1 Introduction

Relations are the fundamental fabric of reality.

(Michele Coscia)

1.1 Motivation

Alongside the exponential growth of computer networks over the last few decades, we have witnessed concurrent and equally rapid growth in a field called *network science*. Once computer networks brought network structure into clearer focus, scientists began to recognize networks almost everywhere, even in phenomena that had already received centuries of attention using other methods, and to apply network theory to organize and expand knowledge right throughout the sciences, in every field and discipline.

The set of possible examples is vast, and sources mentioning or treating hundreds of different applications of network methods and graph theory are listed in the reading notes at the end of the chapter. In computer science and machine learning alone, we see computational graphs, graphical networks, neural networks, and deep learning. In operations research, network analysis focuses on minimum cost flow, traveling salesman, shortest path, and assignment problems. In biology, networks are a standard way to represent interactions between bioentities.

In this book, our interest lies in economic and social phenomena. Here, too, networks are pervasive. Important examples include financial networks, production networks, trade networks, transport networks, and social networks. For example, social and information networks affect trends in sentiments and opinions, consumer decisions, and a range of peer effects. The topology of financial networks helps to determine the relative fragility of the financial system, while the structure of production networks affects trade, innovation, and the propagation of local shocks.

Figures 1.1 and 1.2 show two examples of trade networks. Figure 1.1 is called a *Sankey diagram*, which is a kind of figure used to represent flows. Oil flows from left to right. The countries on the left and below are the top 10 exporters of crude oil, while the countries on the right are the top 20 consumers. The figure relates to one of our

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

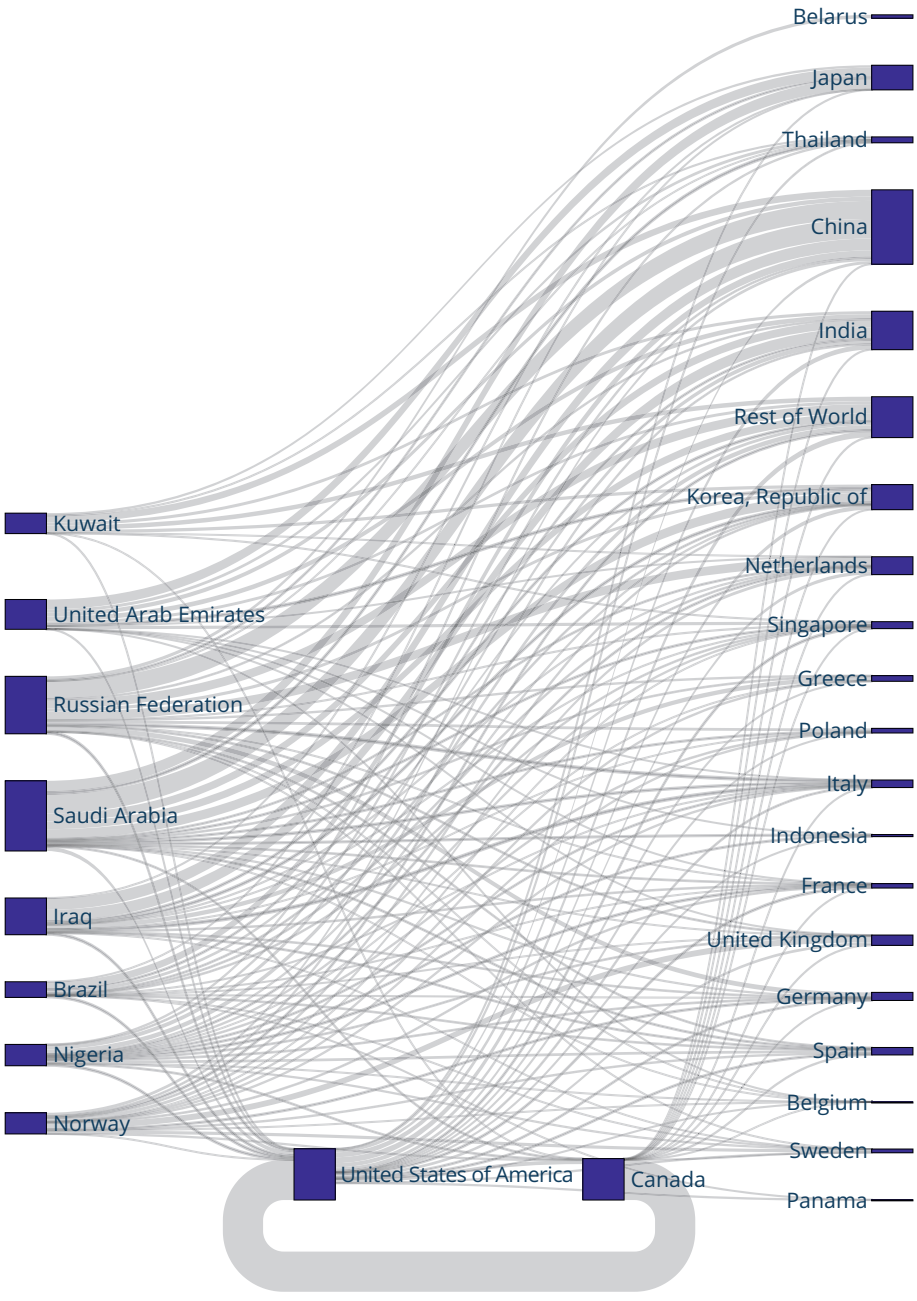

Figure 1.1 International trade in crude oil 2021

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

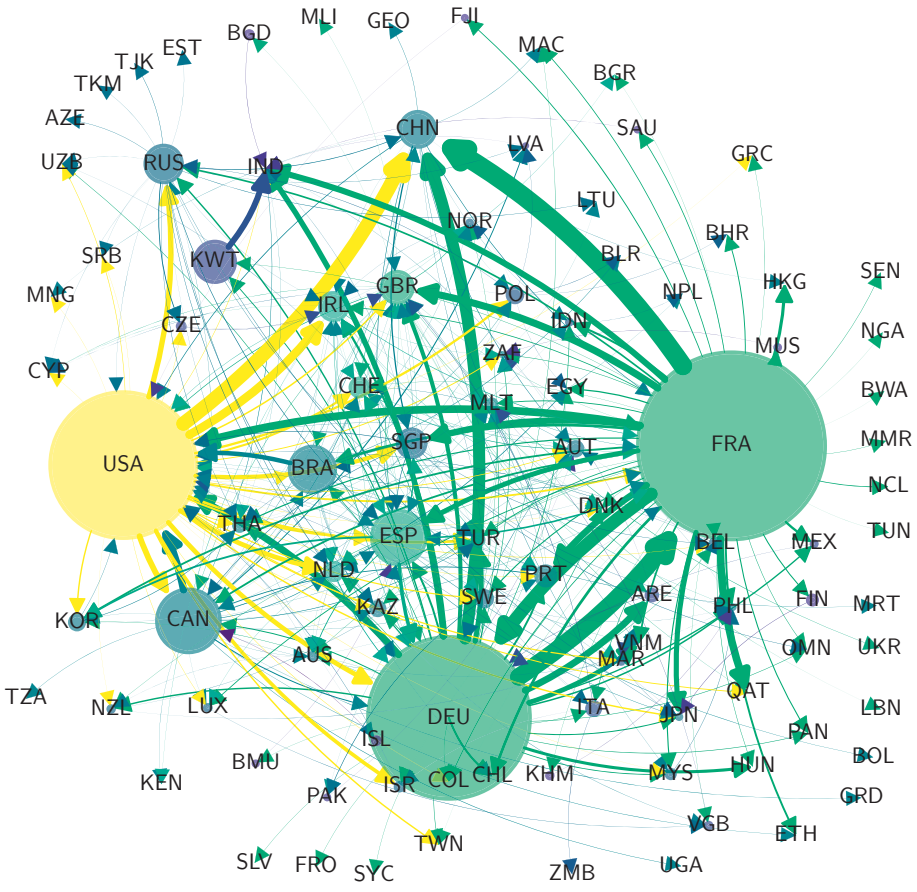

Figure 1.2 International trade in commercial aircraft during 2019

core topics: optimal (and equilibrium) flows across networks. We treat optimal flows at length in Chapter 3.[1]

Figure 1.2 shows international trade in large commercial aircraft in 2019.[2] Node size is proportional to total exports, and link width is proportional to exports to the target country. The USA, France, and Germany are revealed as major export hubs.

While some readers viewing Figures 1.1 and 1.2 might at first suspect that the network perspective adds little more than an attractive technique for visualizing data, it actually adds much more. For example, in Figure 1.2, node colors are based on a ranking of "importance" in the network called *eigenvector centrality*, which we introduce in §1.4.3.4. Such rankings and centrality measures are an active area of

[1] This figure was constructed by QuantEcon research fellow Matthew McKay, using International Trade Data (SITC, Rev 2) collected by The Growth Lab at Harvard University.

[2] This figure was also constructed by Matthew McKay, using data 2019 International Trade Data SITC Revision 2, code 7924. The data pertain to trade in commercial aircraft weighing at least 15,000 kg. It was sourced from CID Dataverse.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

research among network scientists. Eigenvector and other forms of centrality feature throughout the text. For example, we will see that these concepts are closely connected to – and shed new light on – fundamental ideas first developed many years ago by researchers in the field of input–output economics.

In addition, in production networks, it turns out that the nature of shock propagation is heavily dependent on the underlying structure of the network. For example, for a few highly connected nodes, shocks occurring within one firm or sector can have an outsized influence on aggregate-level fluctuations. Economists are currently racing to understand these relationships, their interactions with various centrality measures, and other closely related phenomena.

To understand this line of work, as well as other applications of network methods to economics and finance, some technical foundations are required. For example, to define eigenvector centrality, we need to be familiar with eigenvectors, spectral decompositions, and the Perron–Frobenius theorem. To work with *Katz centrality*, which also features regularly in network science and economics, we require a sound understanding of the Neumann series lemma. The Perron–Frobenius theorem and the Neumann series lemma form much of the technical foundation of this textbook. We review them in detail in §1.2 and develop extensions throughout the remaining chapters.

One reason that analysis of networks is challenging is high dimensionality. To see why, consider implementing a model with n economic agents. This requires n times more data than one representative agent in a setting where agents are atomistic or coordinated by a fixed number of prices. For example, Carvalho and Grassi (2019) model the dynamics of $n = 6 \times 10^6$ firms, all of which need to be tracked when running a simulation. However, if we wish to model interactions between each pair i, j (supply linkages, liabilities, etc.), then, absent sparsity conditions, the data processing requirement grows like $O(n^2)$.[3] In the Carvalho and Grassi (2019) example, n^2 is 3.6×10^{13}, which is very large even for modern computers. One lesson is that network models can be hard to solve, even with powerful computers, unless we think carefully about algorithms.

In general, to obtain a good grasp on the workings of economic networks, we will need computational skills plus a firm understanding of linear algebra, probability, and a field of discrete mathematics called graph theory. The rest of this chapter provides relevant background in these topics. Before tackling this background, we recommend that readers skim the list of common symbols on page xix, as well the mathematical topics in the appendix, which starts on page 180. (The appendix is not intended for sequential reading but rather as a source of definitions and fundamental results to be drawn on in what follows.)

[3] See §A.3 for a discussion of big O notation.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

1.2 Spectral Theory

In this section we review some linear algebra needed for the study of graphs and networks. Highlights include the spectral decomposition of diagonalizable matrices, the Neumann series lemma, and the theorem of Perron and Frobenius.

1.2.1 Eigendecompositions

Our first task is to cover spectral decompositions and the spectral theorem. We begin with a brief review of eigenvalues and their properties. (If you are not familiar with eigenvalues and eigenvectors, please consult an elementary treatment first. See, for example, Cohen (2021).)

1.2.1.1 Eigenvalues

Fix A in $\mathbb{M}^{n\times n}$. A scalar $\lambda \in \mathbb{C}$ is called an **eigenvalue** of A if there exists a nonzero complex vector $e \in \mathbb{C}^n$ such that $Ae = \lambda e$. A vector e satisfying this equality is called an **eigenvector** corresponding to the eigenvalue λ. (Notice that eigenvalues and eigenvectors are allowed to be complex, even though we restrict elements of A to be real.) The set of all eigenvalues of A is called the **spectrum** of A and written as $\sigma(A)$. As we show below, A has at most n distinct eigenvalues.

In Julia, we can check for the eigenvalues of a given square matrix A via `eigvals(A)`. Here is one example

```
using LinearAlgebra
A = [0 -1;
     1  0]
eigenvals = eigvals(A)
```

Running this code in a Jupyter cell (with Julia kernel) produces

```
2-element Vector{ComplexF64}:
 0.0 - 1.0im
 0.0 + 1.0im
```

Here `im` stands for i, the imaginary unit (i.e., $i^2 = -1$).

Exercise 1.2.1. Using pencil and paper, confirm that Julia's output is correct. In particular, show that

$$A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad \Longrightarrow \quad \sigma(A) = \{i, -i\},$$

with corresponding eigenvectors $(-1, i)^\top$ and $(-1, -i)^\top$.

If $\lambda \in \sigma(A)$ and e is an eigenvector for λ, then (λ, e) is called an **eigenpair**.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

EXERCISE 1.2.2. Prove: if (λ, e) is an eigenpair of A and α is a nonzero scalar, then $(\lambda, \alpha e)$ is also an eigenpair of A.

LEMMA 1.2.1 *$\lambda \in \mathbb{C}$ is an eigenvalue of A if and only if* $\det(A - \lambda I) = 0$.

Proof If $\lambda \in \mathbb{R}$, then Lemma 1.2.1 follows directly from Theorem A.4.8 on page 201, since $\det(A - \lambda I) = 0$ is equivalent to the existence of a nonzero vector e such that $(A - \lambda I)e = 0$, which in turn says that λ is an eigenvalue of A. The same arguments extend to the case $\lambda \in \mathbb{C}$ because the statements in Theorem A.4.8 are also valid for complex-valued matrices (see, e.g., Jänich (1994)). □

It can be shown that $p(\lambda) := \det(A - \lambda I)$ is a polynomial of degree n.[4] This polynomial is called the **characteristic polynomial** of A. By the fundamental theorem of algebra, there are n roots (i.e., solutions in $\mathbb{C}$ to the equation $p(\lambda) = 0$), although some may be repeated as in the complete factorization of $p(\lambda)$. By Lemma 1.2.1,

(i) each of these roots is an eigenvalue, and
(ii) no other eigenvalues exist besides these n roots.

If $\lambda \in \sigma(A)$ appears k times in the factorization of the polynomial $p(\lambda)$, then λ is said to have **algebraic multiplicity** k. An eigenvalue with algebraic multiplicity one is called **simple**. A simple eigenvalue λ has the property that its eigenvector is unique up to a scalar multiple, in the sense of Exercise 1.2.2. In other words, the linear span of $\{e \in \mathbb{C}^n : (\lambda, e) \text{ is an eigenpair}\}$ (called the **eigenspace** of λ) is one-dimensional.

EXERCISE 1.2.3. For $A \in \mathbb{M}^{n \times n}$, show that $\lambda \in \sigma(A)$ iff $\tau\lambda \in \sigma(\tau A)$ for all $\tau > 0$.

EXERCISE 1.2.4. A useful fact concerning eigenvectors is that if the characteristic polynomial $p(\lambda) := \det(A - \lambda I)$ has n distinct roots, then the n corresponding eigenvectors form a basis of $\mathbb{C}^n$. Prove this for the case where all eigenvectors are real – that is, show that the n (real) eigenvectors form a basis of $\mathbb{R}^n$. (Bases are defined in §A.4.2. Proving this for $n = 2$ is also a good exercise.)

1.2.1.2 The Eigendecomposition

What are the easiest matrices to work with? An obvious answer to this question is: diagonal matrices. For example, when $D = \text{diag}(\lambda_i)$ with $i \in [n]$,

- the linear system $Dx = b$ reduces to n completely independent scalar equations;
- the t-th power D^t is just $\text{diag}(\lambda_i^t)$; and
- the inverse D^{-1} is just $\text{diag}(\lambda_i^{-1})$, assuming all λ_i's are nonzero.

While most matrices are not diagonal, there is a way that "almost any" matrix can be viewed as a diagonal matrix, after translation of the usual coordinates in $\mathbb{R}^n$ via an alternative basis. This can be extremely useful. The key ideas are described below.

[4] See, for example, Jänich (1994), chapter 6.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

$A \in \mathbb{M}^{n \times n}$ is called **diagonalizable** if

$$A = PDP^{-1} \text{ for some } D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n) \text{ and nonsingular matrix } P.$$

We allow both D and P to contain complex values. The representation PDP^{-1} is called the **eigendecomposition** or the **spectral decomposition** of A.

One way to think about diagonalization is in terms of maps, as in

$$\begin{array}{ccc} \mathbb{R}^n & \xrightarrow{A} & \mathbb{R}^n \\ {\scriptstyle P^{-1}}\downarrow & & \uparrow{\scriptstyle P} \\ \mathbb{C}^n & \xrightarrow{D} & \mathbb{C}^n \end{array}$$

Either we can map directly with A or, alternatively, we can shift to $\mathbb{C}^n$ via P^{-1}, apply the diagonal matrix D, and then shift back to $\mathbb{R}^n$ via P.

The equality $A = PDP^{-1}$ can also be written as $AP = PD$. Decomposed across column vectors, this equation says that each column of P is an eigenvector of A, and each element along the principal diagonal of D is an eigenvalue.

EXERCISE 1.2.5. Confirm this. Why are column vectors taken from P nonzero, as required by the definition of eigenvalues?

EXERCISE 1.2.6. The trace of a matrix is equal to the sum of its eigenvalues, and the determinant is their product. Prove this fact in the case where A is diagonalizable.

EXERCISE 1.2.7. The asymptotic properties of the map $m \mapsto A^m$ are determined by the eigenvalues of A. This is clearest in the diagonalizable case, where $A = P \operatorname{diag}(\lambda_i) P^{-1}$. To illustrate, use induction to show that

$$A = P \operatorname{diag}(\lambda_i) P^{-1} \implies A^m = P \operatorname{diag}(\lambda_i^m) P^{-1} \text{ for all } m \in \mathbb{N}. \tag{1.1}$$

When does diagonalizability hold?

While diagonalizability is not universal, the set of matrices in $\mathbb{M}^{n \times n}$ that fail to be diagonalizable has "Lebesgue measure zero" in $\mathbb{M}^{n \times n}$. (Loosely speaking, only special or carefully constructed examples will fail to be diagonalizable.) The next results provide conditions for the property.

THEOREM 1.2.2 *A matrix $A \in \mathbb{M}^{n \times n}$ is diagonalizable if and only if its eigenvectors form a basis of $\mathbb{C}^n$.*

This result is intuitive: for $A = PDP^{-1}$ to hold, we need P to be invertible, which requires that its n columns are linearly independent. Since $\mathbb{C}^n$ is n-dimensional, this means that the columns form a basis of $\mathbb{C}^n$.

COROLLARY 1.2.3 *If $A \in \mathbb{M}^{n \times n}$ has n distinct eigenvalues, then A is diagonalizable.*

Proof See Exercise 1.2.4. □

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

EXERCISE 1.2.8. Give a counterexample to the statement that the condition in Corollary 1.2.3 is necessary as well as sufficient.

There is another way that we can establish diagonalizability, based on symmetry. Symmetry also lends the diagonalization certain properties that turn out to be very useful in applications. We are referring to the following celebrated theorem.

THEOREM 1.2.4 (Spectral theorem) *If $A \in \mathbb{M}^{n\times n}$ is symmetric, then there exists a real orthonormal $n \times n$ matrix U such that*

$$A = UDU^\top \quad \textit{with} \quad \lambda_i \in \mathbb{R}_+ \textit{ for all } i, \textit{ where } D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n).$$

Since, for the orthonormal matrix U, we have $U^\top = U^{-1}$ (see Lemma A.4.9), one consequence of the spectral theorem is that A is diagonalizable. For obvious reasons, we often say that A is **orthogonally diagonalizable**.

1.2.1.3 Worker Dynamics

Let's study a small application of eigendecomposition. Suppose that, each month, workers are hired at rate α and fired at rate β. Their two states are unemployment (state 1) and employment (state 2). Figure 1.3 shows the transition probabilities for a given worker in each of these two states.

We translate these dynamics into the matrix

$$P_w = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}, \quad \text{where} \quad 0 \leqslant \alpha, \beta \leqslant 1. \tag{1.2}$$

- Row 1 of P_w gives probabilities for unemployment and employment, respectively, when currently unemployed.
- Row 2 of P_w gives probabilities for unemployment and employment, respectively, when currently employed.

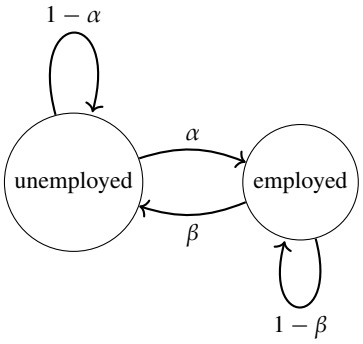

Figure 1.3 Worker transition dynamics

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

Exercise 1.2.9. Using Lemma 1.2.1, show that the two eigenvalues of P_w are $\lambda_1 := 1$ and $\lambda_2 := 1 - \alpha - \beta$. Show that, when $\min\{\alpha, \beta\} > 0$,

$$e_1 := \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{and} \quad e_2 := \begin{pmatrix} -\alpha \\ \beta \end{pmatrix}$$

are two corresponding eigenvectors and that λ_1 and λ_2 are simple.

Exercise 1.2.10. Show that, when $\alpha = \beta = 0$, the eigenvalue λ_1 is not simple.

Below we demonstrate that the m-th power of P_w provides m-step transition probabilities for workers. Anticipating this discussion, we now seek an expression for P_w^m at arbitrary $m \in \mathbb{N}$. This problem is simplified if we use diagonalization.

Exercise 1.2.11. Assume that $\min\{\alpha, \beta\} > 0$. (When $\alpha = \beta = 0$, computing the powers of P_w is trivial.) Show that

$$P_w = EDE^{-1} \quad \text{when} \quad D = \begin{pmatrix} 1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \quad \text{and} \quad E = \begin{pmatrix} 1 & -\alpha \\ 1 & \beta \end{pmatrix}.$$

Using (1.1), prove that

$$P_w^m = \frac{1}{\alpha + \beta} \begin{pmatrix} \beta + \alpha(1 - \alpha - \beta)^m & \alpha(1 - (1 - \alpha - \beta)^m) \\ \beta(1 - (1 - \alpha - \beta)^m) & \alpha + \beta(1 - \alpha - \beta)^m \end{pmatrix} \tag{1.3}$$

for every $m \in \mathbb{N}$.

1.2.1.4 Left Eigenvectors

A vector $\varepsilon \in \mathbb{C}^n$ is called a **left eigenvector** of $A \in \mathbb{M}^{n \times n}$ if ε is an eigenvector of $A^\top$. In other words, ε is nonzero, and there exists a $\lambda \in \mathbb{C}$ such that $A^\top \varepsilon = \lambda \varepsilon$. We can alternatively write the expression as $\varepsilon^\top A = \lambda \varepsilon^\top$, which is where the name "left" eigenvector originates.

Left eigenvectors will play important roles in what follows, including that of stochastic steady states for dynamic models under a Markov assumption. To help distinguish between ordinary and left eigenvectors, we will at times call (ordinary) eigenvectors of A **right eigenvectors** of A.

If A is diagonalizable, then so is $A^\top$. To show this, let $A = PDP^{-1}$ with $D = \operatorname{diag}(\lambda_i)$. We know from earlier discussion that the columns of P are the (right) eigenvectors of A.

Exercise 1.2.12. Let $Q = (P^\top)^{-1}$. Prove that $Q^\top P = I$ and $A^\top = QDQ^{-1}$.

The results of the last exercise show that, when $A = PDP^{-1}$, the columns of $(P^\top)^{-1}$ coincide with the left eigenvectors of A. (Why?) Equivalently, $A = PDQ^\top$, where $Q = (\varepsilon_1, \ldots, \varepsilon_n)$ is the $n \times n$ matrix with i-th column equal to the i-th left eigenvector of A.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

Exercise 1.2.13. Let $(e_i)_{i=1}^n$ be right eigenvectors of A, and let $(\varepsilon_i)_{i=1}^n$ be the left eigenvectors. Prove that

$$\langle \varepsilon_i, e_j \rangle = \mathbb{1}\{i = j\} \qquad (i, j \in [n]). \tag{1.4}$$

[Hint: Use the results of Exercise 1.2.12.]

Exercise 1.2.14. Continuing with the notation defined above and continuing to assume that A is diagonalizable, prove that

$$A = \sum_{i=1}^n \lambda_i e_i \varepsilon_i^\top \quad \text{and} \quad A^m = \sum_{i=1}^n \lambda_i^m e_i \varepsilon_i^\top \tag{1.5}$$

for all $m \in \mathbb{N}$. The expression for A on the left-hand side of (1.5) is called the **spectral representation** of A.

Exercise 1.2.15. Prove that each $n \times n$ matrix $\lambda_i e_i \varepsilon_i^\top$ in the sum $\sum_{i=1}^n \lambda_i e_i \varepsilon_i^\top$ is rank 1.

1.2.1.5 Similar Matrices

Diagonalizability is a special case of a more general concept: $A \in \mathbb{M}^{n\times n}$ is called **similar** to $B \in \mathbb{M}^{n\times n}$ if there exists an invertible matrix P such that $A = PBP^{-1}$. In this terminology, A is diagonalizable if and only if it is similar to a diagonal matrix.

Exercise 1.2.16. Prove that similarity between matrices is an equivalence relation (see §A.1.2) on $\mathbb{M}^{n\times n}$.

Exercise 1.2.17. The fact that similarity is an equivalence relation on $\mathbb{M}^{n\times n}$ implies that this relation partitions $\mathbb{M}^{n\times n}$ into disjoint equivalence classes, elements of which are all similar. Prove that all matrices in each equivalence class share the same eigenvalues.

Exercise 1.2.18. Prove: If A is similar to B, then A^m is similar to B^m. In particular

$$A = PBP^{-1} \implies A^m = PB^mP^{-1} \text{ for all } m \in \mathbb{N}.$$

The last result is a generalization of (1.1). When A is large, calculating the powers A^k can be computationally expensive or infeasible. If, however, A is similar to some simpler matrix B, then we can take powers of B instead, and then transition back to A using the similarity relation.[5]

[5] The only concern with this shift process is that P can be ill-conditioned, implying that the inverse is numerically unstable.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

1.2.2 The Neumann Series Lemma

Most high school students learn that, if a is a number with $|a| < 1$, then

$$\sum_{i \geqslant 0} a^i = \frac{1}{1-a}. \tag{1.6}$$

This geometric series representation extends to matrices: If A is a matrix satisfying a certain condition, then (1.6) holds, in the sense that $\sum_{i \geqslant 0} A^i = (I - A)^{-1}$. (Here I is the identity matrix.) But what is the "certain condition" that we need to place on A, which generalizes the concept $|a| < 1$ to matrices? The answer to this question involves the "spectral radius" of a matrix, which we now describe.

1.2.2.1 Spectral Radius

Fix $A \in \mathbb{M}^{n \times n}$. With $|z|$ indicating the modulus of a complex number z, the **spectral radius** of A is defined as

$$r(A) := \max\{|\lambda| : \lambda \text{ is an eigenvalue of } A\}. \tag{1.7}$$

Within economics, the spectral radius has important applications in dynamics, asset pricing, and numerous other fields. As we will see, the same concept also plays a key role in network analysis.

REMARK 1.2.1 *For any square matrix A, we have $r(A^\top) = r(A)$. This follows from the fact that A and $A^\top$ always have the same eigenvalues.*

Example 1.2.1: As usual, diagonal matrices supply the simplest example: If $D = \operatorname{diag}(d_i)$, then the spectrum $\sigma(D)$ is just $\{d_i\}_{i \in [n]}$, and hence $r(D) = \max_i |d_i|$.

After executing

```
import numpy as np
```

The following Python code computes the spectral radius of a square matrix M:

```
def spec_rad(M):
    return np.max(np.abs(np.linalg.eigvals(M)))
```

1.2.2.2 Geometric Series

We can now return to the matrix extension of (1.6) and state a formal result.

THEOREM 1.2.5 (Neumann series lemma (NSL)) *If A is in $\mathbb{M}^{n \times n}$ and $r(A) < 1$, then $I - A$ is nonsingular and*

$$(I - A)^{-1} = \sum_{m=0}^{\infty} A^m. \tag{1.8}$$

The sum $\sum_{m=0}^{\infty} A^m$ is called the **power series** representation of $(I - A)^{-1}$. Convergence of the matrix series is understood as element-by-element convergence.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

A full proof of Theorem 1.2.5 can be found in Cheney (2013) and many other sources. The core idea is simple: if $S = I + A + A^2 + \cdots$ then $I + AS = S$. Reorganizing gives $(I - A)S = I$, which is equivalent to (1.8). The main technical issue is showing that the power series converges. The full proof shows that this always holds when $r(A) < 1$.

EXERCISE 1.2.19. Fix $A \in \mathbb{M}^{n\times n}$. Prove the following: if $r(A) < 1$, then, for each $b \in \mathbb{R}^n$, the linear system $x = Ax + b$ has the unique solution $x^* \in \mathbb{R}^n$ given by

$$x^* = \sum_{m=0}^{\infty} A^m b. \tag{1.9}$$

1.2.3 The Perron–Frobenius Theorem

In this section we state and discuss a suprisingly far-reaching theorem due to Oskar Perron and Ferdinand Frobenius, which has applications in network theory, machine learning, asset pricing, Markov dynamics, nonlinear dynamics, input–output analysis, and many other fields. In essence, the theorem provides additional information about eigenvalues and eigenvectors when the matrix in question is positive in some sense.

1.2.3.1 Order in Matrix Space

We require some definitions. In what follows, for $A \in \mathbb{M}^{n\times k}$, we write

- $A \geqslant 0$ if all elements of A are nonnegative and
- $A \gg 0$ if all elements of A are strictly positive.

It's easy to imagine how nonnegativity and positivity are important notions for matrices, just as they are for numbers. However, strict positivity of every element of a matrix is hard to satisfy, especially for a large matrix. As a result, mathematicians routinely use two notions of "predominantly strictly positive," which sometimes provide sufficient positivity for the theorems that we need.

Regarding these two notions, for $A \in \mathbb{M}^{n\times n}$, we say that $A \geqslant 0$ is

- **irreducible** if $\sum_{m=0}^{\infty} A^m \gg 0$ and
- **primitive** if there exists an $m \in \mathbb{N}$ such that $A^m \gg 0$.

Evidently, for $A \in \mathbb{M}^{n\times n}$ we have

$$A \gg 0 \implies A \text{ primitive} \implies A \text{ irreducible} \implies A \geqslant 0.$$

A nonnegative matrix is called **reducible** if it fails to be irreducible.

EXERCISE 1.2.20. By examining the expression for P_w^m in (1.3), show that P_w is

(i) irreducible if and only if $0 < \alpha, \beta \leqslant 1$; and
(ii) primitive if and only if $0 < \alpha, \beta \leqslant 1$ and $\min\{\alpha, \beta\} < 1$.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

In addition to the above notation, for $A, B \in \mathbb{M}^{n\times k}$, we also write

- $A \geqslant B$ if $A - B \geqslant 0$ and $A \gg B$ if $A - B \gg 0$,
- $A \leqslant 0$ if $-A \geqslant 0$, etc.

Exercise 1.2.21. Show that $\leqslant$ is a partial order (see §A.2.1) on $\mathbb{M}^{n\times k}$.

The partial order $\leqslant$ discussed in Exercise 1.2.21 is usually called the **pointwise partial order** on $\mathbb{M}^{n\times k}$. Analogous notation and terminology are used for vectors.

The following exercise shows that nonnegative matrices are order-preserving maps (see §A.2.3) on vector space under the pointwise partial order – a fact we shall exploit many times.

Exercise 1.2.22. Show that the map $x \mapsto Ax$ is order-preserving (see §A.2.3) whenever $A \geqslant 0$ (i.e., $x \leqslant y$ implies $Ax \leqslant Ay$ for any conformable vectors x, y).

1.2.3.2 Statement of the Theorem

Let A be in $\mathbb{M}^{n\times n}$. In general, $r(A)$ is not an eigenvalue of A. For example,

$$A = \operatorname{diag}(-1, 0) \implies \sigma(A) = \{-1, 0\} \text{ while } r(A) = 1.$$

But $r(A)$ is always an eigenvalue when $A \geqslant 0$. This is just one implication of the following famous theorem.

Theorem 1.2.6 (Perron–Frobenius) *If $A \geqslant 0$, then $r(A)$ is an eigenvalue of A with nonnegative real right and left eigenvectors:*

$$\exists \text{ nonzero } e, \varepsilon \in \mathbb{R}^n_+ \text{ such that } Ae = r(A)e \text{ and } \varepsilon^\top A = r(A)\varepsilon^\top. \tag{1.10}$$

If A is irreducible, then, in addition,

(i) $r(A)$ is strictly positive and a simple eigenvalue;
(ii) the eigenvectors e and ε are everywhere positive; and
(iii) eigenvectors of A associated with other eigenvalues fail to be nonnegative.

If A is primitive, then, in addition,

(i) the inequality $|\lambda| \leqslant r(A)$ is strict for all eigenvalues λ of A distinct from $r(A)$; and
(ii) with e and ε normalized so that $\langle \varepsilon, e\rangle = 1$, we have

$$r(A)^{-m} A^m \to e\, \varepsilon^\top \qquad (m \to \infty). \tag{1.11}$$

The fact that $r(A)$ is simple under irreducibility means that its eigenvectors are unique up to scalar multiples. We will exploit this property in several important uniqueness proofs.

In the present context, $r(A)$ is called the **dominant eigenvalue** or **Perron root** of A, while ε and e are called the **dominant left and right eigenvectors** of A, respectively.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

Why do we use the word "dominant" here? To help illustrate, let us suppose that $A \in \mathbb{M}^{n\times n}$ is primitive and fix any $x \in \mathbb{R}^n$. Consider what happens to the point $x_m := A^m x$ as m grows. By (1.11) we have $A^m x \approx r(A)^m ce$ for large m, where $c = \varepsilon^\top x$. In other words, asymptotically, the sequence $(A^m x)_{m\in\mathbb{N}}$ is just scalar multiples of e, growing at rate $\ln r(A)$. Thus, $r(A)$ dominates other eigenvalues in controlling the growth rate of $A^m x$, while e dominates other eigenvectors in controlling the direction of growth.

Exercise 1.2.23. The $n \times n$ matrix $P := e\,\varepsilon^\top$ in (1.11) is called the **Perron projection** of A. Prove that $P^2 = P$ (a property that is often used to define projection matrices) and rank $P = 1$. Describe the one-dimensional space that P projects all of $\mathbb{R}^n$ into.

Example 1.2.2: Fix $A \geqslant 0$. If $r(A) = 1$, then $I - A$ is not invertible. To see this, observe that, by Theorem 1.2.6, since $r(A)$ is an eigenvalue of A, there exists a nonzero vector e such that $(I - A)e = 0$. The claim follows. (Why?)

1.2.3.3 Worker Dynamics II

We omit the full proof of Theorem 1.2.6, which is quite long and can be found in Meyer (2000), Seneta (2006b), or Meyer-Nieberg (2012).[6] Instead, to build intuition, let us prove the theorem in a rather simple special case.

The special case we will consider is the class of matrices

$$P_w = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix} \quad \text{with} \quad 0 \leqslant \alpha, \beta \leqslant 1.$$

This example is drawn from the study of worker dynamics in §1.2.1.3.

You might recall from §1.2.1.3 that $\lambda_1 = 1$ and $\lambda_2 = 1 - \alpha - \beta$. Clearly $r(A) = 1$, so $r(A)$ is an eigenvalue, as claimed by the first part of the Perron–Frobenius theorem.

From now on we assume that $\min\{\alpha, \beta\} > 0$, which just means that we are excluding the identity matrix in order to avoid some tedious qualifying remarks.

The two right eigenvectors (e_1, e_2) and two left eigenvectors $(\varepsilon_1, \varepsilon_2)$ are, respectively,

$$e_1 := \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad e_2 := \begin{pmatrix} -\alpha \\ \beta \end{pmatrix}, \quad \varepsilon_1 := \frac{1}{\alpha+\beta}\begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \quad \text{and} \quad \varepsilon_2 := \begin{pmatrix} \alpha \\ -\alpha \end{pmatrix}.$$

Exercise 1.2.24. Verify these claims. (The right eigenvectors were treated in §1.2.1.3.)

Exercise 1.2.25. Recall from Exercise 1.2.20 that P_w is irreducible if and only if both α and β are strictly positive. Show that all the claims about irreducible

[6] See also Glynn and Desai (2018), which provides a new proof of the main results, based on probabilistic arguments, including extensions to infinite state spaces.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

matrices in the Perron–Frobenius theorem are valid for P_w under this irreducibility condition.

EXERCISE 1.2.26. Recall from Exercise 1.2.20 that P_w is primitive if and only if $0 < \alpha, \beta \leqslant 1$ and $\min\{\alpha, \beta\} < 1$. Verify the claim (1.11) for P_w under these conditions. In doing so, you can use the expression for P_w^m in (1.3).

1.2.3.4 Bounding the Spectral Radius

Using the Perron–Frobenius theorem, we can provide useful bounds on the spectral radius of a nonnegative matrix. In what follows, fix $A = (a_{ij}) \in \mathbb{M}^{n \times n}$ and set

- $\text{rowsum}_i(A) := \sum_j a_{ij} =$ the i-th row sum of A and
- $\text{colsum}_j(A) := \sum_i a_{ij} =$ the j-th column sum of A.

LEMMA 1.2.7 *If $A \geqslant 0$, then*

(i) $\min_i \text{rowsum}_i(A) \leqslant r(A) \leqslant \max_i \text{rowsum}_i(A)$ *and*
(ii) $\min_j \text{colsum}_j(A) \leqslant r(A) \leqslant \max_j \text{colsum}_j(A)$.

Proof Let A be as stated and let e be the right eigenvector in (1.10). Since e is nonnegative and nonzero, we can and do assume that $\sum_j e_j = 1$. From $Ae = r(A)e$, we have $\sum_j a_{ij} e_j = r(A) e_i$ for all i. Summing with respect to i gives $\sum_j \text{colsum}_j(A) e_j = r(A)$. Since the elements of e are nonnegative and sum to one, $r(A)$ is a weighted average of the column sums. Hence the second pair of bounds in Lemma 1.2.7 holds. The remaining proof is similar (use the left eigenvector). □

1.3 Probability

Next we review some elements of probability that will be required for analysis of networks.

1.3.1 Discrete Probability

We first introduce probability models on finite sets and then consider sampling methods and stochastic matrices.

1.3.1.1 Probability on Finite Sets

Throughout this text, if S is a finite set, then we set

$$\mathscr{D}(S) := \left\{ \varphi \in \mathbb{R}_+^S : \sum_{x \in S} \varphi(x) = 1 \right\}$$

and call $\mathscr{D}(S)$ the set of **distributions** on S. We say that a random variable X taking values in S has distribution $\varphi \in \mathscr{D}(S)$ and write $X \stackrel{d}{=} \varphi$ if

$$\mathbb{P}\{X = x\} = \varphi(x) \quad \text{for all } x \in S.$$

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

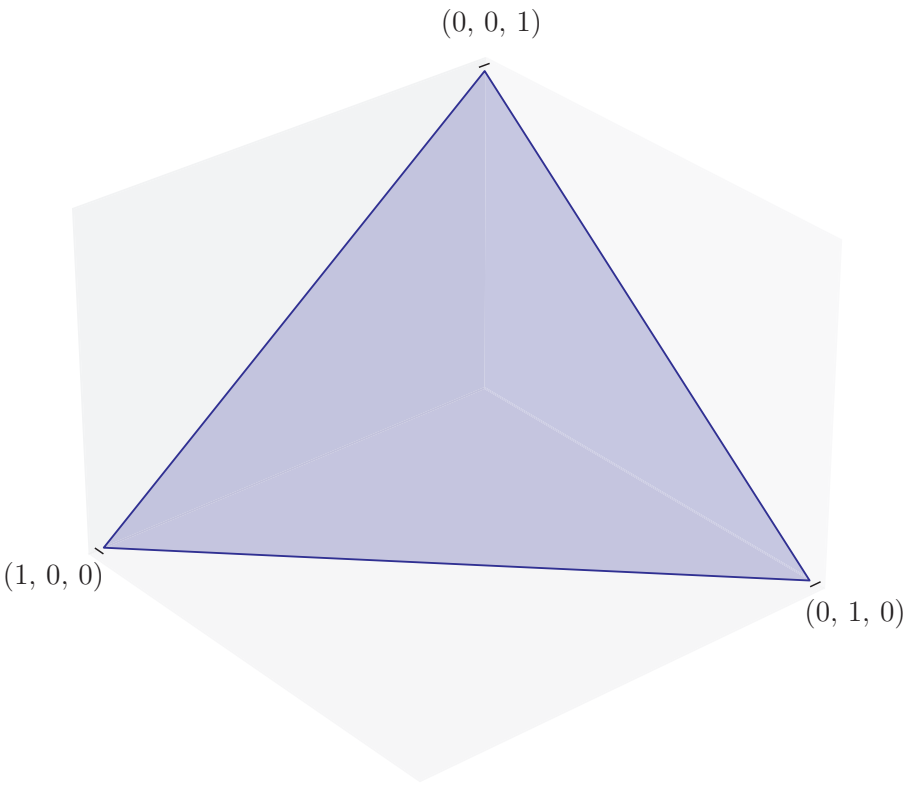

Figure 1.4 If $S = \{1,2,3\}$, then $\mathscr{D}(S)$ is the unit simplex in $\mathbb{R}^3$

A distribution φ can also be understood as a vector $(\varphi(x_i))_{i=1}^n \in \mathbb{R}^n$ (see Lemma A.1.2 in §A.1.3). As a result, $\mathscr{D}(S)$ can be viewed as a subset of $\mathbb{R}^n$. Figure 1.4 provides a visualization when $S = \{1,2,3\}$. Each $\varphi \in \mathscr{D}(S)$ is identified by the point $(\varphi(1), \varphi(2), \varphi(3))$ in $\mathbb{R}^3$.

More generally, if $|S| = n$, then $\mathscr{D}(S)$ can be identified with the **unit simplex** in $\mathbb{R}^n$, which is the set of all n-vectors that are nonnegative and sum to one.

Throughout, given $x \in S$, we use the symbol δ_x to represent the element of $\mathscr{D}(S)$ that puts all mass on x. In other words, $\delta_x(y) = \mathbb{1}\{y = x\}$ for all $y \in S$. In Figure 1.4, each δ_x is a vertex of the unit simplex.

We frequently make use of the **law of total probability**, which states that, for a random variable X on S and arbitrary $A \subset S$,

$$\mathbb{P}\{X \in A\} = \sum_i \mathbb{P}\{X \in A \mid X \in B_i\}\mathbb{P}\{X \in B_i\}, \tag{1.12}$$

where $\{B_i\}$ is a partition of S (i.e., finite collection of disjoint subsets of S such that their union equals S).

Exercise 1.3.1. Prove (1.12) assuming $\mathbb{P}\{X \in B_i\} > 0$ for all i.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

1.3.1.2 Inverse Transform Sampling

Let S be a finite set. Suppose we have the ability to generate random variables that are uniformly distributed on $(0, 1]$. We now want to generate random draws from S that are distributed according to arbitrary $\varphi \in \mathscr{D}(S)$.

Let W be uniformly distributed on $(0, 1]$, so that, for any $a \leqslant b \in (0, 1]$, we have $\mathbb{P}\{a < W \leqslant b\} = b - a$, which is the length of the interval $(a, b]$.[7] Our problem will be solved if we can create a function $z \mapsto \kappa(z)$ from $(0, 1]$ to S such that $\kappa(W)$ has distribution φ. One technique is as follows. First we divide the unit interval $(0, 1]$ into disjoint subintervals, one for each $x \in S$. The interval corresponding to x is denoted $I(x)$ and is chosen to have length $\varphi(x)$. More specifically, when $S = \{x_1, \ldots, x_N\}$, we take

$$I(x_i) := (q_{i-1}, q_i], \quad \text{where} \quad q_i := \varphi(x_1) + \cdots + \varphi(x_i) \quad \text{and} \quad q_0 := 0.$$

You can easily confirm that the length of $I(x_i)$ is $\varphi(x_i)$ for all i.

Now consider the function $z \mapsto \kappa(z)$ defined by

$$\kappa(z) := \sum_{x \in S} x \mathbb{1}\{z \in I(x)\} \qquad (z \in (0, 1]), \tag{1.13}$$

where $\mathbb{1}\{z \in I(x)\}$ is one when $z \in I(x)$ and zero otherwise. It turns out that $\kappa(W)$ has the distribution we desire.

Exercise 1.3.2. Prove:

(i) For all $x \in S$, we have $\kappa(z) = x$ if and only if $z \in I(x)$.
(ii) The random variable $\kappa(W)$ has distribution φ.

Exercise 1.3.3. Let φ, κ, and W be as defined above. Prove that $\mathbb{E}\mathbb{1}\{\kappa(W) = j\} = \varphi(j)$ holds for all $j \in [n]$.

Exercise 1.3.4. Using Julia or another language of your choice, implement the inverse transform sampling procedure described above when $S = \{1, 2, 3\}$ and $\varphi = (0.2, 0.1, 0.7)$. Generate $1,000,000$ (quasi)independent draws (X_i) from φ, and confirm that $(1/n) \sum_{i=1}^{n} \mathbb{1}\{X_i = j\} \approx \varphi(j)$ for $j \in \{1, 2, 3\}$.

The last exercise tells us that the law of large numbers holds in this setting, since, under this law, we expect that

$$\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{X_i = j\} \to \mathbb{E}\mathbb{1}\{X_i = j\}$$

with probability one as $n \to \infty$. In view of Exercise 1.3.3, the right-hand side equals $\varphi(j)$.

[7] The probability is the same no matter whether inequalities are weak or strict.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

Exercise 1.3.5. Suppose that, on a computer, you can generate only uniform random variables on $(0, 1]$, and you wish to simulate a flip of a biased coin with heads probability $\delta \in (0, 1)$. Propose a method.

Exercise 1.3.6. Suppose that, on a computer, you are able to sample from distributions φ and ψ defined on some set S. The set S can be discrete or continuous and, in the latter case, the distributions are understood as densities. Propose a method to sample on a computer from the convex combination $f(s) = \delta\varphi(s) + (1 - \delta)\psi(s)$, where $\delta \in (0, 1)$.

1.3.1.3 Stochastic Matrices

A matrix $P = (p_{ij}) \in \mathbb{M}^{n\times n}$ is called a **stochastic matrix** if

$$P \geqslant 0 \quad \text{and} \quad P\mathbb{1} = \mathbb{1}, \text{ where } \mathbb{1} \in \mathbb{R}^n \text{ is a column vector of ones.}$$

In other words, P is nonnegative and has unit row sums.

We will see many applications of stochastic matrices in this text. Often the applications are probabilistic, where each row of P is interpreted as a distribution over a finite set.

Exercise 1.3.7. Let P, Q be $n \times n$ stochastic matrices. Prove the following facts.

(i) PQ is also stochastic.
(ii) $r(P) = 1$.
(iii) There exists a row vector $\psi \in \mathbb{R}^n_+$ such that $\psi\mathbb{1} = 1$ and $\psi P = \psi$.

The vector ψ in part (iii) of Exercise 1.3.7 is called the *PageRank vector* by some authors, due to its prominence in Google's PageRank algorithm. We will call it a **stationary distribution** instead.[8] Stationary distributions play a key role in the theory of Markov chains, to be treated in §4.1. Ranking methods are discussed again in §1.4.3. PageRank is treated in more detail in §4.2.3.3.

1.3.2 Power Laws

Next we discuss distributions on the (nondiscrete) sets $\mathbb{R}$ and $\mathbb{R}_+$. We are particularly interested in a certain class of distributions that are apparently nonstandard and yet appear with surprising regularity in economics, social science, and the study of networks. We refer to distributions that are said to obey a "power law."

In what follows, given a real-valued random variable X, the function

$$F(t) := \mathbb{P}\{X \leqslant t\} \qquad (t \in \mathbb{R})$$

[8] Stationary distributions of stochastic matrices were intensively studied by many mathematicians well over a century before Larry Page and Sergey Brin patented the PageRank algorithm, so it seems unfair to allow them to appropriate the name.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

is called the **cumulative distribution function** (CDF) of X. The **counter** CDF (CCDF) of X is the function $G(t) := \mathbb{P}\{X > t\} = 1 - F(t)$.

A useful property that holds for any nonnegative random variable X and $p \in \mathbb{R}_+$ is the identity

$$\mathbb{E}\, X^p = \int_0^\infty p t^{p-1} \mathbb{P}\{X > t\}\, \mathrm{d}t. \tag{1.14}$$

See, for example, Çınlar (2011), p. 63.

1.3.2.1 Heavy Tails

Recall that a random variable X on $\mathbb{R}$ is said to be **normally distributed** with mean μ and variance σ^2, and we write $X \overset{d}{=} N(\mu, \sigma^2)$, if X has density

$$\varphi(t) := \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left(\frac{-(t-\mu)^2}{2\sigma^2}\right) \qquad (t \in \mathbb{R}).$$

One notable feature of the normal density is that the tails of the density approach zero quickly. For example, $\varphi(t)$ goes to zero like $\exp(-t^2)$ as $t \to \infty$, which is extremely fast.

A random variable X on $\mathbb{R}_+$ is called **exponentially distributed** and we write $X \overset{d}{=} \mathrm{Exp}(\lambda)$ if, for some $\lambda > 0$, X has density

$$p(t) = \lambda \mathrm{e}^{-\lambda t} \qquad (t \geqslant 0).$$

The tails of the exponential density go to zero like $\exp(-t)$ as $t \to \infty$, which is also relatively fast.

When a distribution is relatively light-tailed, in the sense that its tails go to zero quickly, draws rarely deviate more than a few standard deviations from the mean. In the case of a normal random variable, the probability of observing a draw more than three standard deviations above the mean is around 0.0014. For six standard deviations, the probability falls to 10^{-11}.

In contrast, for some distributions, "extreme" outcomes occur relatively frequently. The left panel of Figure 1.5 helps to illustrate this by simulating 1,000 independent draws from Student's t-distribution, with 1.5 degrees of freedom. For comparison, the right subfigure shows an equal number of independent draws from the $N(0, 4)$ distribution. The Student's t draws reveal tight clustering around zero combined with a few large deviations.

Formally, a random variable X on $\mathbb{R}$ is called **light-tailed** if its **moment generating function**

$$m(t) := \mathbb{E}\mathrm{e}^{tX} \qquad (t \geqslant 0) \tag{1.15}$$

is finite for at least one $t > 0$. Otherwise X is called **heavy-tailed**.[9]

[9] Terminology on heavy tails varies across the literature, but our choice is increasingly standard. See, for example, Foss et al. (2011) or Nair et al. (2021).

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

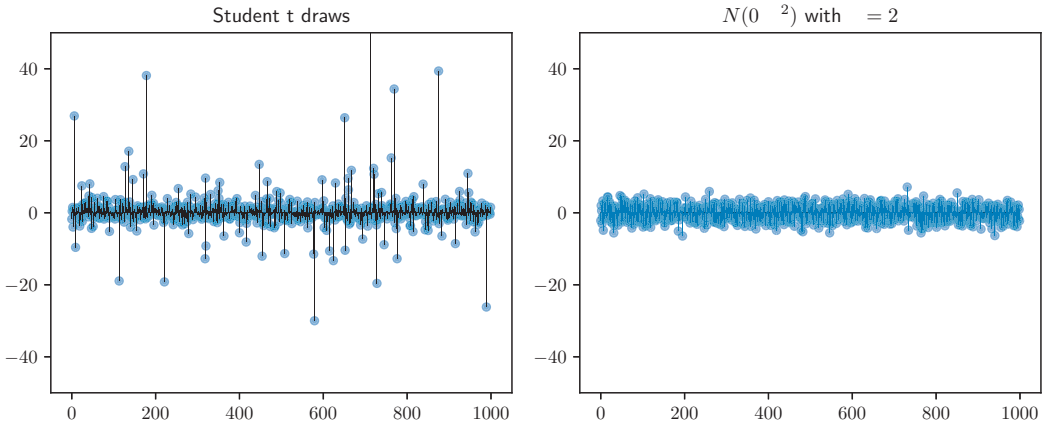

Figure 1.5 Independent draws from Student's t- and normal distributions

Example 1.3.1: If $X \stackrel{d}{=} N(\mu, \sigma^2)$, the moment generating function of X is known to be

$$m(t) = \exp\left(\mu t + \frac{t^2\sigma^2}{2}\right) \qquad (t \geqslant 0).$$

Hence X is light-tailed.

Example 1.3.2: A random variable X on $(0, \infty)$ is said to have **lognormal density** and we write $X \stackrel{d}{=} LN(\mu, \sigma^2)$ if $\ln X \stackrel{d}{=} N(\mu, \sigma^2)$. The mean and variance of this distribution are, respectively,

$$\mathbb{E}\, X = \exp(\mu + \sigma^2/2) \quad \text{and} \quad \text{Var } X = (\exp(\sigma^2) - 1)\exp(2\mu + \sigma^2).$$

The moment generating function $m(t)$ is known to be infinite for all $t > 0$, so any lognormally distributed random variable is heavy-tailed.

For any random variable X and any $r \geqslant 0$, the (possibly infinite) expectation $\mathbb{E}|X|^r$ called the r-th **moment** of X.

LEMMA 1.3.1 *Let X be a random variable on $\mathbb{R}_+$. If X is light-tailed, then all of its moments are finite.*

Proof Pick any $r > 0$. We will show that $\mathbb{E}X^r < \infty$. Since X is light-tailed, there exists a $t > 0$ such that $m(t) = \mathbb{E}\exp(tX) < \infty$. For a sufficiently large constant $\bar{x}$ we have $\exp(tx) \geqslant x^r$ whenever $x \geqslant \bar{x}$. As a consequence, with F as the distribution of X, we have

$$\mathbb{E}X^r = \int_0^{\bar{x}} x^r\, F(\mathrm{d}x) + \int_{\bar{x}}^{\infty} x^r\, F(\mathrm{d}x) \leqslant \bar{x}^r + m(t) < \infty. \qquad \square$$

EXERCISE 1.3.8. Prove that the lognormal distribution has finite moments of every order.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

Together with Lemma 1.3.1, Exercise 1.3.8 shows that existence of an infinite moment is a sufficient but not necessary condition for heavy tails.

1.3.2.2 Pareto Tails

Given $\alpha > 0$, a nonnegative random variable X is said to have a **Pareto tail** with **tail index** α if there exists a $c > 0$ such that

$$\lim_{t \to \infty} t^{\alpha} \, \mathbb{P}\{X > t\} = c. \tag{1.16}$$

In other words, the CCDF G of X satisfies

$$G(t) \approx c t^{-\alpha} \text{ for large } t. \tag{1.17}$$

If X has a Pareto tail for some $\alpha > 0$, then X is also said to obey a **power law**.

Example 1.3.3: A random variable X on $\mathbb{R}_+$ is said to have a **Pareto distribution** with parameters $\bar{x}, \alpha > 0$ if its CCDF obeys

$$G(t) = \begin{cases} 1 & \text{if } t < \bar{x} \\ (\bar{x}/t)^{\alpha} & \text{if } t \geqslant \bar{x} \end{cases}. \tag{1.18}$$

It should be clear that such an X has a Pareto tail with tail index α.

Regarding Example 1.3.3, note that the converse is not true: Pareto-tailed random variables are not necessarily Pareto-distributed, since the Pareto tail property only restricts the far right-hand tail.

EXERCISE 1.3.9. Show that, if X has a Pareto tail with tail index α, then $\mathbb{E}[X^r] = \infty$ for all $r \geqslant \alpha$. [Hint: Use (1.14).]

From Exercise 1.3.9 and Lemma 1.3.1, we see that every Pareto-tailed random variable is heavy-tailed. The converse is not true, since the Pareto tail property (1.16) is very specific. Despite this, it turns out that many heavy-tailed distributions encountered in the study of networks are, in fact, Pareto-tailed.

EXERCISE 1.3.10. Prove: If $X \overset{d}{=} \text{Exp}(\lambda)$ for some $\lambda > 0$, then X does not obey a power law.

1.3.2.3 Empirical Power Law Plots

When the Pareto tail property holds, the CCDF satisfies $\ln G(t) \approx \ln c - \alpha \ln t$ for large t. In other words, G is eventually log linear. Figure 1.6 illustrates this using a Pareto distribution. For comparison, the CCDF of an exponential distribution is also shown.

If we replace the CCDF G with its empirical counterpart – which returns, for each x, the fraction of the sample with values greater than x – we should also obtain an approximation to a straight line under the Pareto tail assumption.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

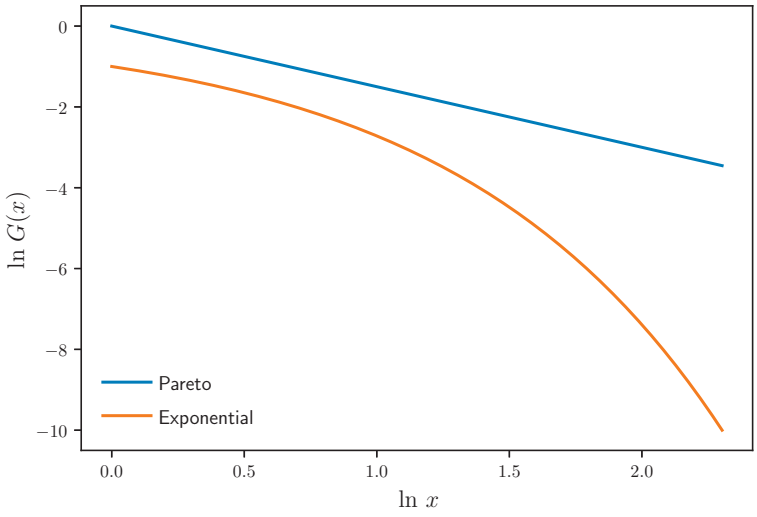

Figure 1.6 CCDF plots for the Pareto and exponential distributions

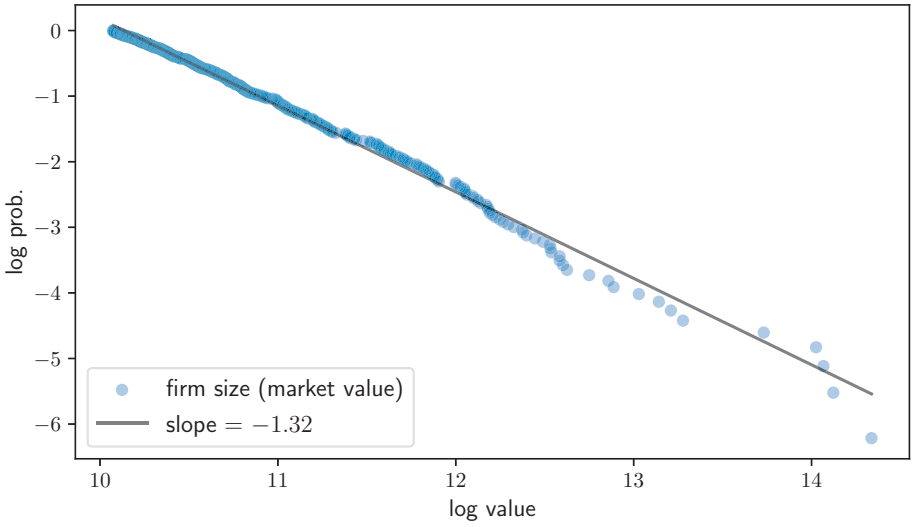

Figure 1.7 Empirical CCDF plots for largest firms (Forbes)

For example, consider the cross-sectional distribution of firm sizes. While the precise nature of this distribution depends on the measure of firm size, the sample of firms, and other factors, the typical picture is one of extreme heavy tails. As an illustration, Figure 1.7 shows an empirical CCDF log–log plot for market values of the largest 500 firms in the Forbes Global 2000 list, as of March 2021. The slope estimate and data distribution are consistent with a Pareto tail and infinite population variance.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

1.3.2.4 Discrete Power Laws

Let X be a random variable with the Pareto distribution, as described in Example 1.3.3. The density of this random variable on the set $[\bar{x}, \infty)$ is $p(t) = ct^{-\gamma}$ with $c := \alpha \bar{x}^{\alpha}$ and $\gamma := \alpha + 1$. The next exercise extends this idea.

EXERCISE 1.3.11. Let X be a random variable with density p on $\mathbb{R}_+$. Suppose that, for some constants $c > 0$, $\gamma > 1$, and $\bar{x} \in \mathbb{R}_+$, we have

$$p(t) = ct^{-\gamma} \quad \text{whenever} \quad t \geqslant \bar{x}. \tag{1.19}$$

Prove that X is Pareto-tailed with tail index $\alpha := \gamma - 1$.

The discrete analog of (1.19) is a distribution on the positive integers with

$$f(k) = ck^{-\gamma} \tag{1.20}$$

for large k. In the special case where this equality holds for all $k \in \mathbb{N}$, and c is chosen so that $\sum_{k \in \mathbb{N}} f(k) = 1$, we obtain the **zeta distribution**.[10]

In general, when we see a probability mass function with the specification (1.20) for large k, we can identify this with a Pareto tail, with tail index $\alpha = \gamma - 1$. Figure 1.8 illustrates with $\gamma = 2$.

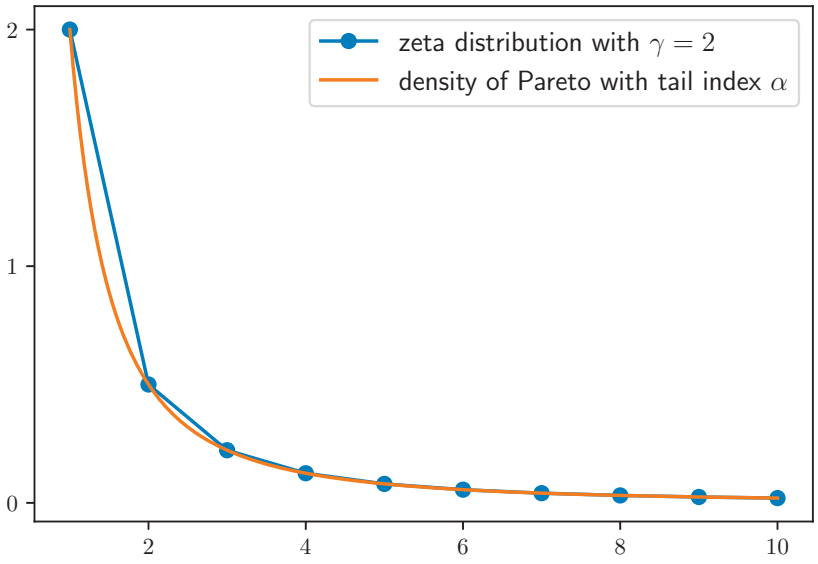

Figure 1.8 Zeta and Pareto distributions

[10] Obviously the correct value of c depends on γ, so we can write $c = H(\gamma)$ for some suitable function H. The correct function for this normalization is called the Riemann zeta function.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

1.4 Graph Theory

Graph theory is a major branch of discrete mathematics. It plays an essential role in this text because it forms the foundations of network analysis. This section provides a concise introduction to graph theory suitable for our purposes.[11]

Graph theory has another closely related use: Many economic models are stochastic and dynamic, which means that they specify states of the world and rates of transition between them. One of the most natural ways to conceptualize these notions is to view states as vertices in a graph and transition rates as relationships between them.

We begin with definitions and fundamental concepts. We focus on directed graphs, where there is a natural asymmetry in relationships (bank A lends money to bank B, firm A supplies goods to firm B, etc.). This costs no generality, since undirected graphs (where relationships are symmetric two-way connections) can be recovered by insisting on symmetry (i.e., existence of a connection from A to B implies existence of a connection from B to A).

1.4.1 Unweighted Directed Graphs

We begin with unweighted directed graphs and examine standard properties, such as connectedness and aperiodicity.

1.4.1.1 Definition and Examples

A **directed graph** or **digraph** is a pair $\mathcal{G} = (V, E)$, where

- V is a finite nonempty set and
- E is a collection of ordered pairs $(u, v) \in V \times V$ called **edges**.

Elements of V are called the **vertices** or **nodes** of $\mathcal{G}$. Intuitively and visually, an edge (u, v) is understood as an arrow from vertex u to vertex v.

Two graphs are given in Figures 1.9 and 1.10. Each graph has three vertices. In these cases, the arrows (edges) could be thought of as representing positive possibility of transition over a given unit of time.

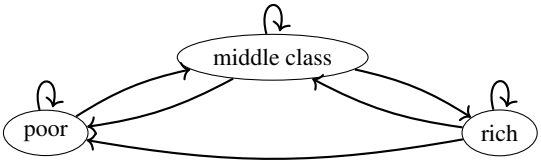

Figure 1.9 A digraph of classes

[11] Graph theory is often regarded as originating from work by the brilliant Swiss mathematician Leonhard Euler (1707–1783), including his famous paper on the "Seven Bridges of Königsberg."

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

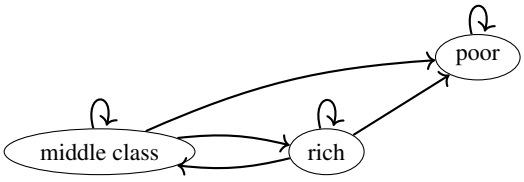

Figure 1.10 An alternative edge list

For a given edge (u, v), the vertex u is called the **tail** of the edge, while v is called the **head**. Also, u is called a **direct predecessor** of v, and v is called a **direct successor** of u. For $v \in V$, we use the following notation:

- $\mathcal{I}(v) :=$ the set of all direct predecessors of v
- $\mathcal{O}(v) :=$ the set of all direct successors of v

Also, the **in-degree** and **out-degree** of $v \in V$ are defined by

- the $i_d(v) := |\mathcal{I}(v)|$ and
- the $o_d(v) := |\mathcal{O}(v)|$, respectively.

If $i_d(v) = 0$ and $o_d(v) > 0$, then v is called a **source**. If either $\mathcal{O}(v) = \emptyset$ or $\mathcal{O}(v) = \{v\}$, then v is called a **sink**. For example, in Figure 1.10, "poor" is a sink with an in-degree of 3.

1.4.1.2 Digraphs in NetworkX

Both Python and Julia provide valuable interfaces to numerical computing with graphs. Of these libraries, the Python package NetworkX is probably the most mature and fully developed. It provides a convenient data structure for representing digraphs and implements many common routines for analyzing them. To import it into Python we run

```
import networkx as nx
```

In all of the code snippets shown below, we assume readers have executed this import statement, as well as

```
import numpy as np
import matplotlib.pyplot as plt
```

As an example, let us create the digraph in Figure 1.10, which we denote henceforth by $\mathcal{G}_p$. To do so, we first create an empty `DiGraph` object:

```
G_p = nx.DiGraph()
```

Next we populate it with nodes and edges. To do this we write down a list of all edges, with `poor` represented by `p` and so on:

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

```
edge_list = [
    ('p', 'p'),
    ('m', 'p'), ('m', 'm'), ('m', 'r'),
    ('r', 'p'), ('r', 'm'), ('r', 'r')
]
```

Finally, we add the edges to our `DiGraph` object:

```
for e in edge_list:
    u, v = e
    G_p.add_edge(u, v)
```

Adding the edges automatically adds the nodes, so `G_p` is now a correct representation of $\mathcal{G}_p$. For our small digraph we can verify this by plotting the graph via NetworkX with the following code:

```
fig, ax = plt.subplots()
nx.draw_spring(G_p, ax=ax, node_size=500, with_labels=True,
                font_weight='bold', arrows=True, alpha=0.8,
                connectionstyle='arc3,rad=0.25', arrowsize=20)
plt.show()
```

This code produces Figure 1.11, which matches the original digraph in Figure 1.10.

`DiGraph` objects have methods that calculate the in-degrees and out-degrees of vertices. For example,

```
G_p.in_degree('p')
```

prints 3.

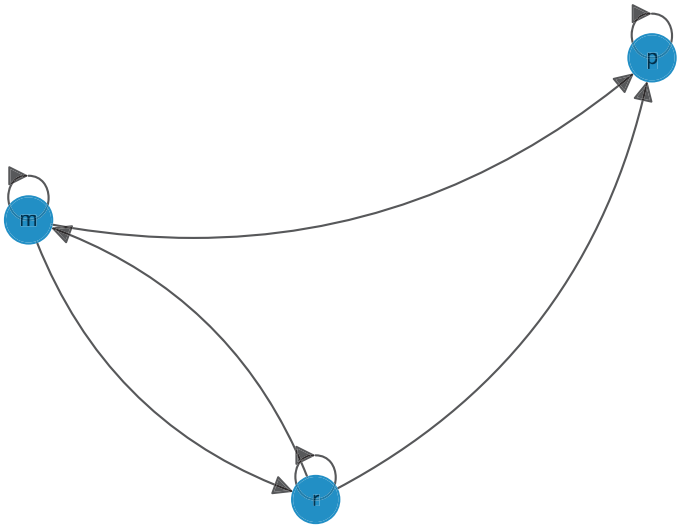

Figure 1.11 NetworkX digraph plot

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

1.4.1.3 Communication

Next we study communication and connectedness, which have important implications for production, financial, transportation, and other networks, as well as for dynamic properties of Markov chains.

A **directed walk** from vertex u to vertex v of a digraph $\mathscr{G}$ is a finite sequence of vertices, starting with u and ending with v, such that any consecutive pair in the sequence is an edge of $\mathscr{G}$. A **directed path** from u to v is a directed walk from u to v such that all vertices in the path are distinct. For example, in Figure 1.12, $(3, 2, 3, 2, 1)$ is a directed walk from 3 to 1 but not a directed path, while $(3, 2, 1)$ is both a directed path and a directed walk from 3 to 1.

As is standard, the **length** of a directed walk (or path) counts the number of edges rather than vertices. For example, the directed path $(3, 2, 1)$ from 3 to 1 in Figure 1.12 is said to have length 2.

Vertex v is called **accessible** (or **reachable**) from vertex u, and we write $u \to v$, if either $u = v$ or there exists a directed path from u to v. A set $U \subset V$ is called **absorbing** for the directed graph (V, E) if no element of $V \setminus U$ is accessible from U.

Example 1.4.1: Let $\mathscr{G} = (V, E)$ be a digraph representing a production network, where elements of V are sectors, and $(i, j) \in E$ means that i supplies products or services to j. Then sector m is an upstream supplier of sector ℓ whenever $m \to \ell$.

Example 1.4.2: The vertex {poor} in the Markov digraph displayed in Figure 1.10 is absorbing, since {middle, rich} is not accessible from {poor}.

Two vertices u and v are said to **communicate** if $u \to v$ and $v \to u$.

Exercise 1.4.1. Let (V, E) be a directed graph, and write $u \sim v$ if u and v communicate. Show that $\sim$ is an equivalence relation (see §A.1.2).

Since communication is an equivalence relation, it induces a partition of V into a finite collection of equivalence classes. Within each of these classes, all elements communicate. These classes are called **strongly connected components**. The graph itself is called **strongly connected** if there is only one such component; that is, v is accessible from u for any pair $(u, v) \in V \times V$. This corresponds to the idea that any node can be reached from any other.

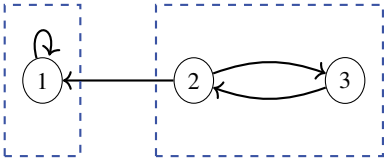

Figure 1.12 Strongly connected components of a digraph (rectangles)

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

Example 1.4.3: Figure 1.12 shows a digraph with strongly connected components $\{1\}$ and $\{2,3\}$. The digraph is not strongly connected.

Example 1.4.4: In Figure 1.9, the digraph is strongly connected. In contrast, in Figure 1.10, rich is not accessible from poor, so the graph is not strongly connected.

NetworkX can be used to test for communication and strong connectedness, as well as to compute strongly connected components. For example, applied to the digraph in Figure 1.12, the code

```
G = nx.DiGraph()
G.add_edge(1, 1)
G.add_edge(2, 1)
G.add_edge(2, 3)
G.add_edge(3, 2)

list(nx.strongly_connected_components(G))
```

prints `[{1}, {2, 3}]`.

1.4.1.4 Aperiodicity

A **cycle** $(u, v, w, \ldots, u)$ of a directed graph $\mathcal{G} = (V, E)$ is a directed walk in $\mathcal{G}$ such that (i) the first and last vertices are equal and (ii) no other vertex is repeated. The graph is called a **directed acyclic graph** if it contains no cycles. The graph is called **periodic** if it contains at least one cycle and, moreover, there exists a $k > 1$ such that the length of every cycle is a multiple of k. The graph is called **aperiodic** if it is not periodic.

Example 1.4.5: In Figure 1.13, the cycles are (a,b,a), (b,a,b), (b,c,b), (c,b,c), (c,d,c) and (d,c,d). Hence the length of every cycle is 2 and the graph is periodic.

Exercise 1.4.2. Prove the following: If $\mathcal{G}$ is a directed acyclic graph, then, for any node u in $\mathcal{G}$, there exists a node v such that $u \to v$ and $o_d(v) = 0$.

An obvious sufficient condition for aperiodicity is existence of even one self-loop. The digraphs in Figures 1.9–1.12 are aperiodic for this reason.

The next result provides an easy way to understand aperiodicity for connected graphs. Proofs can be found in Norris (1998) and Häggström et al. (2002).

Figure 1.13 A periodic digraph

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

LEMMA 1.4.1 *Let $\mathcal{G} = (V, E)$ be a digraph. If $\mathcal{G}$ is strongly connected, then $\mathcal{G}$ is aperiodic if and only if, for all $v \in V$, there exists a $q \in \mathbb{N}$ such that, for all $k \geqslant q$, there exists a directed walk of length k from v to v.*

It is common to call a vertex v satisfying the condition in Lemma 1.4.1 **aperiodic**. With this terminology, Lemma 1.4.1 states that a strongly connected digraph is aperiodic if and only if every vertex is aperiodic.

NetworkX can be used to check for aperiodicity of vertices or graphs. For example, if `G` is a `DiGraph` object, then `nx.is_aperiodic(G)` returns `True` or `False` depending on the aperiodicity of `G`.

1.4.1.5 Adjacency Matrices

There is a simple map between edges of a graph with fixed vertices and a binary matrix called an adjacency matrix. The benefit of viewing connections through adjacency matrices is that they bring the power of linear algebra to the analysis of digraphs. We illustrate this briefly here and extensively in §1.4.2.

If $\mathcal{G} = (V, E)$ is a digraph with $V = \{v_1, \ldots, v_n\}$, then the $n \times n$ **adjacency matrix** corresponding to (V, E) is defined by[12]

$$A = (a_{ij})_{1 \leqslant i,\, j \leqslant n} \quad \text{with} \quad a_{ij} = \mathbb{1}\{(v_i, v_j) \in E\}. \tag{1.21}$$

For example, with {poor, middle, rich} mapped to $(1, 2, 3)$, the adjacency matrix corresponding to the digraph in Figure 1.10 is

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}. \tag{1.22}$$

An adjacency matrix provides us with enough information to recover the edges of a graph. More generally, given a set of vertices $V = \{v_1, \ldots, v_n\}$, an $n \times n$ matrix $A = (a_{ij})_{1 \leqslant i,\, j \leqslant n}$ with binary entries generates a digraph $\mathcal{G}$ with vertices V and edges

$$E = \{(v_i, v_j) \in V \times V : a_{ij} = 1\}.$$

The adjacency matrix of this graph (V, E) is A.

EXERCISE 1.4.3. A digraph (V, E) is called **undirected** if $(u, v) \in E$ implies $(v, u) \in E$. What property does this imply on the adjacency matrix?

REMARK 1.4.1 *The idea that a digraph can be undirected, presented in Exercise 1.4.3, seems contradictory. After all, a digraph is a* directed *graph. Another way to introduce undirected graphs is to define them as a vertex-edge pair (V, E), where each edge $\{u, v\} \in E$ is an unordered pair, rather than an ordered pair (u, v). However,*

[12] Note that, in some applied fields, the adjacency matrix is transposed: $a_{ij} = 1$ if there is an edge from j to i, rather than from i to j. We will avoid this odd and confusing definition (which contradicts both standard graph theory and standard notational conventions in the study of Markov chains).

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

the definition in Exercise 1.4.3 is essentially equivalent and more convenient for our purposes, since we mainly study directed graphs.

Like NetworkX, the Python library `quantecon` provides a graph object that supplies certain graph-theoretic algorithms. In the case of QuantEcon's `DiGraph` object, algorithms are implemented by interfacing with routines in SciPy, and an instance is created by supplying an adjacency matrix. For example, to construct a digraph corresponding to Figure 1.10, we use the corresponding adjacency matrix (1.22):

```
import quantecon as qe
import numpy as np

A = ((1, 0, 0),
     (1, 1, 1,),
     (1, 1, 1))
A = np.array(A)        # Convert to NumPy array
G = qe.DiGraph(A)
```

Let's print the set of strongly connected components, as a list of NumPy arrays:

```
G.strongly_connected_components
```

The output is `[array([0]), array([1, 2])]`.

1.4.2 Weighted Digraphs

Early quantitative work on networks tended to focus on unweighted digraphs, where the existence or absence of an edge is treated as sufficient information (e.g., following or not following on social media, existence or absence of a road connecting two towns). However, for some networks, this binary measure is less significant than the size or strength of the connection.

As one illustration, consider Figure 1.14, which shows flows of funds (i.e., loans) between private banks, grouped by country of origin. An arrow from Japan to the USA, say, indicates aggregate claims held by Japanese banks on all US-registered banks, as collected by the Bank of International Settlements (BIS). The size of each node in the figure is increasing in the total foreign claims of all other nodes on this node. The widths of the arrows are proportional to the foreign claims they represent.[13] The country codes are given in Table 1.1.

In this network, an edge (u, v) exists for almost every choice of u and v (i.e., almost every country in the network).[14] Hence existence of an edge is not particularly informative. To understand the network, we need to record not just the existence or

[13] Data for the figure was obtained from the BIS consolidated banking statistics, for Q4 of 2022. Our calculations used the immediate counterparty basis for financial claims of domestic and foreign banks, which calculates the sum of cross-border claims and local claims of foreign affiliates in both foreign and local currencies. The foreign claim of a node to itself is set to zero.

[14] In fact arrows representing foreign claims less than US$10 million are cut from Figure 1.14, so the network is even denser than it appears.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

Table 1.1 Codes for the 16-country financial network

AU	Australia	DE	Germany	CL	Chile	ES	Spain
PT	Portugal	FR	France	TR	Turkey	GB	United Kingdom
US	United States	IE	Ireland	AT	Austria	IT	Italy
BE	Belgium	JP	Japan	SW	Switzerland	SE	Sweden

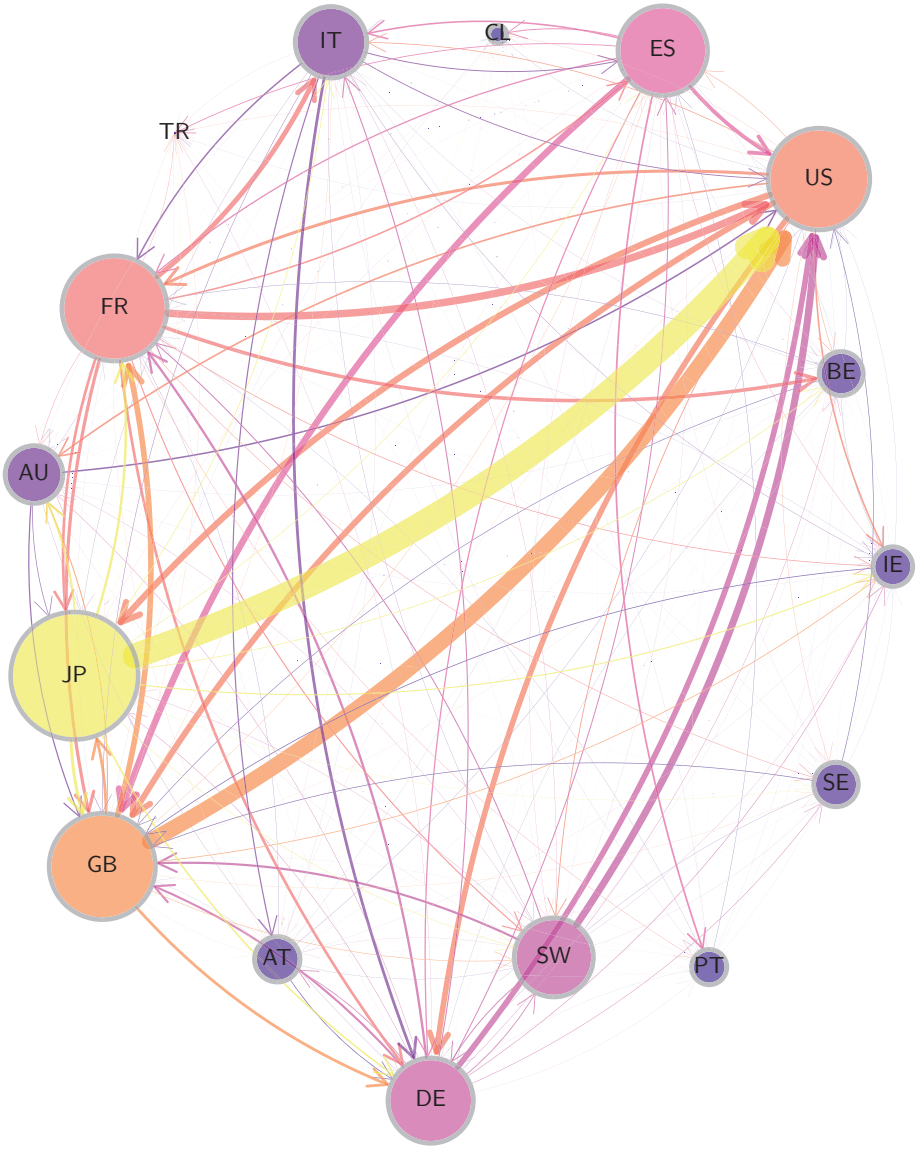

Figure 1.14 International private credit flows by country

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

absence of a credit flow, but also the size of the flow. The correct data structure for recording this information is a "weighted directed graph," or "weighted digraph." In this section we define this object and investigate its properties.

1.4.2.1 Definitions

A **weighted digraph** $\mathcal{G}$ is a triple (V, E, w) such that (V, E) is a digraph and w is a function from E to $(0, \infty)$, called the **weight function**.

REMARK 1.4.2 *Weights are traditionally regarded as nonnegative. In this text we insist that weights are also positive, in the sense that* $w(u, v) > 0$ *for all* $(u, v) \in E$. *The reason is that the intuitive notion of zero weight is understood, here and below, as absence of a connection. In other words, if* (u, v) *has "zero weight," then* (u, v) *is not in* E, *so* w *is not defined on* (u, v).

Example 1.4.6: As suggested by the discussion above, the graph shown in Figure 1.14 can be viewed as a weighted digraph. Vertices are countries of origin, and an edge exists between country u and country v when private banks in u lend nonzero quantities to banks in v. The weight assigned to edge (u, v) gives total loans from u to v as measured according to the discussion of Figure 1.14.

Example 1.4.7: Figure 1.15 shows a weighted digraph, with arrows representing edges of the induced digraph (compare with the unweighted digraph in Figure 1.9). The numbers next to the edges are the weights. In this case, you can think of the numbers on the arrows as transition probabilities for a household over, say, one year. For example, a rich household has a 10% chance of becoming poor.

The definitions of **accessibility**, **communication**, **periodicity**, and **connectedness** extend to any weighted digraph $\mathcal{G} = (V, E, w)$ by applying them to (V, E). For example, (V, E, w) is called strongly connected if (V, E) is strongly connected. The weighted digraph in Figure 1.15 is strongly connected.

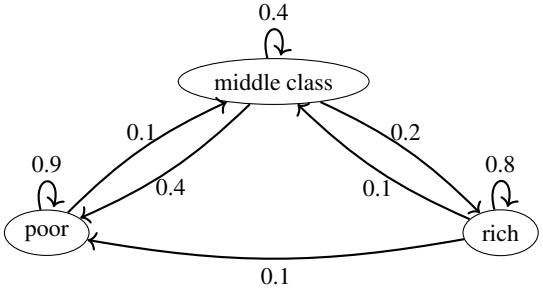

Figure 1.15 A weighted digraph

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

1.4.2.2 Adjacency Matrices of Weighted Digraphs

In §1.4.1.5 we discussed adjacency matrices of unweighted digraphs. The **adjacency matrix** of a *weighted* digraph (V, E, w) with vertices $\{v_1, \ldots, v_n\}$ is the matrix

$$A = (a_{ij})_{1 \leqslant i,\, j \leqslant n} \quad \text{with} \quad a_{ij} = \begin{cases} w(v_i, v_j) & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, once the vertices in V are enumerated, the weight function and adjacency matrix provide the same information. We often work with the latter, since it facilitates computations.

Example 1.4.8: With {poor, middle, rich} mapped to $(1, 2, 3)$, the adjacency matrix corresponding to the weighted digraph in Figure 1.15 is

$$A = \begin{pmatrix} 0.9 & 0.1 & 0 \\ 0.4 & 0.4 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{pmatrix}. \tag{1.23}$$

In QuantEcon's `DiGraph` implementation, weights are recorded via the keyword `weighted`:

```
A = ((0.9, 0.1, 0.0),
     (0.4, 0.4, 0.2),
     (0.1, 0.1, 0.8))
A = np.array(A)
G = qe.DiGraph(A, weighted=True)     # Store weights
```

REMARK 1.4.3 *Every unweighted digraph can be regarded as a weighted digraph by introducing a weight function that assigns unit weight to each edge. The resulting adjacency matrix is binary and agrees with our original definition for unweighted digraphs in* (1.21)*. In this sense, the set of unweighted digraphs is a subset of the set of all weighted digraphs.*

One of the key points to remember about adjacency matrices is that taking the transpose "reverses all the arrows" in the associated digraph.

Example 1.4.9: The digraph in Figure 1.16 can be interpreted as a stylized version of a financial network, with vertices as banks and edges showing flow of funds, similar to Figure 1.14 on page 31. For example, we see that bank 2 extends a loan of size 200 to bank 3. The corresponding adjacency matrix is

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

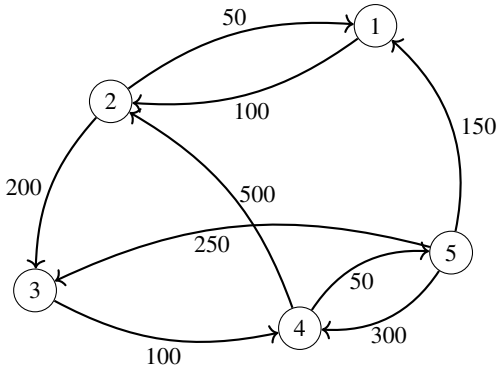

Figure 1.16 A network of credit flows across institutions

$$A = \begin{pmatrix} 0 & 100 & 0 & 0 & 0 \\ 50 & 0 & 200 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 \\ 0 & 500 & 0 & 0 & 50 \\ 150 & 0 & 250 & 300 & 0 \end{pmatrix}. \tag{1.24}$$

The transposition is

$$A^\top = \begin{pmatrix} 0 & 50 & 0 & 0 & 150 \\ 100 & 0 & 0 & 500 & 0 \\ 0 & 200 & 0 & 0 & 250 \\ 0 & 0 & 100 & 0 & 300 \\ 0 & 0 & 0 & 50 & 0 \end{pmatrix}. \tag{1.25}$$

The corresponding network is visualized in Figure 1.17. This figure shows the network of liabilities after the loans have been granted. Both of these networks (original and transpose) are useful for the analysis of financial markets (see, e.g., Chapter 5).

It is not difficult to see that each nonnegative $n \times n$ matrix $A = (a_{ij})$ can be viewed as the adjacency matrix of a weighted digraph with vertices equal to $[n]$. The weighted digraph $\mathscr{G} = (V, E, w)$ in question is formed by setting

$$V = [n], \quad E = \{(i,j) \in V \times V : a_{ij} > 0\} \quad \text{and} \quad w(i,j) = a_{ij} \text{ for all } (i,j) \in E.$$

We call $\mathscr{G}$ the **weighted digraph induced by** A.

The next exercise helps to reinforce the point that transposes reverse the edges.

Exercise 1.4.4. Let $A = (a_{ij})$ be a nonnegative $n \times n$ matrix, and let $\mathscr{G} = ([n], E, w)$ and $\mathscr{G}' = ([n], E', w')$ be the weighted digraphs induced by A and $A^\top$, respectively. Show that

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

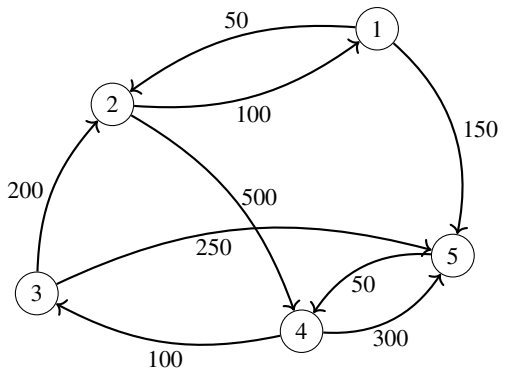

Figure 1.17 The transpose: a network of liabilities

(i) $(j,k) \in E'$ if and only if $(k,j) \in E$.
(ii) $j \to k$ in $\mathscr{G}'$ if and only if $k \to j$ in $\mathscr{G}$.

1.4.2.3 Application: Quadratic Network Games

Acemoglu et al. (2016) and Zenou (2016) consider quadratic games with n agents where agent k seeks to maximize

$$u_k(x) := -\frac{1}{2}x_k^2 + \alpha x^\top A x + x_k \varepsilon_k. \tag{1.26}$$

Here $x = (x_i)_{i=1}^n$, A is a symmetric matrix with $a_{ii} = 0$ for all i, $\alpha \in (0,1)$ is a parameter, and $\varepsilon = (\varepsilon_i)_{i=1}^n$ is a random vector. (This is the set up for the quadratic game in §21.2.1 of Acemoglu et al. (2016).) The k-th agent takes the decisions x_j as given for all $j \neq k$ when maximizing (1.26).

In this context, A is understood as the adjacency matrix of a graph with vertices $V = [n]$, where each vertex is one agent. We can reconstruct the weighted digraph (V, E, w) by setting $w(i,j) = a_{ij}$ and letting E be all (i,j) pairs in $[n] \times [n]$ with $a_{ij} > 0$. The weights identify some form of relationship between the agents, such as influence or friendship.

Exercise 1.4.5. A **Nash equilibrium** for the quadratic network game is a vector $x^* \in \mathbb{R}^n$ such that, for all $i \in [n]$, the choice x_i^* of agent i maximizes (1.26) taking x_j^* as given for all $j \neq i$. Show that, whenever $r(A) < 1/\alpha$, a unique Nash equilibrium x^* exists in $\mathbb{R}^n$ and, moreover, $x^* := (I - \alpha A)^{-1}\varepsilon$.

The network game described in this section has many interesting applications, including social networks, crime networks and peer networks. References are provided in §1.5.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

1.4.2.4 Properties

In this section, we examine some of the fundamental properties of and relationships among digraphs, weight functions, and adjacency matrices. Throughout this section, without loss of generality, we consider a weighted digraph with $V = [n]$.

As an additional convention, if A is an adjacency matrix, and A^k is the k-th power of A, then we write a_{ij}^k for a typical element of A^k. With this notation, we observe that, since $A^{(s+t)} = A^s A^t$, the rules of matrix multiplication imply

$$a_{ij}^{s+t} = \sum_{\ell=1}^{n} a_{i\ell}^s \, a_{\ell j}^t \qquad (i, j \in [n], \;\; s, t \in \mathbb{N}). \tag{1.27}$$

(A^0 is the identity.) The next proposition explains the significance of the powers.

PROPOSITION 1.4.2 *Let $\mathcal{G}$ be a weighted digraph with adjacency matrix A. For distinct vertices $i, j \in [n]$ and $k \in \mathbb{N}$, we have*

$$a_{ij}^k > 0 \iff \textit{there exists a directed walk of length } k \textit{ from } i \textit{ to } j.$$

Proof ($\Leftarrow$) The statement is true by definition when $k = 1$. Suppose in addition that $\Leftarrow$ holds at $k-1$, and suppose there exists a directed walk $(i, \ell, m, \ldots, n, j)$ of length k from i to j. By the induction hypothesis we have $a_{in}^{k-1} > 0$. Moreover, (n, j) is part of a directed walk, so $a_{nj} > 0$. Applying (1.27) now gives $a_{ij}^k > 0$.

($\Rightarrow$) Left as an exercise (just use the same logic). □

Example 1.4.10: In §4.1 we show that if elements of A represent one-step transition probabilities across states, then elements of A^t, the t-th power of A, provide t-step transition probabilities. In Markov process theory, (1.27) is called the *Chapman–Kolmogorov equation*.

In this context, the next result is fundamental.

THEOREM 1.4.3 *Let $\mathcal{G}$ be a weighted digraph. The following statements are equivalent:*

(i) $\mathcal{G}$ is strongly connected.
(ii) The adjacency matrix generated by $\mathcal{G}$ is irreducible.

Proof Let $\mathcal{G}$ be a weighted digraph with adjacency matrix A. By Proposition 1.4.2, strong connectedness of $\mathcal{G}$ is equivalent to the statement that, for each $i, j \in V$, we can find a $k \geqslant 0$ such that $a_{ij}^k > 0$. (If $i = j$ then set $k = 0$.) This, in turn, is equivalent to $\sum_{m=0}^{\infty} A^m \gg 0$, which is irreducibility of A. □

Example 1.4.11: Strong connectivity fails in the digraph in Figure 1.18, since vertex 4 is a source. By Theorem 1.4.3, the adjacency matrix must be reducible.

---

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

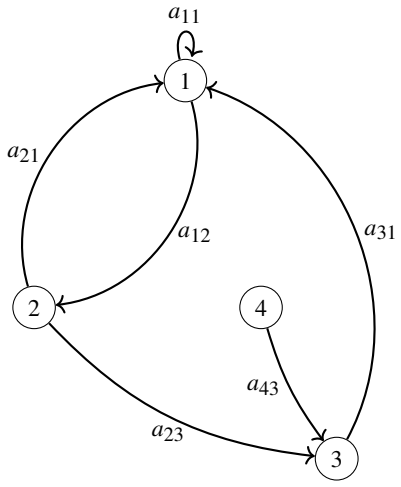

Figure 1.18 Failure of strong connectivity

We will find that the property of being primitive is valuable for analysis. (The Perron–Frobenius theorem hints at this.) What do we need to add to strong connectedness to obtain primitiveness?

THEOREM 1.4.4 *For a weighted digraph $\mathcal{G} = (V, E, w)$, the following statements are equivalent:*

(i) $\mathcal{G}$ is strongly connected and aperiodic.
(ii) The adjacency matrix generated by $\mathcal{G}$ is primitive.

Proof of Theorem 1.4.4 First we show that, if $\mathcal{G}$ is aperiodic and strongly connected, then, for all $i, j \in V$, there exists a $q \in \mathbb{N}$ such that $a_{ij}^k > 0$ whenever $k \geqslant q$. To this end, pick any i, j in V. Since $\mathcal{G}$ is strongly connected, there exists an $s \in \mathbb{N}$ such that $a_{ij}^s > 0$. Since $\mathcal{G}$ is aperiodic, we can find an $m \in \mathbb{N}$ such that $\ell \geqslant m$ implies $a_{jj}^\ell > 0$. Picking $\ell \geqslant m$ and applying (1.27), we have

$$a_{ij}^{s+\ell} = \sum_{r \in V} a_{ir}^s a_{rj}^\ell \geqslant a_{ij}^s a_{jj}^\ell > 0.$$

Thus, with $t = s + m$, we have $a_{ij}^k > 0$ whenever $k \geqslant t$.

((i) $\Rightarrow$ (ii)). By the preceding argument, given any $i, j \in V$, there exists an $s(i, j) \in \mathbb{N}$ such that $a_{ij}^m > 0$ whenever $m \geqslant s(i, j)$. Setting $k := \max s(i, j)$ over all (i, j) yields $A^k \gg 0$.

((ii) $\Rightarrow$ (i)). Suppose that A is primitive. Then, for some $k \in \mathbb{N}$, we have $A^k \gg 0$. Strong connectedness of the digraph follows directly from Proposition 1.4.2. It remains to check aperiodicity.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

Aperiodicity will hold if we can establish that $a_{11}^{k+1} > 0$, since then we have a cycle of length k and another of length $k+1$. To show that this holds, we use (1.27) to write

$$a_{11}^{k+1} = \sum_{\ell \in V} a_{1\ell} a_{\ell 1}^{k} \geqslant \bar{a} \sum_{\ell \in V} a_{i\ell},$$

where $\bar{a} := \min_{\ell \in V} a_{\ell 1}^{k} > 0$. The proof will be done if $\sum_{\ell \in V} a_{1\ell} > 0$. But this must be true, since otherwise vertex 1 is a sink, which contradicts strong connectedness. □

Example 1.4.12: In Exercise 1.2.20 we worked hard to show that P_w is irreducible if and only if $0 < \alpha, \beta \leqslant 1$, using the approach of calculating and then examining the powers of P_w (as shown in (1.3)). However, the result is trivial when we examine the corresponding digraph in Figure 1.3 and use the fact that irreducibility is equivalent to strong connectivity. Similarly, the result in Exercise 1.2.20 that P_w is primitive if and only if $0 < \alpha, \beta \leqslant 1$ and $\min\{\alpha, \beta\} < 1$ becomes much easier to establish if we examine the digraph and use Theorem 1.4.4.

1.4.3 Network Centrality

When studying networks of all varieties, a recurring topic is the relative "centrality" or "importance" of different nodes. One classic application is the ranking of web pages by search engines. Here are some examples related to economics:

- In which industry will one dollar of additional demand have the most impact on aggregate production, once we take into account all the backward linkages? In which sector will a rise in productivity have the largest effect on national output?
- A negative shock endangers the solvency of the entire banking sector. Which institutions should the government rescue, if any?
- In the network games considered in §1.4.2.3, the Nash equilibrium is $x^* = (I - \alpha A)^{-1} \varepsilon$. Players' actions are dependent on the topology of the network, as encoded in A. A common finding is that the level of activity or effort exerted by an agent (e.g., severity of criminal activity by a participant in a criminal network) can be predicted from their "centrality" within the network.

In this section we review essential concepts related to network centrality.[15]

1.4.3.1 Centrality Measures

Let G be the set of weighted digraphs. A **centrality measure** associates to each $\mathcal{G} = (V, E, w)$ in G a vector $m(\mathcal{G}) \in \mathbb{R}^{|V|}$, where the i-th element of $m(\mathcal{G})$ is interpreted as the centrality (or rank) of vertex v_i. In most cases $m(\mathcal{G})$ is nonnegative. In what follows, to simplify notation, we take $V = [n]$.

[15] Centrality measures are sometimes called "influence measures," particularly in connection with social networks.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

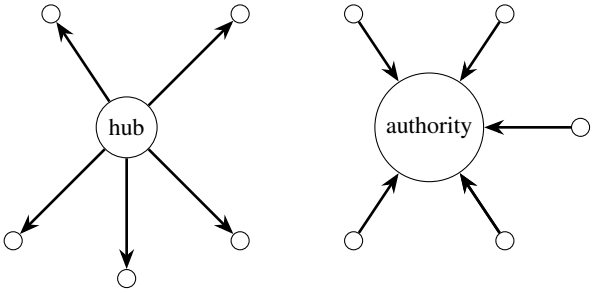

Figure 1.19 Hub vs. authority

(Unfortunately, the definitions and terminology associated with even the most common centrality measures vary widely across the applied literature. Our convention is to follow the mathematicians, rather than the physicists. For example, our terminology is consistent with Benzi and Klymko (2015).)

1.4.3.2 Authorities vs. Hubs

Search engine designers recognize that web pages can be important in two different ways. Some pages have high **hub centrality**, meaning that they *link to* valuable sources of information (e.g., news aggregation sites). Other pages have high **authority centrality**, meaning that they contain valuable information, as indicated by the number and significance of *incoming* links (e.g., websites of respected news organizations). Figure 1.19 helps to visualize the difference.

Similar ideas can be and have been applied to economic networks (often using different terminology). For example, in the production networks we study below, high hub centrality is related to upstreamness: Such sectors tend to supply intermediate goods to many important industries. Conversely, a high authority ranking will coincide with downstreamness.

In what follows we discuss both hub-based and authority-based centrality measures, providing definitions and illustrating the relationship between them.

1.4.3.3 Degree Centrality

Two of the most elementary measures of "importance" of a vertex in a given digraph $\mathscr{G} = (V, E)$ are its in-degree and out-degree. Both of these provide a centrality measure. **In-degree centrality** $i(\mathscr{G})$ is defined as the vector $(i_d(v))_{v \in V}$. **Out-degree centrality** $o(\mathscr{G})$ is defined as $(o_d(v))_{v \in V}$. If $\mathscr{G}$ is expressed as a NetworkX `DiGraph` called `G` (see, e.g., §1.4.1.2), then $i(\mathscr{G})$ can be calculated via

```
iG = [G.in_degree(v) for v in G.nodes()]
```

This method is relatively slow when $\mathscr{G}$ is a large digraph. Since vectorized operations are generally faster, let's look at an alternative method using operations on arrays.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

To illustrate the method, recall the network of financial institutions in Figure 1.16. We can compute the in-degree and out-degree centrality measures by first converting the adjacency matrix, which is shown in (1.24), to a binary matrix that corresponds to the adjacency matrix of the same network viewed as an unweighted graph:

$$U = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}. \tag{1.28}$$

Now $U(i, j) = 1$ if and only if i points to j. The out-degree and in-degree centrality measures can be computed as

$$o(\mathcal{G}) = U\mathbb{1} \quad \text{and} \quad i(\mathcal{G}) = U^\top \mathbb{1}, \tag{1.29}$$

respectively. That is, summing the rows of U gives the out-degree centrality measure, while summing the columns gives the in-degree measure.

The out-degree centrality measure is a hub-based ranking, while the vector of in-degrees is an authority-based ranking. For the financial network in Figure 1.16, a high out-degree for a given institution means that it lends to many other institutions; a high in-degree indicates that many institutions lend to it.

Notice that, to switch from a hub-based ranking to an authority-based ranking, we need only transpose the (binary) adjacency matrix U. We will see that the same is true for other centrality measures. This is intuitive, since transposing the adjacency matrices reverses the directions of the edges (Exercise 1.4.4).

For a weighted digraph $\mathcal{G} = (V, E, w)$ with adjacency matrix A, the **weighted out-degree centrality** and **weighted in-degree centrality** measures are defined as

$$o(\mathcal{G}) = A\mathbb{1} \quad \text{and} \quad i(\mathcal{G}) = A^\top \mathbb{1}, \tag{1.30}$$

respectively, by analogy with (1.29). We present some intuition for these measures in applications below.

Unfortunately, while in- and out-degree measures of centrality are simple to calculate, they are not always informative. As an example, consider again the international credit network shown in Figure 1.14. There, an edge exists between almost every node, so the in- or out-degree-based centrality ranking fails to effectively separate the countries. This can be seen in the out-degree ranking of countries corresponding to that network in the top left panel of Figure 1.20 and in the in-degree ranking in the top right.

There are other limitations of degree-based centrality rankings. For example, suppose web page A has many inbound links, while page B has fewer. Even though page A dominates in terms of in-degree, it might be less important than web page B to, say, a potential advertiser, when the links into B are from more heavily trafficked pages. Thinking about this point suggests that importance can be recursive: The importance of a given node depends on the importance of other nodes that link to it. The next set of centrality measures we turn to has this recursive property.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

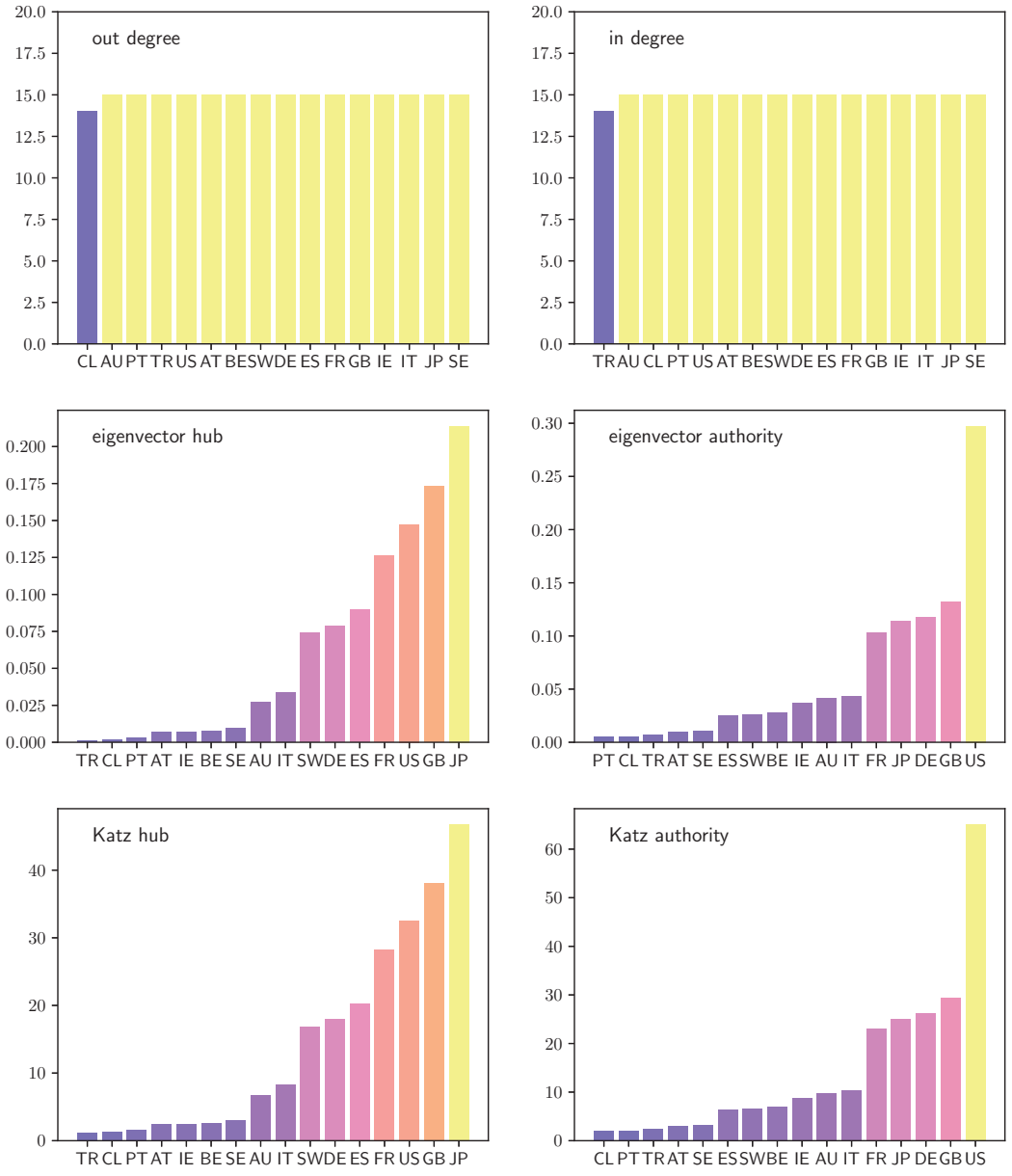

Figure 1.20 Centrality measures for the credit network

1.4.3.4 Eigenvector Centrality

Let $\mathcal{G} = (V, E, w)$ be a weighted digraph with adjacency matrix A. Recalling that $r(A)$ is the spectral radius of A, the **hub-based eigenvector centrality** of $\mathcal{G}$ is defined as the $e \in \mathbb{R}^n_+$ that solves

$$e = \frac{1}{r(A)} Ae. \tag{1.31}$$

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

Element-by-element, this is

$$e_i = \frac{1}{r(A)} \sum_{j \in [n]} a_{ij} e_j \qquad \text{for all } i \in [n]. \tag{1.32}$$

Note the recursive nature of the definition: The centrality obtained by vertex i is proportional to a sum of the centralities of all vertices, weighted by the "rates of flow" from i into these vertices. A vertex i is highly ranked if (a) there are many edges leaving i, (b) these edges have large weights, and (c) the edges point to other highly ranked vertices.

When we study demand shocks in §2.1.3, we will provide a more concrete interpretation of eigenvector centrality. We will see that, in production networks, sectors with high hub-based eigenvector centrality are important *suppliers*. In particular, they are activated by a wide array of demand shocks once orders flow backwards through the network.

Exercise 1.4.6. Show that (1.32) has a unique solution, up to a positive scalar multiple, whenever A is strongly connected.[16]

As the name suggests, hub-based eigenvector centrality is a measure of hub centrality: Vertices are awarded high rankings when they *point to* important vertices. The next two exercises help to reinforce this point.

Exercise 1.4.7. Show that nodes with zero out-degree always have zero hub-based eigenvector centrality.

To compute eigenvector centrality when the adjacency matrix A is primitive, we can employ the Perron–Frobenius theorem, which tells us that $r(A)^{-m} A^m \to e\, \varepsilon^\top$ as $m \to \infty$, where ε and e are the dominant left and right eigenvectors of A. This implies

$$r(A)^{-m} A^m \mathbb{1} \to ce, \quad \text{where } c := \varepsilon^\top \mathbb{1}. \tag{1.33}$$

Thus, evaluating $r(A)^{-m} A^m \mathbb{1}$ at large m returns a scalar multiple of e. The package NetworkX provides a function for computing eigenvector centrality via (1.33).

One issue with this method is the assumption of primitivity, since the convergence in (1.33) can fail without it. The following function uses an alternative technique, based on Arnoldi iteration, which typically works even when primitivity fails. (The `authority` option is explained below.)

```
import numpy as np
from scipy.sparse import linalg

def eigenvector_centrality(A, m=40, authority=False):
    """
    Computes and normalizes the dominant eigenvector of A.
    """
```

[16] While the dominant eigenvector is only defined up to a positive scaling constant, this is no reason for concern, since positive scaling has no impact on the ranking. In most cases, users of this centrality ranking choose the dominant eigenvector e satisfying $\|e\| = 1$.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

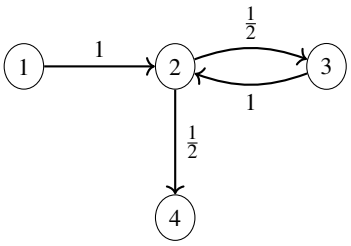

Figure 1.21 A network with a source and a sink

```
    A_temp = A.T if authority else A
    r, vec_r = linalg.eigs(A_temp, k=1, which='LR')
    e = vec_r.flatten().real
    return e / np.sum(e)
```

Exercise 1.4.8. Show that the digraph in Figure 1.21 is not primitive. Using the code above or another suitable routine, compute the hub-based eigenvector centrality rankings. You should obtain values close to $e = (0.3694, 0.2612, 0.3694, 0)$. Note that the sink vertex (vertex 4) obtains the lowest rank.

The middle left panel of Figure 1.20 shows the hub-based eigenvector centrality ranking for the international credit network shown in Figure 1.14. Countries that are rated highly according to this rank tend to be important players in terms of supply of credit. Japan takes the highest rank according to this measure, although countries with large financial sectors, such as Great Britain and France, are not far behind. (The color scheme in Figure 1.14 is also matched to hub-based eigenvector centrality.)

The **authority-based eigenvector centrality** of $\mathcal{G}$ is defined as the $e \in \mathbb{R}^n_+$ solving

$$e = \frac{1}{r(A)} A^\top e. \tag{1.34}$$

The difference between (1.34) and (1.32) is just transposition of A. (Transposes do not affect the spectral radius of a matrix.) Element-by-element, this is

$$e_j = \frac{1}{r(A)} \sum_{i \in [n]} a_{ij} e_i \qquad \text{for all } j \in [n]. \tag{1.35}$$

We see e_j will be high if many nodes with high authority rankings link to j.

The middle right panel of Figure 1.20 shows the authority-based eigenvector centrality ranking for the international credit network shown in Figure 1.14. Highly ranked countries are those that attract large inflows of credit, or credit inflows from other major players. The USA clearly dominates the rankings as a target of inter-bank credit.

Exercise 1.4.9. Assume that A is strongly connected. Show that authority-based eigenvector centrality is uniquely defined up to a positive scaling constant and equal to the dominant *left* eigenvector of A.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

1.4.3.5 Katz Centrality

Eigenvector centrality can be problematic. Although the definition in (1.32) makes sense when A is strongly connected (so that, by the Perron–Frobenius theorem, $r(A) > 0$), strong connectedness fails in many real world networks. We will see examples of this in §2.1, for production networks defined by input–output matrices.

In addition, while strong connectedness yields strict positivity of the dominant eigenvector, many vertices can be assigned a zero ranking when it fails (see, e.g., Exercise 1.4.7). This zero ranking often runs counter to our intuition when we examine specific networks.

Considerations such as these encourage the use of an alternative notion of centrality for networks called Katz centrality, originally due to Katz (1953), which is positive under weaker conditions and uniquely defined up to a tuning parameter. Fixing β in $(0, 1/r(A))$, the **hub-based Katz centrality** of weighted digraph $\mathcal{G}$ with adjacency matrix A, at parameter β, is defined as the vector $\kappa := \kappa(\beta, A) \in \mathbb{R}^n_+$ that solves

$$\kappa_i = \beta \sum_{j \in [n]} a_{ij} \kappa_j + 1 \qquad \text{for all } i \in [n]. \tag{1.36}$$

The intuition is very similar to that provided for eigenvector centrality: High centrality is conferred on i when it is linked to by vertices that themselves have high centrality. The difference between (1.36) and (1.32) is just in the additive constant 1.

Exercise 1.4.10. Show that, under the stated condition $0 < \beta < 1/r(A)$, hub-based Katz centrality is always finite and uniquely defined by

$$\kappa = (I - \beta A)^{-1} \mathbb{1} = \sum_{\ell \geqslant 0} (\beta A)^\ell \mathbb{1}, \tag{1.37}$$

where $\mathbb{1}$ is a column vector of ones.

Exercise 1.4.11. We know from the Perron–Frobenius theorem that the eigenvector centrality measure will be everywhere positive when the digraph is strongly connected. A condition weaker than strong connectivity is that every vertex has positive outdegree. Show that the Katz measure of centrality is strictly positive on each vertex under this condition.

The attenuation parameter β is used to ensure that κ is finite and uniquely defined under the condition $0 < \beta < 1/r(A)$. It can be proved that, when the graph is strongly connected, hub-based (resp., authority-based) Katz centrality converges to the hub-based (resp., authority-based) eigenvector centrality as $\beta \uparrow 1/r(A)$.[17] This is why, in the bottom two panels of Figure 1.20, the hub-based (resp., authority-based) Katz centrality ranking is seen to be close to its eigenvector-based counterpart.

When $r(A) < 1$, we use $\beta = 1$ as the default for Katz centrality computations.

Exercise 1.4.12. Compute the hub-based Katz centrality rankings for the simple digraph in Figure 1.21 when $\beta = 1$. You should obtain $\kappa = (5, 4, 5, 1)$. Hence, the

[17] See, for example, Benzi and Klymko (2015).

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

source vertex (vertex 1) obtains equal highest rank, and the sink vertex (vertex 4) obtains the lowest rank.

Analogously, the **authority-based Katz centrality** of $\mathcal{G}$ is defined as the $\kappa \in \mathbb{R}^n_+$ that solves

$$\kappa_j = \beta \sum_{i \in [n]} a_{ij} \kappa_i + 1 \qquad \text{for all } j \in [n]. \tag{1.38}$$

Exercise 1.4.13. Show that, under the restriction $0 < \beta < 1/r(A)$, the unique solution to (1.38) is given by

$$\kappa = (I - \beta A^\top)^{-1} \mathbb{1} \quad \Longleftrightarrow \quad \kappa^\top = \mathbb{1}^\top (I - \beta A)^{-1}. \tag{1.39}$$

(Verify the stated equivalence.)

Exercise 1.4.14. Compute the authority-based Katz centrality rankings for the digraph in Figure 1.21 when $\beta = 1$. You should obtain $\kappa = (1, 6, 4, 4)$. Notice that the source vertex now has the lowest rank. This is due to the fact that hubs are devalued relative to authorities.

1.4.4 Scale-Free Networks

What kinds of properties do large, complex networks typically possess? One of the most striking facts about complex networks is that many exhibit the **scale-free** property, which means, loosely speaking, that the number of connections possessed by each vertex in the network follows a power law. The scale-free property is remarkable because it holds for a wide variety of networks, from social networks to citation, sales, financial, and production networks, each of which is generated by different underlying mechanisms. Nonetheless, they share this specific statistical structure.

We begin this section by defining the degree distribution and then discuss its properties, including possible power law behavior.

1.4.4.1 Empirical Degree Distributions

Let $\mathcal{G} = (V, E)$ be a digraph. Assuming without loss of generality that $V = [n]$ for some $n \in \mathbb{N}$, the **in-degree distribution** of G is the sequence $(\varphi_{in}(k))_{k=0}^n$ defined by

$$\varphi_{in}(k) = \frac{\sum_{v \in V} \mathbb{1}\{i_d(v) = k\}}{n} \qquad (k = 0, \ldots, n), \tag{1.40}$$

where $i_d(v)$ is the in-degree of vertex v. In other words, the in-degree distribution evaluated at k is the fraction of nodes in the network that have in-degree k. In Python, when $\mathcal{G}$ is expressed as a NetworkX `DiGraph` called `G` and `import numpy as np` has been executed, the in-degree distribution can be calculated via

```
def in_degree_dist(G):
    n = G.number_of_nodes()
    iG = np.array([G.in_degree(v) for v in G.nodes()])
```

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

```
    phi = [np.mean(iG == k) for k in range(n+1)]
    return phi
```

The **out-degree distribution** is defined analogously, replacing i_d with o_d in (1.40), and denoted by $(\varphi_{out}(k))_{k=0}^{n}$.

Recall that a digraph $\mathcal{G} = (V, E)$ is called undirected if $(u, v) \in E$ implies $(v, u) \in E$. If $\mathcal{G}$ is undirected, then $i_d(v) = o_d(v)$ for all $v \in V$. In this case we usually write φ instead of φ_{in} or φ_{out} and refer simply to the **degree-distribution** of the digraph.

A **scale-free network** is a network whose degree distribution obeys a power law, in the sense that there exist positive constants c and γ with

$$\varphi(k) \approx ck^{-\gamma} \quad \text{for large } k. \tag{1.41}$$

Here $\varphi(k)$ can refer to the in-degree or the out-degree (or both), depending on our interest. In view of the discussion in §1.3.2.4, this can be identified with the idea that the degree distribution is Pareto-tailed with tail index $\alpha = \gamma - 1$.

Although we omit formal tests, the degree distribution for the commercial aircraft international trade network shown in Figure 1.2 on page 3 is approximately scale-free. Figure 1.22 illustrates this by plotting the degree distribution alongside $f(x) = cx^{-\gamma}$ with $c = 0.2$ and $\gamma = 1.1$. (In this calculation of the degree distribution, performed by the NetworkX function `degree_histogram`, directions are ignored and the network is treated as an undirected graph.)

Attention was drawn to the scale-free nature of many networks by Barabási and Albert (1999). They found, for example, that the in-degree and out-degree distributions for internet pages connected by hyperlinks both follow power laws. In subsequent years, many networks have been found to have the scale-free property, up to a first

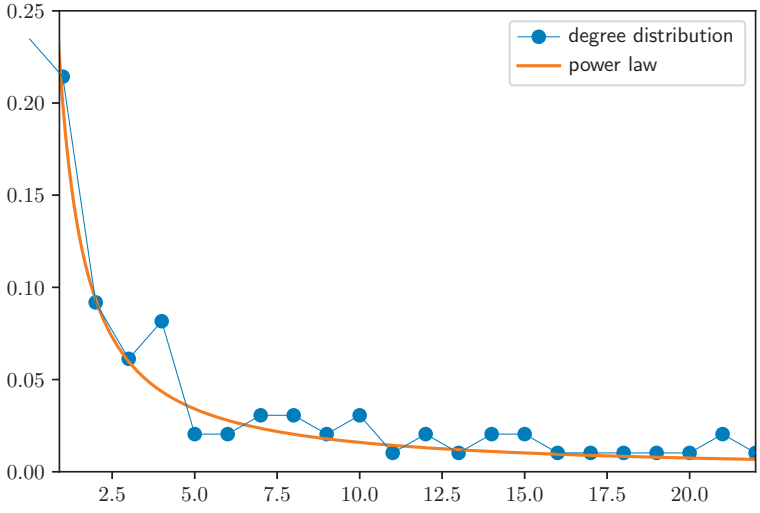

Figure 1.22 Degree distribution for international aircraft trade

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

approximation, including networks of followers on Twitter (Pearce, 2017; Punel and Ermagun, 2018), other social networks (Rybski et al., 2009), and academic collaboration networks (e.g., papers plus citations).

Within economics and finance, Carvalho (2014) shows that the weighted out-degree distribution for US input–output data (discussed further in Chapter 2) obeys a power law, as does the Katz centrality measure. Carvalho et al. (2021) document power law tails for the in-degree (suppliers) and out-degree (customers) distributions in a Japanese network of interacting firms. Scale-free degree distributions have also been observed in a number of financial and inter-bank credit networks (Kim et al., 2007; Ou et al., 2007; De Masi et al., 2011).

In many cases, the scale-free property of a given network has significant implications for economic outcomes and welfare. For example, a power law in input–output networks often typically indicates dominance by a small number of very large sectors or firms. This in turn affects both the dynamism of the industry and the likelihood of aggregate instability caused by firm-level shocks. We explore some of these issues in Chapter 2.

1.4.4.2 Random Graphs

One way to explore the implications of different dynamics for the degree distribution of graphs is to specify a law for generating graphs randomly and then examine the degree distribution that results. This methodology leads to insights on the kinds of mechanisms that can generate scale-free networks.

We begin with one of the most popular and elementary ways of randomly generating an undirected graph, originally examined by Erdös and Rényi (1960). The process to generate a graph $\mathscr{G} = (V, E)$ is

(i) fix an integer $n \in \mathbb{N}$ and a $p \in (0, 1)$;
(ii) view $V := [n]$ as a collection of vertices;
(iii) let $E = \{\emptyset\}$; and
(iv) for each $(i, j) \in V \times V$ with $i \neq j$, add the undirected edge $\{i, j\}$ to the set of edges E with probability p.

In the last step additions are independent – each time, we flip an unbiased IID coin with head probability p and add the edge if the coin comes up heads.

The Python code below provides a function that can be called to randomly generate an undirected graph using this procedure. It applies the `combinations` function from the `itertools` library, which, for the call `combinations(A, k)`, returns a list of all subsets of A of size k. For example,

```
import itertools
letters = 'a', 'b', 'c'
list(itertools.combinations(letters, 2))
```

returns `[('a', 'b'), ('a', 'c'), ('b', 'c')]`.

We use `combinations` to produce the set of all possible edges and then add them to the graph with probability p:

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

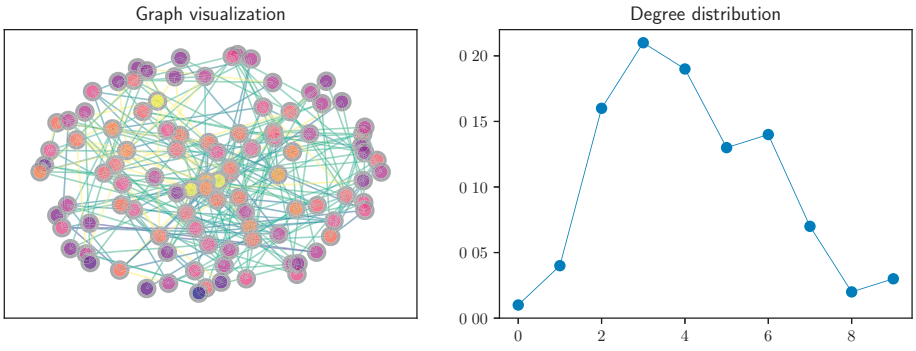

Figure 1.23 An instance of an Erdos–Renyi random graph

```
def erdos_renyi_graph(n=100, p=0.5, seed=1234):
    "Returns an Erdos-Renyi random graph."
    np.random.seed(seed)
    edges = itertools.combinations(range(n), 2)
    G = nx.Graph()

    for e in edges:
        if np.random.rand() < p:
            G.add_edge(*e)
    return G
```

(The code presented here is a simplified version of functionality provided by the library NetworkX. It is written for clarity rather than efficiency. More efficient versions can be found both in NetworkX and in Julia's Graphs.jl library.)

The left-hand side of Figure 1.23 shows one instance of a graph that was generated by the `erdos_renyi_graph` function, with $n = 100$ and $p = 0.05$. Lighter colors on a node indicate higher degree (more connections). The right-hand side shows the degree distribution, which exhibits a bell-shaped curve typical for Erdos–Renyi random graphs. In fact one can show (see, e.g., Bollobás (1999) or Durrett (2007)) that the degree distribution is binomial, with

$$\varphi(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k} \qquad (k = 0, \ldots, n-1).$$

1.4.4.3 Preferential Attachment

Clearly Erdos–Renyi random graphs fail to replicate the heavy right-hand tail of the degree distribution observed in many networks. In response to this, Barabási and Albert (1999) proposed a mechanism for randomly generating graphs that feature the scale-free property.

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

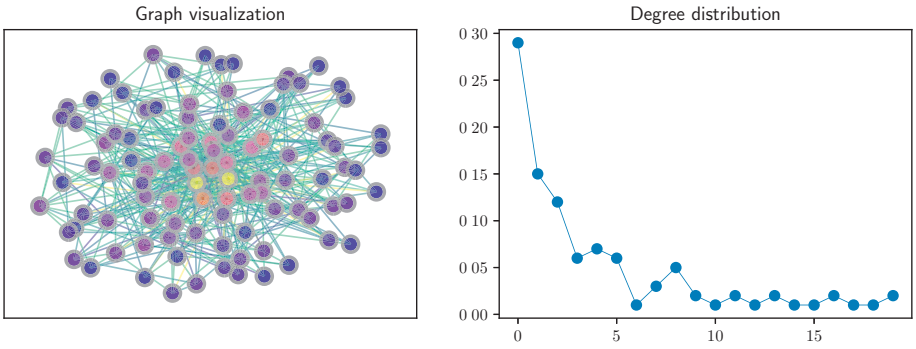

Figure 1.24 An instance of a preferential attachment random graph

The stochastic mechanism they proposed is called **preferential attachment**. In essence, each time a new vertex is added to an undirected graph, it is attached by edges to m of the existing vertices, where the probability of vertex v being selected is proportional to the degree of v. Barabási and Albert (1999) showed that the resulting degree distribution exhibits a Pareto tail in the limit as the number of vertices converges to $+\infty$. A careful proof can be found in Chapter 4 of Durrett (2007).

Although we omit details of the proof, we can see the power law emerge in simulations. For example, Figure 1.24 shows a random graph with 100 nodes generated by NetworkX's `barabasi_albert_graph` function. The number of attachments m is set to 5. The simulated degree distribution on the right-hand side of Figure 1.24 already exhibits a long right tail.

The preferential attachment model is popular not just because it replicates the scale-free property of many real-world networks but also because its mechanism is simple and plausible. For example, in citation networks, we can imagine that a well-cited paper is more likely to attract additional citations than a poorly cited paper. Similar intuition can be applied to an individual on a social network, where the number of links is measured in terms of the number of followers.

1.5 Chapter Notes

The Perron–Frobenius theorem is due to Oskar Perron (1880–1975) and Ferdinand Georg Frobenius (1849–1917). The main results were proved by 1912. As early as 1915, Dénes König (1884–1944) saw the connection between the Perron–Frobenius theorem and graph theory, and provided an alternative proof using bipartite graphs. Some of the history is discussed in Schrijver (2005).

We have already mentioned the textbooks on economic and social networks by Jackson (2010), Easley et al. (2010), Borgatti et al. (2018), and Goyal (2023), as well as the handbook by Bramoullé et al. (2016). Jackson (2014) gives a survey of the

https://doi.org/10.1017/9781009456340.003 Published online by Cambridge University Press

literature. Within the realm of network science, the high level texts by Newman (2018), Menczer et al. (2020), and Coscia (2021) are excellent.

One good text on graphs and graph-theoretic algorithms is Kepner and Gilbert (2011). Ballester et al. (2006) provide an interpretation of Katz centrality (which they call Bonacich centrality) in terms of Nash equilibria of quadratic games. Sharkey (2017) presents a new interpretation of Katz centrality using control theory. Polovnikov et al. (2022) use Katz centrality to uncover hidden ultimate owners in firm ownership data. Du et al. (2015) show how PageRank can be obtained as a competitive equilibrium of an economic problem. Calvó-Armengol et al. (2009) develop a model in which the outcomes for agents embedded in a network are proportional to the Katz centrality. Elliott and Golub (2019) show that, in a setting where agents can create nonrival, heterogeneous public goods, an important set of efficient solutions are characterized by contributions being proportional to agents' eigenvector centralities in the network.

Kumamoto and Kamihigashi (2018) provide a detailed survey of power laws in economics and the social sciences, including a discussion of the preferential attachment model of Barabási and Albert (1999). Newman (2005) is also highly readable. The textbook of Durrett (2007) is rigorous, carefully written, and contains interesting motivational background, as well as an extensive citation list for studies of scale-free networks.

It should be clear from the symbol $\approx$ in (1.41) that the definition of scale-free networks is not entirely rigorous. Moreover, when connecting the definition to observed networks, we cannot obtain complete clarity by taking a limit, as we did when we defined power laws in §1.3.2, since the number of vertices is always finite. This imprecision in the definition has led to heated debate (see, e.g., Holme (2019)). Given the preponderance of positive empirical studies, we take the view that, up to a reasonable degree of approximation, the scale-free property is remarkably widespread.

In §1.4.2.3 we briefly mentioned network games, social networks, and key players. These topics deserve more attention than we are able to provide. An excellent overview is given in Zenou (2016). Amarasinghe et al. (2020) apply these ideas to problems in economic development. Valuable related papers include Allouch (2015), Belhaj et al. (2016), Demange (2017), Belhaj and Deroïan (2019), and Galeotti et al. (2020).

Another topic we reluctantly omit in order to keep the textbook short is endogenous network formation in economic environments. Influential papers in this field include Jackson and Wolinsky (1996), Bala and Goyal (2000), Watts (2001), Hojman and Szeidl (2008), Galeotti and Goyal (2010), and Graham (2017).

Finally, Candogan et al. (2012) study the profit maximization problem for a monopolist who sells items to participants in a social network. The main idea is that, in certain settings, the monopolist will find it profitable to offer discounts to key players in the network. Atalay et al. (2011) argue that in-degrees observed in US buyer–supplier networks have lighter tails than a power law and supply a model that better fits their data.