighbors is in state-1, then this triggers a cascade of flips from left to right.

nodes adopting one state, for example, the state of 1, while the rest of the nodes adopt the other state, the state of 0. We can consider the state-1 nodes as the early adopters of a new product, service, or trend, so it is likely they would be a small minority in the network and not necessarily aggregated together.

Now the question is when, if at all, will all the nodes flip to the new trend, i.e., flip from state-0 to state-1?

Unlike the diffusion model in the last chapter, now it is the *local* population, the set of neighbors of each node, rather than the global population, that matters. One possible local flipping rule is a memoryless, threshold model: if a fraction of p or more of your neighbors have flipped to state-1, you will flip too. For now, let us say all the nodes have the same flipping threshold: the same p for all nodes.

An example is shown in Figure 8.3 with a flipping threshold of $p = 0.49$, and the highlighted node being initialized at state-1. At time 1, the leftmost two nodes flip, and that triggers a cascade of flipping from left to right in the network.

In general, the first question we ask is will the entire network flip? It turns out there is a clear-cut answer: yes, if and only if there is no cluster of density $1 - p$ or higher, in the set of nodes with state-0 at initialization. As will be elaborated in the Advanced Material, a **cluster** of density p is a set of nodes such that each of these nodes has at least a fraction of p of its neighbors also in this set, as illustrated in Figure 8.4.

Without going through the proof of this answer, the "only if" direction of the statement is intuitively clear: a cluster of density $1-p$ or higher, all with state-0, will never see any of its nodes flip since the inertia from within the cluster suffices to avoid flipping. *Homophily* creates a blocking cluster in the network.

The second question is, if the entire network eventually flips, how many iterations will that take? And the third question is, if only some part of the network flips in the end, how big a portion will flip (and where is it in the network)? These two questions are much harder to answer, and depend a lot on the exact network topology.

But perhaps the most useful question to answer for viral marketing is one

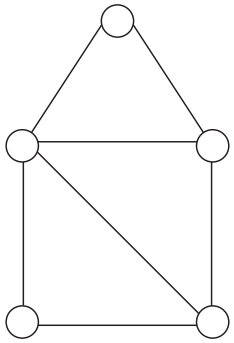

Figure 8.4 A small graph illustrating clusters with various densities. The set of nodes forming the lower left triangle is a cluster with density 1/2, whereas the set of nodes forming the square is a cluster with density 2/3.

of design: suppose each node in a given, known graph can be influenced at the initialization stage: if you pay node i $\$x_i$, it will flip. Presumably those nodes that perceive themselves as more influential will charge a higher price. Under a total budget constraint, which nodes would you seed (pay to change their state and advertise that to neighbors) in order to maximize the extent of flipping at equilibrium, and furthermore, minimize the time it takes to reach that equilibrium?

While this question is hard to answer, some intuitions are clear. If you can seed just one node, it should be the most important one (by some centrality measure). But once you can seed two nodes, it is the *combined* influential power of the pair that matters. For example, you want the two nodes to be close enough to ensure some nodes will flip, but you also want them to be far apart enough from each other that more nodes can be covered. This tradeoff is further influenced by the heterogeneity of flipping thresholds: for the same cost, it is more effective to influence those easier to flip. Network topology is also important: you want to flip them in order to create a cascade so that some nodes can help flip others.

8.2.4 Infection: Population-based model

We have already seen five influence models between the last and this chapter. There is another model that is frequently used in modeling the spread of infectious disease. Unlike the other models, this one has a state transition, between two, three, or even more states that each node may find itself in. We will first describe the interaction using differential equations (over continuous time) and assuming that each node can interact with any other node (which could have been introduced in the last chapter since this model does not depend on the topology). We will then bring in network topology so that a node can directly interact only with its neighbors.

These variants of the infection model differ from each other in terms of the kind

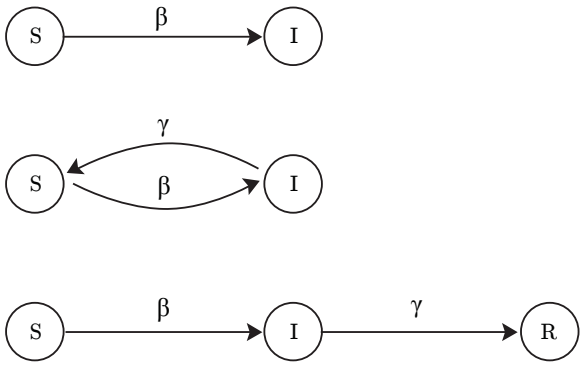

Figure 8.5 Three of the simplest state-transition models for infection: the SI model, SIS model, and SIR model. S stands for susceptible population, I for infected, and R for recovered. Each model can be mathematically described through a system of differential equations, some of which have analytic solutions while others need to be numerically solved. There is a caveat in this graphic representation of the differential equations: the β arrows indicate the transition rate that multiplies the product of the S and I populations, while the γ arrows indicate the transition rate that multiplies just the I population.

of state transitions that are allowed. We will cover only two-state and three-state models. As shown in Figure 8.5, S stands for susceptible, I stands for infected, and R stands for recovered, and the symbols above the state-transition arrows represent the rates of those transitions: how many switch per unit of time. We will use $S(t), I(t)$, and $R(t)$ to represent the *proportions* of the population in that state at time t. The initial conditions at time 0 are denoted as $S(0), I(0)$, and $R(0)$. Since time is continuous here, we use round instead of square brackets around t.

The first model is called the **SI model**, which is very similar to the Bass model for diffusion in the last chapter. It is described by the following pair of differential equations, where the transition rates are proportional to the product $S(t)I(t)$ at each time t:

$$\frac{dS(t)}{dt} = -\beta S(t)I(t), \tag{8.6}$$

$$\frac{dI(t)}{dt} = \beta S(t)I(t). \tag{8.7}$$

We could have also used just one of the two equations above and the normalization equation: all population proportions need to add up to 1: $S(t) + I(t) = 1$ at all times t. Substituting $S(t) = 1 - I(t)$ into (8.6), we have

$$\frac{dI(t)}{dt} = \beta(1 - I(t))I(t) = \beta(I(t) - I^2(t)).$$

This is a simple second-order differential equation just like what we used for the Bass model in Chapter 7. The closed-form solution is indeed a special case

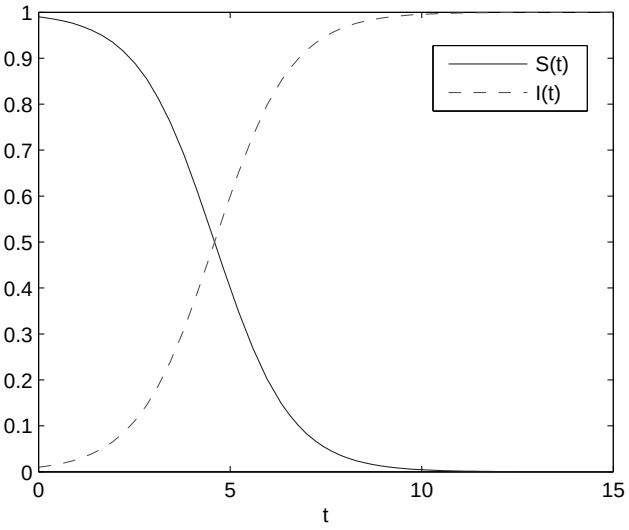

Figure 8.6 Population evolution for the SI model. Eventually everyone is infected.

of the Bass model's solution. It is a sigmoidal curve for the infected population over time, parameterized as a **logistic-growth** equation:

$$I(t) = \frac{I(0)e^{\beta t}}{S(0) + I(0)e^{\beta t}}. \tag{8.8}$$

And, of course, $S(t) = 1 - I(t)$.

We do not go into differential-equation solvers here, but it is easy to *verify*, through differentiation, that the above equation does indeed match the differential equations of the SI model.

When t is small, $I(t)$'s growth is similar to exponential growth. When t becomes large, the ratio in (8.8) approaches 1. An example of the whole curve is shown in Figure 8.6.

The SI model assumes that, once infected, a person stays infected forever. In some diseases, a person can become non-infected but still remain susceptible to further infections. As in Figure 8.5, this **SIS model** is described by the following equations:

$$\frac{dS(t)}{dt} = \gamma I(t) - \beta S(t)I(t),$$
$$\frac{dI(t)}{dt} = \beta S(t)I(t) - \gamma I(t).$$

Without even solving the equations, we can guess that, if $\beta < \gamma$, the infected proportion depletes exponentially. If $\beta > \gamma$, we will see a sigmoidal curve of $I(t)$ going up, but not to 100% since some of the infected will be going back to the susceptible state. The exact saturation percentage of $I(t)$ depends on β/γ.

These intuitions are indeed confirmed in the closed-form solution. Again using

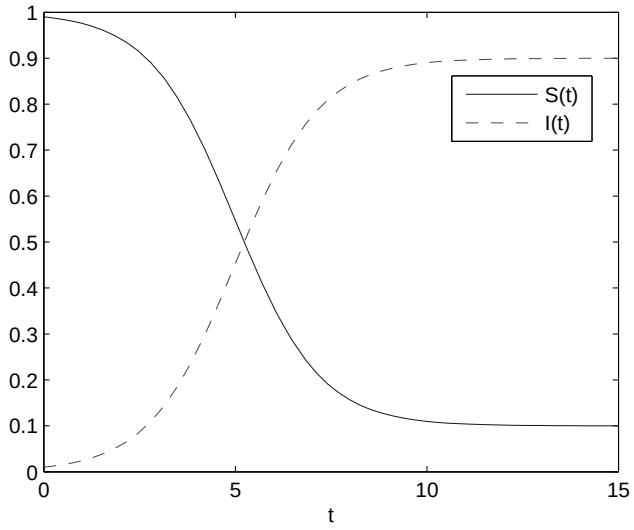

Figure 8.7 Population evolution for the SIS model. At equilibrium, some people are not infected.

$S(t) = 1 - I(t)$ and solving the resulting differential equation in $I(t)$, we have

$$I(t) = (1 - \gamma/\beta)\frac{ce^{(\beta-\gamma)t}}{1 + ce^{(\beta-\gamma)t}}, \tag{8.9}$$

for some constant c that depends on the initial condition. A sample trajectory, where $\beta > \gamma$, is shown in Figure 8.7.

Indeed, the growth pattern depends on whether $\beta > \gamma$ or not, and the $I(t)$ saturation level (as $t \to \infty$) depends on β/γ too. This important constant,

$$\sigma = \beta/\gamma,$$

is called the **basic reproduction number**.

Both the SI and SIS models miss a common feature in many diseases: once infected and then recovered, a person becomes immunized. This is the R state. In the **SIR model** (not to be confused with the Signal-to-Interference Ratio in wireless networks in Chapter 1), one of the most commonly used, simple models for infection, the infected population eventually goes down to 0. As shown in Figure 8.5, the dynamics are described by the following equations:

$$\begin{aligned}
\frac{dS(t)}{dt} &= -\beta S(t)I(t), \\
\frac{dI(t)}{dt} &= \beta S(t)I(t) - \gamma I(t), \\
\frac{dR(t)}{dt} &= \gamma I(t).
\end{aligned}$$

Here, $\sigma = \beta/\gamma$ is the contact rate β (per unit time) times the average infection

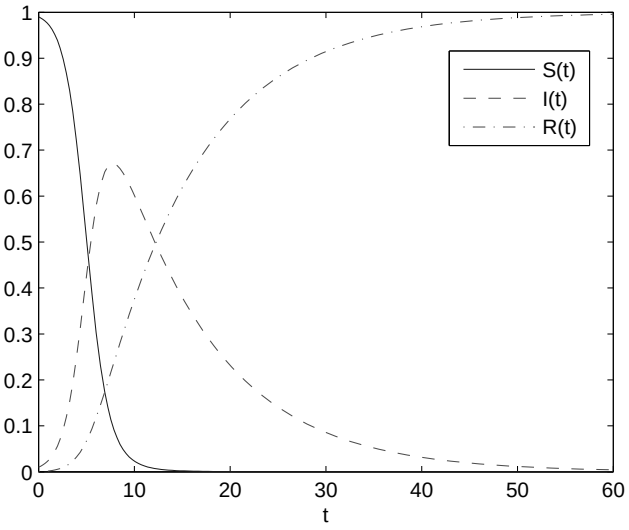

Figure 8.8 Population evolution in the SIR model, for $\sigma S(0) > 1$. Eventually everyone recovers.

period $1/\gamma$. We can run substitution twice to eliminate two of the three equations above, but there is no closed-form solution to the resulting differential equation. Still, we can show that $\sigma S(0)$ plays the role of the threshold level that determines whether $I(t)$ will go up first before coming down. The trajectory of this SIR model has the following properties over a period of time $[0, T]$.

- If $\sigma S(0) \leq 1$, then $I(t)$ decreases to 0 as $t \to \infty$. The initial value $S(0)$ is not large enough to create an epidemic.
- If $\sigma S(0) > 1$, then $I(t)$ increases to a maximum of

$$I_{max} = I(0) + S(0) - 1/\sigma - \log(\sigma S(0))/\sigma,$$

  then decreases to 0 as $t \to \infty$. This is the typical curve of an **epidemic** outbreak.
- $S(t)$ is always a decreasing function, the limit $S(\infty)$ as $t \to \infty$ is the unique root in the range $(0, 1/\sigma)$ of the following equation:

$$I(0) + S(0) - S(\infty) + \frac{1}{\sigma} \log\left(\frac{S(\infty)}{S(0)}\right) = 0.$$

A typical picture of the evolution is shown in Figure 8.8. Eventually, everyone is recovered.

There are many other variants beyond the simplest three above, including the SIRS model where the recovered may become susceptible again, models where new states are introduced, and models where births and deaths (due to infection) are introduced.

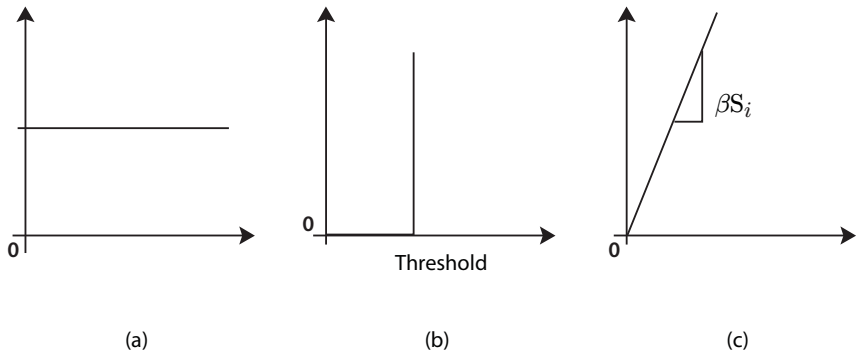

Figure 8.9 Each local node's processing models, the rate of change of node i's state (y-axis) vs. its neighbors' influence (x-axis), for (a) random walk, (b) contagion, and (c) infection. Contagion exhibits a flipping threshold, while infection has a linear dependence of the rate of change on the influence of a neighbor.

8.2.5 Infection: Topology-based model

Up to this point, we have assumed that only the global population matters: each node feels the averaged influence from the entire network. This is sometimes called the *mean-field approximation* approach to enable tractable mathematical analysis.

In reality, of course, infectious diseases spread only between two neighbors in a graph (however you may define the "link" concept for each disease). The difference between the contagion model and the infection model, as well as the random-walk model, now boils down to how we model local processing.

As illustrated in Figure 8.9, in contagion each node makes a deterministic decision to flip or not (depending on whether the local influence is strong enough), whereas in infection each node makes a probabilistic "decision", namely the likelihood of catching the disease, with the rate of change of that likelihood dependent on the amount of local influence. So the discrete state actually turns into a continuous state representing the probability of finding a node in that state.

Intuitively, if the topology is such that there is a bottleneck subset of nodes, it will be harder to spread the disease to the entire network. To model it more precisely, we need to include the adjacency matrix in the update equation.

For example, take the simplest case of the SI model. Now we have for each node i the following differential equations:

$$\frac{dS_i(t)}{dt} = -\beta \sum_j A_{ij}[S_i(t)I_j(t)], \tag{8.10}$$

$$\frac{dI_i(t)}{dt} = \beta \sum_j A_{ij}[S_i(t)I_j(t)]. \tag{8.11}$$

There are two tricky points in this seemingly simple translation from the original population-based model to the topology-based model. First, quantities S_i

and I_i should be read as the *probabilities* of node i being in state S or state I, respectively. Second, it is tempting to pull S_i out of the summation over j since it does not depend on j, but actually we need to read S_iI_j as one quantity: the *joint* probability that node i is in state S and its neighbor node j in state I. So the above notation is actually wrong. But, to estimate this joint probability, we need to know the probability that some neighbor of node j (other than node i) was in state I while node j itself was in state S, for that is the only way we can get to the current state of i in S and j in I. Following this line of reasoning, we have to enumerate all the possible paths of evolution of global states across the *whole network* over time. That is too much computation, and we have to stop at some level and approximate.

The first order of approximation is actually to break the joint probability exactly as in (8.10): the *joint* probability of node i being in state S and node j being in state I is approximated as the *product* of the individual probabilities of node i being in state S and node j being in state I.

For many other network computation problems, from decoding over wireless channels to identifying people by voice recognition, it is common to reduce the computational load by breaking down *global* interactions to *local* interactions. For certain topologies like trees, a low-order approximation can even be exact, or at least accurate enough for practical purposes.

With this first-order approximation, we have the following differential equation for the SI model (which can also be readily extended to the Bass model) with topology taken into account:

$$\frac{dI_i(t)}{dt} = \beta S_i(t) \sum_j A_{ij} I_j(t) \tag{8.12}$$

$$= \beta(1 - I_i(t)) \sum_j A_{ij} I_j(t). \tag{8.13}$$

The presence of the quadratic term and of the adjacency matrix makes it difficult to solve the above equation in closed form. But, during the early times of the infection, $I_i(t)$ is very small, and we can approximate the equation as a linear one by approximating $1 - I_i(t)$ with 1. In vector form, it becomes

$$\frac{d\mathbf{I}(t)}{dt} = \beta \mathbf{A}\mathbf{I}(t). \tag{8.14}$$

Here, $\mathbf{I}(t)$ is not an identity matrix, but a vector of the probabilities of the nodes being in state I at time t. We can, as in (8.1), decompose $\mathbf{I}$ as a weighted sum of eigenvectors $\{\mathbf{v}_k\}$ of $\mathbf{A}$:

$$\mathbf{I}(t) = \sum_k w_k(t)\mathbf{v}_k.$$

The eigenvectors $\{\mathbf{v}_k\}$ are determined by the topology of the graph. The weights $\{w_k(t)\}$ vary over time as the solution to the following linear, *scalar*, differential

equation for each eigenvector k:

$$\frac{dw_k(t)}{dt} = \beta\lambda_k w_k(t),$$

giving rise to the solution

$$w_k(t) = w_k(0)e^{\beta\lambda_k t}.$$

Since $\mathbf{I}(t) = \sum_k w_k(t)\mathbf{v}_k$, the solution to (8.14) is

$$\mathbf{I}(t) = \sum_k w_k(0)e^{\beta\lambda_k t}\mathbf{v}_k.$$

For example, the first two terms of the above sum are

$$w_1(0)e^{\beta\lambda_1 t}\mathbf{v}_1 + w_2(0)e^{\beta\lambda_2 t}\mathbf{v}_2.$$

So the growth is still exponential at the beginning, but the growth exponent is weighted by the eigenvalues $\{\lambda_k\}$ of the adjacency matrix $\mathbf{A}$ now.

There are several other approaches to study infection with topological impact.

- A different approximation is to assume that all nodes of the same degree at the same time have the same S or I value. Like the order-based approximation above, it is clearly incorrect, but useful for generating another tractable way of solving the problem.
- So far we have assumed a detailed topology with an adjacency matrix given. An alternative is to take a generative model of topology that gives rise to features like small-world connectivity, and run infection models on those topologies. We will be studying generative models in the next two chapters.
- We can also randomly pick a link in the given topology to be in an "open" state, with probability p, that a disease will be transmitted from one node to another, and in a "closed" state with $1-p$. Then, from any given initial condition, say, an infected node, there is a set of nodes connected to the original infected node through this maze of open links, and another set of nodes not reachable since they do not lie on the paths consisting of open links. This turns the infection model into the study of **percolation**.

8.3 Examples

8.3.1 Seeding a contagion

Contagion depends on the topology. The graph in Figure 8.10 is obtained by repositioning three links in the graph in Figure 8.3. Even with the same node is initialized as state-1, the number of eventual flips decreases sharply from ten to three. This shows how sensitive contagion outcome is with respect to network topology. We can also check that the density of the set of state-0 nodes is $2/3$ after the leftmost two nodes flip, which is higher than 1 minus the flipping threshold $p = 0.49$, thus preventing a complete flipping of the network.

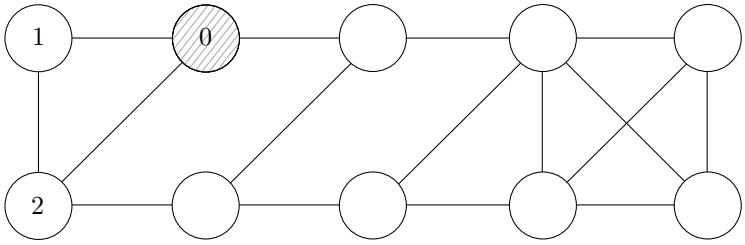

Figure 8.10 An example for contagion model with $p = 0.49$. The shaded node with 0 written in it represents the initial seed at iteration 0. Nodes with numbers written in them are flipped, and the numbers indicate the iteration at which each is flipped. One node is initialized to be at state-1 (flipped), and the final number of flipped nodes is three.

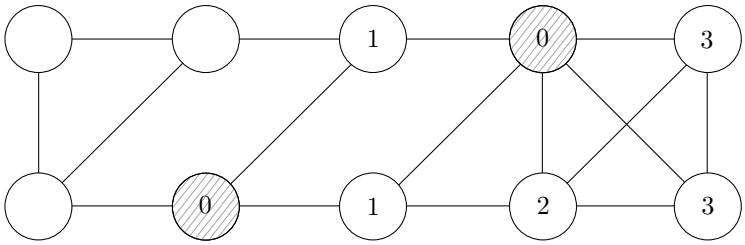

Figure 8.11 An example for contagion model with $p = 0.49$. Two nodes are initialized to be at state-1, and the final number of flipped nodes becomes seven.

Suppose now you want to stage a social media campaign by hiring Twitter, Facebook, and blog writers to spread their influence. Consider the problem of buying off, or seeding, nodes, assuming each node can be seeded at the same price. The aim is to maximize the number of eventual flips. If we can buy off only one node, choosing the node highlighted in Figure 8.10 is actually the optimal strategy. If we can seed two nodes, Figure 8.11 shows the optimal strategy and the final number of flips is seven, a significant improvement.

We also see that the nodes to be picked for seeding in the two-seed example do not include the node we picked to seed in the one-seed example. Optimal seeding strategies cannot be successively refined.

8.3.2 Infection: A case study

Just like the Netflix recommendation algorithm in Chapter 4 and the Bass diffusion model in Chapter 7, we actually do not know the model parameter values until we have some trial or historical data to train the model first.

In the SIR model, we can observe through historical data for each disease the initial population $S(0)$ and final population $S(\infty)$ of susceptible people. Let us assume the initial infected population $I(0)$ is negligible. This means we can

approximate the key parameter σ as

$$\sigma = \frac{\log(S(0)/S(\infty))}{S(0) - S(\infty)}. \tag{8.15}$$

In making public-health decisions, a crucial one is the target vaccination rate for **herd immunity**: we want the immunized population to be large enough that infection does not go up at all, i.e., $S(0) < 1/\sigma$, or,

$$R(0) > 1 - 1/\sigma. \tag{8.16}$$

That means the fraction of the population with immunity, either through catching and recovering from the disease, or through vaccination (the more likely case), must be large enough. Using (8.15) to estimate σ, we can then estimate the vaccination rate $R(0)$ needed. These estimates are not exact, but at least they provide a sense of relative difficulty in controlling different infectious diseases.

Let us take a look at the case of measles. It causes about 1 million deaths worldwide each year, but in developed countries populations are sufficiently vaccinated that it affects very few. Its σ is quite large, having been estimated to be 16.67. By (8.16), this translates into a vaccination rate of 94% needed for herd immunity. But the vaccine's efficacy is not 100%, but more like 95% for measles. So the vaccination rate needs to be 99% to achieve herd immunity. This is a very high target number, and can be achieved only through a two-dose program, which is more expensive than the standard single-dose program.

When measles vaccination was first introduced in 1963 in the USA, the measles infection population dropped but did not disappear: it stayed at around 50,000 a year. In 1978, the US government increased coverage of immunization in an attempt to eliminate measles, and the infection population further dropped to 5,000, but was still not near 0. In fact the number went back up to above 15,000 in 1989–1991. Just increasing the coverage of immunization did not make the immunization rate high enough. In 1989, the US government started using the two-dose program for measles: one vaccination at around 1 year old and another in around 5 years' time. This time the immunization rate went up past the herd-immunity threshold of 99% before children reach school age. Consequently, the number of reported cases of measles dropped to just 86 ten years later.

In a 2011 US movie "Contagion" (we use the term "infection" for spreading of disease), the interactions among three types of networks: social networks, information networks, and disease-spreading networks were depicted. Kate Winslet explained the basic reproduction number too (using the notation R_0, which is equivalent to σ for the cases we mentioned here). Some of the scenes in this drama actually occurred in real life during the last major epidemic, SARS in 2003, e.g., the Chinese government suppressing the news of the disease, some healthcare workers staying on their jobs despite there being a very high basic reproduction number and mortality rate, and the speed of information transmission exceeding that of disease transmission.

8.4 Advanced Material

8.4.1 Random walk

One influence model with network topology has already been introduced in Chapter 3: the PageRank algorithm. In that chapter, we wanted to see what set of numbers, one per node, is *self-consistent* on a directed graph: if each node spreads its number evenly across all its outgoing neighbors, will the resulting numbers be the same? It is a state of equilibrium in the influence model, where the influence is spread across the outgoing neighbors.

In sociology, the **DeGroot model** is similar, except that it starts with a *given* set of numbers $\mathbf{v}$, one per node (so the state of each node is a real number rather than a discrete one), and you want to determine what happens over time under the above influence mechanism.

The evolution of $\mathbf{x}[t]$, over discrete timeslots indexed by t, can be expressed as follows:

$$\mathbf{x}[t] = \mathbf{A}^t \mathbf{v}. \tag{8.17}$$

Here $\mathbf{A}$ is an influence-relationship adjacency matrix. If $A_{ii} = 1$ and $A_{ij} = 0$ for all j, that means node i is an "opinion seed" that is not influenced by any other nodes.

We have seen this linear equation many times by now: power control, PageRank, centrality measures. It is also called random walk on graphs. When does it converge? Will it converge only on a subset of nodes (and the associated links)?

Following standard results in linear algebra, we can show that, for any subset of nodes that is **closed** (there is no link pointing from a node in the subset to a node outside), the necessary and sufficient conditions on matrix $\mathbf{A}$ for convergence are that it is

- *irreducible*: an adjacency matrix being irreducible means that the corresponding graph is connected: there is a directed path from any node to any other node; and
- *aperiodic*: an adjacency matrix being aperiodic means that the lengths of all the cycles in the directed graph have the greatest common denominator of 1.

What about the *rate of convergence*? That is much harder to analytically characterize. But, to first order, an approximation is that the convergence rate is governed by the *ratio* of the second-largest eigenvalue λ_2 and the largest eigenvalue λ_1. An easy way to see this is to continue the development of eigenvector centrality (8.2). The solution to (8.17) can be expressed through the eigenvector decomposition $\mathbf{v} = \sum_i c_i \mathbf{v}_i$:

$$\mathbf{x}[t] = \mathbf{A}^t \sum_i c_i \mathbf{v}_i = \sum_i c_i \lambda_1^t \left(\frac{\lambda_i}{\lambda_1}\right)^t \mathbf{v}_i.$$

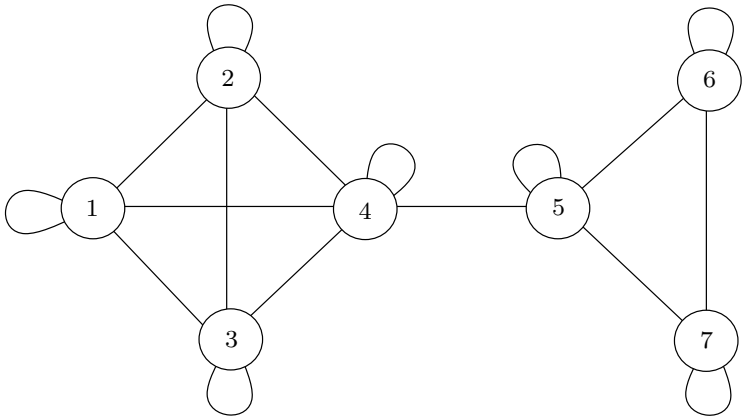

Figure 8.12 Example for the DeGroot model and the contagion model. The initial scores on nodes $1-4$ eventually spread evenly to all nodes. The rate of convergence to this equilibrium is governed by the second-largest eigenvalue of the adjacency matrix.

Dividing both sides by the leading term $c_1\lambda_1^t$, since we want to see how accurate the first-order approximation is, and rearranging the terms, we have

$$\frac{\mathbf{x}[t]}{c_1\lambda_1^t} = \mathbf{v}_1 + \frac{c_2}{c_1}\left(\frac{\lambda_2}{\lambda_1}\right)^t \mathbf{v}_2 + \cdots .$$

This means that the leading term of the *error* between $\mathbf{x}(t)$ and the first-order approximation $c_1\lambda_1^t\mathbf{v}_1$ is the second-order term in the eigenvector expansion, with a magnitude that evolves over time t proportionally to

$$\left(\frac{\lambda_2}{\lambda_1}\right)^t .$$

For certain matrices like the Google matrix $\mathbf{G}$, the largest eigenvalue λ_1 is 1. Then it is the second-largest eigenvalue λ_2 that governs the rate of convergence.

As a small example, consider the network shown in Figure 8.12 consisting of two clusters with a link between them. Suppose the state of each node is a score between 0 and 100, and the initial score vector is

$$\mathbf{v} = \begin{bmatrix}100 & 100 & 100 & 100 & 0 & 0 & 0\end{bmatrix}^T ,$$

i.e., all nodes in the left cluster have an initial score of 100, and the right cluster has an initial score of 0.

From the network we can also write out $\mathbf{A}$, normalized per row as in Google's

PageRank matrices:

$$\mathbf{A} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 & 0 & 0 \\ 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 1/3 & 1/3 & 1/3 \end{bmatrix}.$$

Then we iterate the equation

$$\mathbf{x}[t] = \mathbf{A}\mathbf{x}[t-1]$$

to obtain

$$\begin{aligned} \mathbf{x}[0] &= \begin{bmatrix} 100 & 100 & 100 & 100 & 0 & 0 & 0 \end{bmatrix}^T, \\ \mathbf{x}[1] &= \begin{bmatrix} 100 & 100 & 100 & 80 & 25 & 0 & 0 \end{bmatrix}^T, \\ \mathbf{x}[2] &= \begin{bmatrix} 95 & 95 & 95 & 81 & 26.25 & 8.333 & 8.333 \end{bmatrix}^T, \\ \mathbf{x}[3] &= \begin{bmatrix} 91.5 & 91.5 & 91.5 & 78.45 & 30.98 & 14.31 & 14.31 \end{bmatrix}^T, \\ &\vdots \\ \mathbf{x}[\infty] &= \begin{bmatrix} 62.96 & 62.96 & 62.96 & 62.96 & 62.96 & 62.96 & 62.96 \end{bmatrix}^T. \end{aligned}$$

We see that the network is connected, i.e., $\mathbf{A}$ is irreducible. The existence of self-loops ensures the network is also aperiodic. Convergence to an equilibrium is therefore guaranteed. The scores at equilibrium are biased towards the initial scores of the left cluster because it is the larger cluster.

8.4.2 Measuring subgraph connectedness

We have finished our tour of seven influence models in two chapters. We conclude this cluster of chapters with a discussion of what constitutes a group in a network.

Intuitively, a set of nodes form a group if there are many connections (counting links or paths) among them, but not that many between them and other sets of nodes. Suppose we divide a graph into two parts. Count the number of links between the two parts; call that A. Then, for each part of the graph, count the total number of links with at least one end in that part, and call those B_1 and B_2. A different way to divide the graph may lead to a different ratio

$$\frac{A}{\min(B_1, B_2)}.$$

The smallest possible value of the ratio is called the **conductance** of this graph. It is also used to characterize convergence speed in random walk.

To dig deeper into what constitutes a group, we need some metrics that quantify the notion of connectedness in a **subgraph**: a subset of nodes, and the

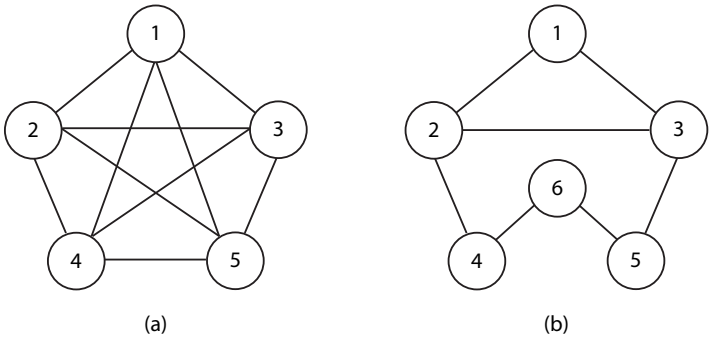

Figure 8.13 Two graphs to illustrate some of the definitions discussed. The entirety of (a) is a 4-component, as any node can reach any other using one of 4 node-disjoint paths. It is also a clique, since all nodes are neighbors. In (b), nodes 1-5 form a 2-clique, whereas either node 4 or 5 must be removed from this set to make a 2-club.

links that start and end with these nodes. First of all, let us focus on end-to-end connectedness, i.e., the existence of paths. We say a subset is a **connected component**, or just a **component**, if each node in the subset can reach any other node in the subset through some path. For directed graphs, the path needs to be directed too, and it is called a **strongly connected component**. In almost all our networks, there is just one component: the entire network itself. We can also further strengthen the notion of component to k**-component**: a maximal subset of nodes in which each node can be connected to any other node through not just one path, but k different paths.

As is typical in this subsection, to make the definitions useful, we refer to the *maximal* subset of nodes: you cannot add another node to the subset and still satisfy the definition.

What if we shift our attention from end-to-end path connectedness to one-hop connectedness? We call a maximal subset of nodes a **clique** if every node in the subset is every other node's neighbor, i.e., there is a link between every pair of nodes in the subset. It is sometimes called a **full mesh**.

A clique is very dense. More likely we will encounter a cluster of density p as defined before in the analysis of the contagion model: a maximal subset of nodes in which each node has at least $p\%$ of its neighbors in this subset.

In-between a component and a clique, there is a middle ground. A k**-clique** is a maximal subset of nodes in which any node can reach any other node through no more than k links. If these k links also are all in-between nodes belonging to this subset, we call the subset a k**-club**.

The two graphs in Figure 8.13 will help illustrate these terms. In graph (a), the star in the middle adds various connections between the nodes. Consider node 1. It has a direct connection to each of the other nodes, making each its neighbor. By symmetry of the graph, all the nodes are neighbors, which means the graph is a clique. Obviously, graph (a) is a component, as each node can reach any other through some path. But we can say more. Again, consider node

1. It can reach node 2 by going directly to it, or by first visiting any of nodes 3, 4, or 5, before going to it. All of these paths are difference in the sense of being node-disjoint, as none of the intermediate nodes are the same. We can say the same for the other nodes. Hence, graph (a) is a 4-component.

Now consider graph (b). Most of the middle links from (a) have been removed, and a node has been added. We can no longer say the graph is a 4-component or a clique. But consider the set of nodes $V = \{1, 2, 3, 4, 5\}$. Any of these nodes can reach any other using no more than *two* links, and therefore the subgraph V is a 2-clique. If we had included node 6, it would have been a 3-clique, as any path between nodes 1 and 6 has at least 3 links. But notice that for node 4 to reach node 5 using two links, it has to use links $(4, 6)$ and $(6, 5)$, which are *not* in V. Hence, V is not a 2-club. But the subgraphs $V\backslash\{4\}$ and $V\backslash\{5\}$ (V without nodes 4 and 5, respectively) are both 2-clubs.

We have so far assumed that geographic proximity in a graph reflects social distance too. But that does not have to be the case. Sometimes, we have a system of labeling nodes by some characteristics, and we want a metric quantifying the notion of **associative mixing** based on this labeling: that nodes which are alike tend to associate with each other.

Consider labeling each node in a given graph as belonging to one of M given types, e.g., M social clubs, M undergraduate majors, or M dorms. We can easily count the number of links connecting nodes of the same type. From the given adjacency matrix $\mathbf{A}$, we have

$$\sum_{ij \in \text{same type}} A_{ij}. \tag{8.18}$$

But this expression is not quite right to use for our purpose. Some nodes have large degrees anyway. So we have to calibrate with respect to that. Consider an undirected graph, and pick node i with degree d_i. Each of its neighbors, indexed by j, has a degree d_j. Let us pick one of node i's links. What is the chance that on the other end of this link is node j? That would be d_j/L, where L is the total number of links in the network. Now multiply by the number of links node i has, and sum over node pairs (ij) of the same type, we have

$$\sum_{ij \in \text{same type}} \frac{d_i d_j}{L}. \tag{8.19}$$

The difference between (8.18) and (8.19), normalized by the number of links L, is the **modularity** Q of a given graph (with respect to a particular labeling of the nodes):

$$Q = \frac{1}{L} \sum_{ij \in \text{same type}} \left(A_{ij} - \frac{d_i d_j}{L} \right). \tag{8.20}$$

Q can be positive, in which case we have *associative* mixing: people of the same type connect more with each other (relative to a random drawing). It can be negative, in which case we have *dissociative* mixing: people of different types

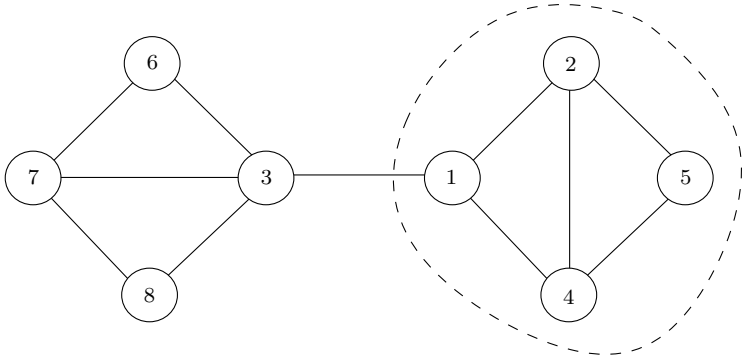

Figure 8.14 An associative labeling of a small graph. Nodes close to each other are labeled into the same type, and the modularity Q is positive.

connect more with each other. With the normalization by L in its definition, we know $Q \in [-1, 1]$.

Still using the same graph in Figure 8.2, we consider a labeling of the nodes into two types. In the following example, nodes enclosed in dashed lines belong to type 1, and the remaining nodes belong to type 2.

Obviously, the degrees are

$$\mathbf{d} = \begin{bmatrix} 3 & 3 & 4 & 3 & 2 & 2 & 3 & 2 \end{bmatrix}^T.$$

We also have

$$L = 11 \times 2 = 22,$$

since the links are undirected. In the computation of modularity, all pairs (ij) (such that $ij \in$ same type) are considered, which means every undirected link is counted twice; thus the normalization constant L should be counted the same way.

Now consider associative mixing with the grouping in Figure 8.14. The modularity can be expressed as

$$\begin{aligned} Q =& \frac{1}{L} \sum_{ij \in \text{same type}} \left(A_{ij} - \frac{d_i d_j}{L} \right) \\ =& \frac{1}{L} \sum_{ij} S_{ij} \left(A_{ij} - \frac{d_i d_j}{L} \right), \end{aligned}$$

where the index S_{ij} denotes whether i and j are of the same type as specified

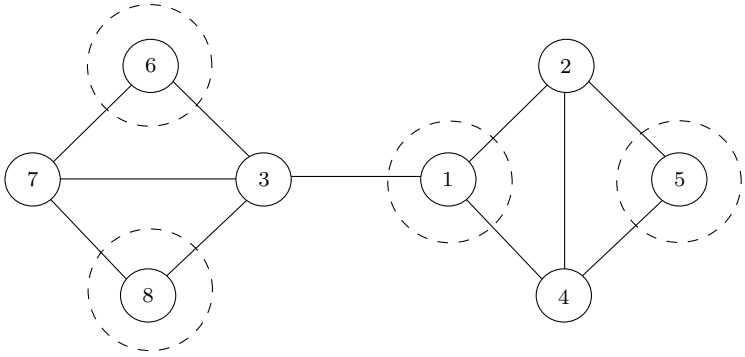

Figure 8.15 A disassociative labeling of a small graph. Nodes close to each other are labeled into different types, and the modularity Q is negative.

by the labeling. We can also collect these indices into a binary matrix:

$$\mathbf{S} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Given $\mathbf{S}, \mathbf{A}$, and $\mathbf{d}$, we can compute Q. We see the modularity value is indeed positive for this labeling system, and reasonably high:

$$Q = 0.5413.$$

But suppose the labeling is changed to what is shown in Figure 8.15, i.e.,

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

Then the modularity value is negative, as we would expect from dissociative mixing of putting nodes of different types next to each other in Figure 8.15.

$$Q = -0.2025.$$

8.4.3 Graph partition and community detection

It is actually not easy to infer either the network topology or the traffic pattern from limited (local, partial, and noisy) measurement. In the last part of this chapter, we assume someone has built an accurate topology already, and our job is to detect the non-overlapping communities, or subgraphs, through some centralized, off-line computation.

The easiest version of the problem statement is that we are given the number of communities and the number of nodes in each community. This is when you have pretty good prior knowledge about the communities in the first place. It is called the **graph-partition** problem. We will focus on the special case in which we partition a given graph into two subgraphs, the **graph-bisection** problem, with a fixed target size (number of nodes) in each subgraph. The input is a graph $G = (V, E)$, and the output is two sets of nodes that add up to the original node set V.

How do we even define that one graph partition is "better" than another? One metric is the number of links between the two subgraphs, called the **cut size**. Later you will see the "max flow min cut" theorem in routing. For now, we just want to find a bisection that minimizes the cut size.

An algorithm that is simple to describe although heavy in computational load is the **Kernighan–Lin algorithm**. There are two loops in the algorithm. In each step of the outer loop indexed by k, we start with a bisection: graphs $G_1[k]$ and $G_2[k]$. To initialize the first outer loop, we put some nodes in subgraph $G_1[1]$ and the rest in the other subgraph $G_2[1]$.

Now we go through an inner loop, where at each step we pick the pair of nodes (i, j), where $i \in G_1$ and $j \in G_2$, such that swapping them reduces the cut size most. If cut size can only be increased, then pick the pair such that the cut size increases by the smallest amount. After each step of the inner loop, the pair that has been swapped can no longer be considered in future swaps. When there are no more pairs to consider, we complete the inner loop and pick the configuration $(G_1^*[k], G_2^*[k])$ (i.e., which nodes belong to which subgraph) with the smallest cut size $c^*[k]$.

Then we take that configuration $(G_1^*[k], G_2^*[k])$ as the initialization of the next step $k+1$ in the outer loop. This continues until the cut size cannot be decreased further through the outer loops. The configuration with the smallest cut size throughout the outer loop,

$$\min_k \{c^*[k]\},$$

is the bisection returned by this algorithm.

More often, we do *not* know how many communities there are, or how many nodes are in each. That is part of the job of **community detection**. For example, you may wonder about the structure of communities in the graph of Facebook connections. And we may be more interested in the richness of connectivity within each subgraph than in the sparsity of connectivity between them.

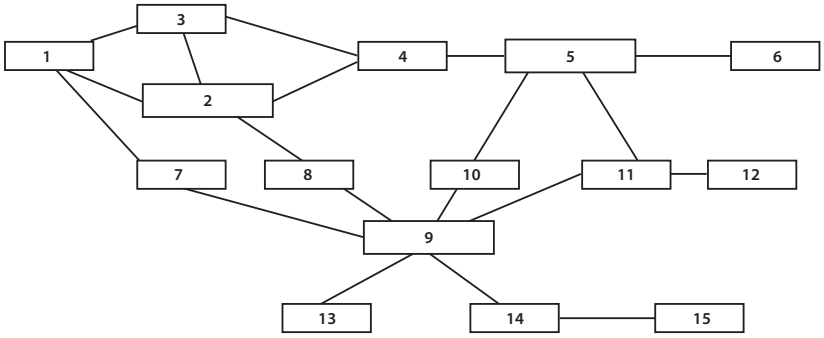

Figure 8.16 The original graph of the Florentian families, viewed as a single community.

Again, we focus on the simpler case of two subgraphs (G_1, G_2), but this time not imposing the number of nodes in each subgraph a priori.

Modularity is an obvious metric to quantify how much more connected a set of nodes is relative to the connectivity if links were randomly established among the nodes. Modularity is defined with respect to a labeling system. Now there are two labels on the nodes, those belonging to G_1 and those to G_2.

So we can simply run the Kernighan–Lin algorithm again. But instead of picking the pair of nodes to *swap* across G_1 and G_2 in order to minimize the cut size, now we select one node to *move* from G_1 to G_2 (or the other way around), in order to maximize the modularity of the graph.

A very different approach gets back to the cut-size-minimization idea, and tries to disconnect the graph into many pieces by *deleting* one link after another. This is useful for detecting not just two communities, but any number of communities. Which link should we delete first? A greedy heuristic computes the betweenness metric of all the links (8.5), and then deletes the link with the highest betweenness value. If that does not break the graph into two subgraphs, then compute the betweenness values of all the remaining links, and delete the link with the highest value again. Eventually, this process will break the graph and give you 2 subgraphs (and $3, 4, \ldots, N$ graphs as you keep deleting links).

As an example, we consider the Renaissance Florentian family graph again, shown in Figure 8.16.

Now we run the following community-detection algorithm.

1. From graph $G = (V, E)$, write down the adjacency matrix $\mathbf{A}$.
2. Compute the betweenness of all links $(i, j) \in E$ from $\mathbf{A}$.
3. Find the link $(i, j)^*$ that has the highest betweenness value. If more than one such link exists, select one of these randomly.
4. Remove link $(i, j)^*$ from G. Check whether any communities have been detected. If not, return to step 1.

Link	Betweenness	Link	Betweenness
(1, 2)	5	(5, 11)	17
(1, 3)	6	(7, 9)	19
(1, 7)	13	(8, 9)	18.17
(2, 3)	4	(9, 10)	15.5
(2, 4)	9.5	(9, 11)	23
(2, 8)	13.5	(9, 13)	14
(3, 4)	8	(9, 14)	**26**
(4, 5)	18.83	(11, 12)	14
(5, 6)	14	(14, 15)	14
(5, 10)	12.5		

Table 8.1 Betweenness values of the links from the initial graph in Figure 8.16. Link (9, 14) has the largest betweenness and will be eliminated first.

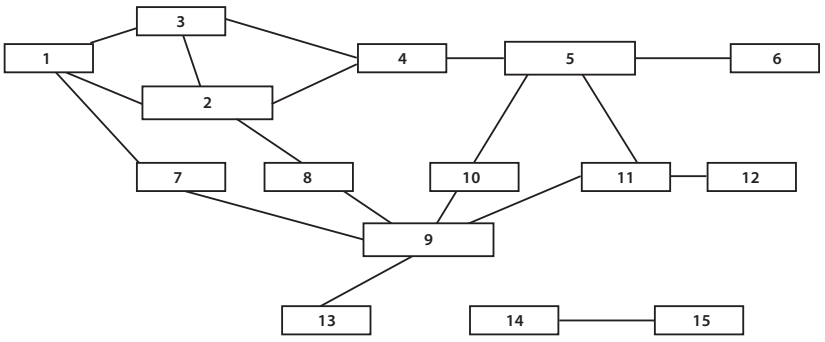

Figure 8.17 Eliminating the link with the highest betweenness, (9, 14), detects two communities within the graph: Node sets $V_1 = \{1, ..., 13\}$ and $V_2 = \{14, 15\}$.

The betweenness values of all the links in the initial graph are shown in Table 8.1. The largest betweenness is that of link (9, 14), with a value of 26. As a result, it is eliminated from the graph. The result is shown in Figure 8.17. As one can see, removing (9, 14) has created two distinct communities: the first is the node set $V_1 = \{1, ..., 13\}$, and the second is the set $V_2 = \{14, 15\}$. If we just want to detect two communities, we can stop now. If we want to detect one more community, we have to keep going.

Next, the adjacency matrix is modified according to the new graph, and the betweenness values are calculated again. The results are shown in Table 8.2.

The largest betweenness is that of link (4, 5): 17.5. As a result, it will be eliminated from the graph. However, removing this link does not detect additional communities, so the process is repeated. Computing betweenness with (4, 5) eliminated, the maximum is that of link (8, 9): 25.5. Again, this link is eliminated. Finally, after running the procedure again, link (7, 9) is found to

Link	Betweenness	Link	Betweenness
(1, 2)	5	(5, 11)	14.33
(1, 3)	5	(7, 9)	14
(1, 7)	10	(8, 9)	12.5
(2, 3)	3	(9, 10)	10.83
(2, 4)	8.83	(9, 11)	16.33
(2, 8)	9.83	(9, 13)	12
(3, 4)	8	(9, 14)	–
(4, 5)	**17.5**	(11, 12)	12
(5, 6)	12	(14, 15)	1
(5, 10)	9.83		

Table 8.2 Betweenness values of the links from the two-component graph in Figure 8.17. Link (4, 5) has the largest betweenness and will be eliminated.

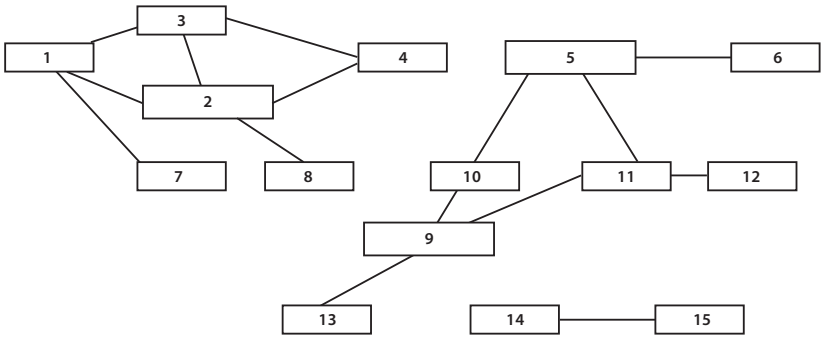

Figure 8.18 Eliminating links (4, 5), (8, 9), and (7, 9) detects three communities: Node sets $V_1 = \{1, 2, 3, 4, 7, 8\}$, $V_2 = \{5, 6, 9, 10, 11, 13\}$, and $V_3 = \{14, 15\}$.

have the highest betweenness value of 42 and is eliminated. This separates the graph into three communities: $V_1 = \{1, 2, 3, 4, 7, 8\}$, $V_2 = \{5, 6, 9, 10, 11, 13\}$, and $V_3 = \{14, 15\}$, as shown in Figure 8.18.

In addition to modularity maximization and betweenness-based link removal, there are several other algorithms for community detection, including graph Laplacian optimization, maximum-likelihood detection, and latent-space modeling. When it comes to a large-scale community-detection problem in practice, it remains unclear which of these will be most helpful in attaining the eventual goal of detecting communities, while remaining robust against measurement noise.

Instead of deleting links, how about *adding* links from a set of disconnected nodes? This way of constructing communities is called **hierarchical clustering**. Now, finding one pair of similar nodes is easy, for example, by using node-similarity metrics like the cosine coefficient in Chapter 4. The difficulty is in defining a consistent and useful notion of similarity between two *sets* of nodes.

If there are N_1 nodes in G_1 and N_2 nodes in G_2, there are then N_1N_2 node pairs. Therefore, there are N_1N_2 similarity-metric values. We need to scalarize this long vector. We can take the largest, the smallest, the average, or any scalar representation of this vector of values as the similarity metric between the two sets of nodes. Once a similarity metric has been fixed and a scalarization method picked, we can hierarchically run clustering by greedily adding nodes, starting from a pair of nodes until there are only two groups of nodes left.

Summary

Box 8.1 Influence models in a network

Contagion creates a complete flip of all the nodes if and only if no cluster of non-flipped nodes is dense enough initially. Infection models the change of states in each node through differential equations. The importance of nodes and links in a graph can be quantified through a variety of centrality measures. Communities in a graph can be detected by deleting links, clustering nodes, or computing how connected a set of nodes are among themselves relative to the other nodes.

Further Reading

Similar to the last chapter, there is a gap between the rich foundation of graph theory and algorithms on the one hand, and the actual operation of Facebook and Twitter and their third-party service providers on the other.

1. The standard reference on contagion models is the following one:
S. Morris, "Contagion," *Review of Economic Studies*, vol. 67, pp. 57–78, 2000.

2. Our discussion of infection models follows the comprehensive survey article:
H. W. Hethcote, "The mathematics of infectious diseases," *SIAM Review*, vol. 42, no. 4, pp. 599–653, October 2000.

3. A classic work on innovation diffusion, both quantitative models and qualitative discussions, can be found in the following book:
E. M. Rogers, *Diffusion of Innovation*, 5th edn., Free Press, 2003.

4. Many graph-theoretic quantities on node importance, link importance, and group connectedness can be found in the following textbook:
M. E. J. Newman, *Networks: An Introduction*, Oxford University Press, 2010.

5. Another standard reference for social network analysis is the following textbook:
S. Wasserman and K. Faust, *Social Network Analysis*, Cambridge University Press, 1994.

Problems

8.1 *Computing centrality and betweenness* ⋆

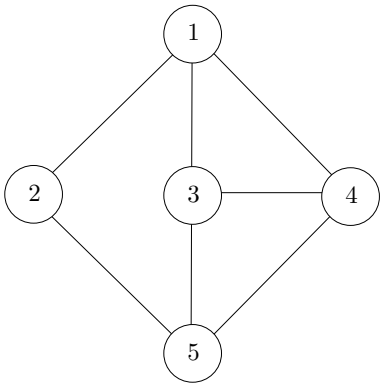

Figure 8.19 A simple graph for computing centrality measures.

(a) Compute the degree, closeness, and eigenvector centrality of each node in the graph in Figure 8.19.

(b) Compute the node betweenness centrality of nodes 2 and 3.

(c) Compute the link betweenness centrality of the links $(3, 4)$ and $(2, 5)$.

8.2 *Contagion* ⋆

Consider the contagion model in the graph in Figure 8.20 with $p = 0.3$.

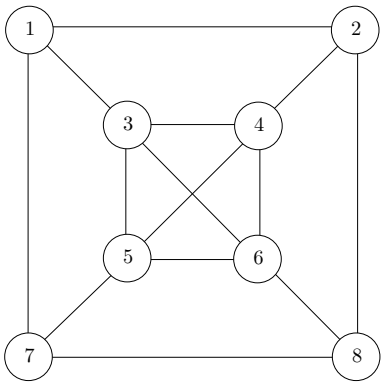

Figure 8.20 A simple graph for studying contagion.

(a) Run the contagion model with node 1 initialized at state-1 and the other nodes initialized at state-0.

(b) Run the contagion model with node 3 initialized at state-1 and the other nodes initialized at state-0.

(c) Contrast the results from (a) and (b) and explain in terms of the cluster densities of the sets of initially state-0 nodes.

8.3 *SIRS infection model* ⋆⋆

We consider an extension to the SIR model that allows nodes in state R to go to state S. This model, known as the SIRS model, accounts for the possibility that a person loses the acquired immunity over time.

Consider the state diagram in Figure 8.21. We can write out the set of differ-

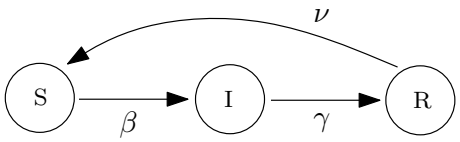

Figure 8.21 The state-transition diagram for the SIRS infection model.

ential equations as

$$\frac{dS(t)}{dt} = -\beta S(t)I(t) + \nu R(t)$$
$$\frac{dI(t)}{dt} = \beta S(t)I(t) - \gamma I(t)$$
$$\frac{dR(t)}{dt} = \gamma I(t) - \nu R(t).$$

Modify the Matlab code `www.network20q.com/hw/simulate_SIR.m` for the numerical solution of the SIR model. Solve for $t = 1, 2, \ldots, 200$ (set the `tspan` vector in code accordingly) with the following parameters and initial conditions: $\beta = 1$, $\gamma = 1/3$, $\nu = 1/50$, $I(0) = 0.1$, $S(0) = 0.9$, and $R(0) = 0$. Describe and explain your observations.

8.4 *Information centrality* ⋆⋆

Consider a weighted, undirected, and connected graph with N nodes, where the weight for link (i, j) is x_{ij}. First construct a matrix $\mathbf{A}$ where the diagonal entries $A_{ii} = 1 + \sum_{ij} x_{ij}$, $A_{ij} = 1 - x_{ij}$ if nodes i and j are adjacent, and $A_{ij} = 1$ otherwise.

Now compute the inverse: $\mathbf{C} = \mathbf{A}^{-1}$. The following quantity is called the **information centrality** of node i:

$$C_I(i) = \frac{1}{C_{ii} + (T - 2R)/N},$$

where $T = \sum_i C_{ii}$ is the trace of matrix $\mathbf{C}$ and $R = \sum_j C_{ij}$ is (any) row sum of matrix $\mathbf{C}$.

Can you think of why this metric is called information centrality?

8.5 *Hypergraphs and bipartite graphs* ⋆⋆

Why must a link be defined as the connection between just *two* nodes? Suppose eight papers are in the fields of physics or chemistry. Group membership, i.e., to which field a papers belongs, is presented as a **hypergraph** in Figure 8.22. Each dotted area is a group or a **hyperedge**, which is a generalization of an undirected edge to connect possibly more than two nodes. Papers 4 and

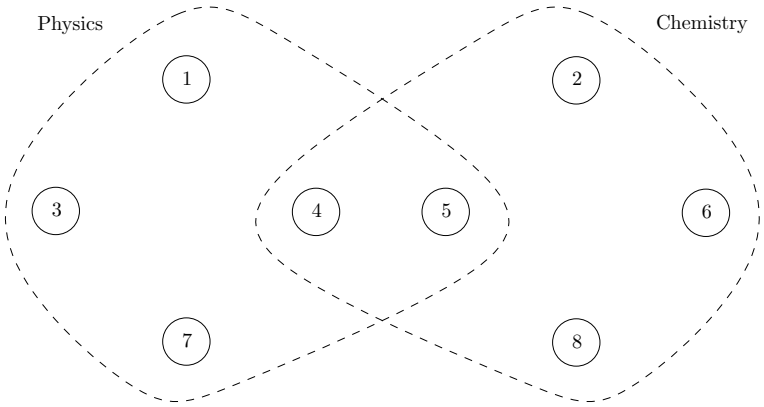

Figure 8.22 Hypergraphs allow each link to connect more than two nodes. Each dotted curve represents a hyperlink.

5 are "interdisciplinary" papers, so their nodes are contained in both hyperedges.

(a) We can transform the hypergraph in Figure 8.22 into an undirected bipartite graph by introducing two more nodes, each representing one of the hyperedges, and linking a "standard" node to a "hyperedge" node if the former is contained in the corresponding hyperedge. Draw this bipartite graph.

(b) Define an *incidence matrix* $\mathbf{B}$ of size 2×8 with

$$B_{ij} = \begin{cases} 1 & \text{node } j \text{ is contained in group } i, \\ 0 & \text{otherwise,} \end{cases}$$

where group 1 is "Physics" and group 2 is "Chemistry." Write down $\mathbf{B}$ for this graph.

(c) Compute the matrix $\mathbf{B}^T\mathbf{B}$. What is its interpretation?

(d) Compute the matrix $\mathbf{B}\mathbf{B}^T$. What is its interpretation?