this version: 26 February 2009

7 Communication Classes

Perhaps surprisingly, we can learn much about the long-run behavior of a Markov chain merely from the *zero pattern* of its transition matrix. In the next section, we use the zero-pattern matrix to partition the states of a chain into *communication classes*, and then obtain a *reduced transition diagram* which can be used to determine whether each communication class is *open* or *closed*. Essentially, this section extends our previous analysis of absorbing states to recognize that a Markov chain may be absorbed into a *set* of states (a closed communication class) even if no individual state in this class is absorbing.

Our analysis of communication classes leads naturally to several definitions useful for understanding the long-run behavior of Markov chains. Based on its zero pattern, we may determine whether a matrix is *reducible* or *irreducible*. Irreducibile matrices may be either primitive or *cyclic*. From the zero pattern, we may also determine whether a matrix is *centered*. We will see that centeredness implies the long-run convergence of linear dynamics, while cyclicity leads to non-convergence.

Finally, we see how the *influence diagram* reflects the interdependence of variables in a simultaneous-equation system. Reducing this diagram, the communication classes correspond to sets of variables that are mutally dependent. When there is more than one communication class, we can "reduce" the overall problem (of solving for all variables in the system) to a series of simpler subproblems. This insight will be employed again in subsequent chapters on influence networks and demography.

7.1 Finding communication classes

This section offers a "recipe" for finding communication classes, constructing the reduced transition diagram, and determining whether each class is open or closed. While developed here for Markov chains, this procedure is also useful for a variety of other applications, as we will see in future chapters.

7.1.1 The zero pattern

The *zero pattern* of a transition matrix is constructed by setting each positive element to 1 while the other elements remain 0. More formally, given a transition matrix P, the elements of its zero-pattern matrix Z are given by

$$Z(i,j) = \begin{cases} 1 & \text{if } P(i,j) > 0 \\ 0 & \text{otherwise} \end{cases} .$$

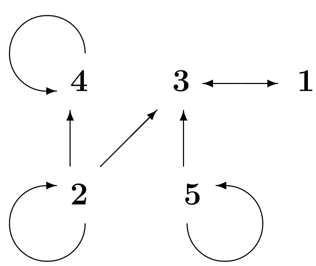
Thus, while P is a *valued* matrix (with entries between 0 and 1), Z is a *binary* matrix (with entries equal to 0 or 1).

Using Matlab, the zero-pattern matrix can be computed simply by testing whether the elements of the transition matrix are positive.

```
>> P   % transition matrix for a 5-state chain

P =
         0         0    1.0000         0         0
         0    0.5000    0.2500    0.2500         0
    1.0000         0         0         0         0
         0         0         0    1.0000         0
         0         0    0.3333         0    0.6667

>> Z = double(P > 0)   % zero-pattern matrix

Z =
     0     0     1     0     0
     0     1     1     1     0
     1     0     0     0     0
     0     0     0     1     0
     0     0     1     0     1
```

The `double` command has been used here to convert the `(P > 0)` matrix from a logical to numeric (double-precision) matrix that can be used in subsequent computations.[1]

The transition diagram for the zero pattern includes a directed edge from state i to state j if and only if $Z(i, j) = 1$. To illustrate, the transition diagram for our example is given below.



Of course, this is quite similar to the transition diagram for the matrix P. However, there is no need to label the edges because the zero-pattern matrix Z merely indicates the possibility (not the precise probability) of a transition.

[1] Matlab will produce an error message if you attempt to multiply together two logical matrices (even if their sizes imply that multiplication is possible). Given any matrix `A` in the Matlab workspace, you can check whether this matrix is numeric or logical (or belongs to some other object class) by typing `class(A)` on the command line. To ensure that this matrix is numeric, you can enter `A = double(A)` on the command line.

7.1.2 Reachability

Borrowing some terminology from social network analysis, we say that state i can *reach* state j when $i = j$ or there is a path of any length from i to j on the transition diagram. For simple cases (such as our present example), we can easily determine *reachability* by inspection of the transition diagram. However, especially when the number of states is large, matrix methods are more convenient. Raising the Z matrix to the power t, element $Z^t(i, j)$ gives the number of paths of length t from i to j.[2] To illustrate, consider the Z^t matrix for $t \in \{2, 3, 4\}$.

```
>> Z^2   % Z^2(i,j) = number of paths of length 2 from i to j

ans =
     1     0     0     0     0
     1     1     1     2     0
     0     0     1     0     0
     0     0     0     1     0
     1     0     1     0     1

>> Z^3   % Z^3(i,j) = number of paths of length 3 from i to j

ans =
     0     0     1     0     0
     1     1     2     3     0
     1     0     0     0     0
     0     0     0     1     0
     1     0     2     0     1

>> Z^4   % Z^4(i,j) = number of paths of length 4 from i to j

ans =
     1     0     0     0     0
     2     1     2     4     0
     0     0     1     0     0
     0     0     0     1     0
     2     0     2     0     1
```

If the interpretation of these matrices is not immediately obvious, readers may find it useful to reconcile these computations with the transition diagram given above, finding the indicated number of paths of each length for each pair of states.[3]

---

[2] We have thus implicitly defined a *path* as any sequence of directed edges leading from i to j. While irrelevant for present purposes, we might reserve the term *simple path* for sequences that never use the same edge more than once. (See, e.g., Johnsonbaugh, *Discrete Mathematics*, 2001.) Simple paths are more difficult to identify through matrix techniques.

[3] The rationale for this procedure is quite similar to rationale for iterated multiplication of the transition matrix presented in Chapter 1. Briefly, it is apparent that $Z^2(i, j) = \sum_k Z(i, k)Z(k, j)$ gives the number of paths of length 2 from i to j. Consequently, $Z^3(i, j) = \sum_k Z^2(i, k)Z(k, j)$ gives the number of paths of length 3 from i to j, and the general result follows by induction.

To characterize the reachability relation using matrix methods, we may thus construct a binary reachability matrix R with elements

$$R(i,j) = \begin{cases} 1 & \text{if } Z^t(i,j) > 0 \text{ for some } t \geq 0 \\ 0 & \text{otherwise} \end{cases} .$$

This equation might suggest the need to compute Z^t for every value of t up to infinity. Fortunately, if Z is an $n \times n$ matrix, we actually need to consider only those values of t up to $n-1$. Intuitively, a path of length $n-1$ could potentially include every node in the transition diagram. If state i cannot reach state j through a path of length $n-1$ or less, then i cannot reach j at all. Consequently, the preceding equation may be rewritten as

$$R(i,j) = \begin{cases} 1 & \text{if } (I + Z + Z^2 + \ldots + Z^{n-1})(i,j) > 0 \\ 0 & \text{otherwise} \end{cases} .$$

Further recognizing that we are interested only in the existence of a path (not the number of paths) from i to j, this is equivalent to

$$R(i,j) = \begin{cases} 1 & \text{if } (I + Z)^{n-1}(i,j) > 0 \\ 0 & \text{otherwise} \end{cases} .$$

For our present example, we may thus compute the reachability matrix as follows.

```
>> R = (eye(5) + Z)^4 > 0   % reachability matrix

R =
     1     0     1     0     0
     1     1     1     1     0
     1     0     1     0     0
     0     0     0     1     0
     1     0     1     0     1
```

To reconcile this matrix with our transition diagram, note that $Z(2,1) = Z(5,1) = 0$ and hence the process cannot transition directly from state 2 or 5 to state 1. However, both states 2 and 5 can reach state 1 indirectly (through state 3) and hence $R(2,1) = R(5,1) = 1$.

7.1.3 Communication classes

Having determined reachability, we can now partition the states of the Markov chain into *communication classes*, assigning states i and j to the same class if and only if each of these states can reach *and be reached by* the other. Formally, we construct a binary *can reach and be reached by* matrix C where

$$C(i,j) = \begin{cases} 1 & \text{if } R(i,j) = 1 \text{ and } R(j,i) = 1 \\ 0 & \text{otherwise} \end{cases} .$$

It is evident that the can-reach-and-be-reached-by relation is reflexive, symmetric, and transitive. Thus, it is an equivalence relation, and we can partition the set of states into its equivalence classes. Graph theorists would refer to these equivalence classes as the *strong components* of the transition diagram.[4] But we will refer to them as *communication classes*, following the convention in the Markov chain literature.

Using Matlab, this matrix may be computed as follows.[5]

```
>> R'   % the can-be-reached-by matrix

ans =
     1     1     1     0     1
     0     1     0     0     0
     1     1     1     0     1
     0     1     0     1     0
     0     0     0     0     1

>> C = R & R'   % can-reach-AND-be-reached-by matrix

C =
     1     0     1     0     0
     0     1     0     0     0
     1     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
```

If we drew the graph representing the C matrix, the communication classes would appear as mutually exclusive cliques. That is, node i can reach and be reach by node j if and only if i and j belong to the same clique. Consequently, if states i and j belong to the same communication class, then

$$C(i,k) \;=\; C(j,k) \;\; \text{for all } k \in \{1,\ldots,n\}.$$

Equivalently, if i and j belong to the same communication class, then rows i and j of the C matrix are the same. To determine the membership of each communication class, we may thus list the unique rows of the C matrix.

```
>> U = unique(C,'rows')   % communication classes

U =
     0     0     0     0     1
     0     0     0     1     0
     0     1     0     0     0
     1     0     1     0     0
```

[4] In contrast, the *weak components* of a directed graph are given by equivalence classes of the can-reach-*or*-be-reached-by relation. For undirected graphs, the reachability relation is always symmetric. Hence, there is no distinction between strong components and weak components, and equivalence classes of the reachability relation are simply called *components*.

[5] But note that elementwise multiplication (`C = R .* R'`) would work just as well.

Each row of this matrix characterizes the membership of a different communication class. Thus, the communication classes are $\{\mathbf{5}\}$, $\{\mathbf{4}\}$, $\{\mathbf{2}\}$, and $\{\mathbf{1,3}\}$.

7.1.4 The reduced transition diagram

Having partitioned the states into communication classes, we may now draw a *reduced transition diagram* in which each communication class is "collapsed" into a single node. More precisely, letting $[i]$ denote the communication class containing state i, the reduced transition diagram includes an edge from class $[i]$ to class $[j]$ when there is a pair of states (i, j) such that $Z(i, j) = 1$ and $[i] \neq [j]$. For our present example, we thus obtain the reduced transition diagram shown below.

$$\begin{tikzcd}
\{\mathbf{4}\} & \{\mathbf{1}, \mathbf{3}\} \\
\{\mathbf{2}\} \arrow[u] \arrow[ur] & \{\mathbf{5}\} \arrow[u]
\end{tikzcd}$$

Intuitively, the reduced transition diagram will never include symmetric edges or cycles because this would imply that some communication classes should have been merged together.

The reduced transition diagram can be characterized by its *image matrix*. Using Matlab, we may compute this matrix as follows.

```
>> M = U * Z * U' > 0   % image matrix (with 1s on main diagonal)

M =
     1     0     0     1
     0     1     0     0
     0     1     1     1
     0     0     0     1

>> M = M & ~eye(4)   % removing 1s from main diagonal

M =
     0     0     0     1
     0     0     0     0
     0     1     0     1
     0     0     0     0
```

Intuitively, the `U` matrix relates communication classes to states, the `Z` matrix relates states to states, and the `U'` matrix relates states to communication classes. Thus, `U*Z*U'` is a square matrix relating communication classes to communication classes. Note that the ordering of these classes in the `M` matrix is determined by the ordering of rows in the `U` matrix (which was determined by Matlab).

7.1.5 Open and closed classes

Finally, using either the reduced transition diagram or its image matrix, we may determine whether each communication class is open or closed. A communication class $[i]$ is *open* when there is a directed edge from class $[i]$ to some class $[j]$ on the reduced transition diagram. Otherwise, class $[i]$ is *closed*. Equivalently, a communication class is closed when every entry of its row of the image matrix is 0. For our example, the classes $\{\mathbf{2}\}$ and $\{\mathbf{5}\}$ are open, while classes $\{\mathbf{1},\mathbf{3}\}$ and $\{\mathbf{4}\}$ are closed. Closed communication class may be understood as a generalization of the concept of an absorbing state introduced in Chapter 4. Any absorbing state constitutes its own communication class (because it cannot reach any other state). However, a Markov chain may be absorbed into a set of states (a closed communication class) even if no individual state in the class is absorbing.

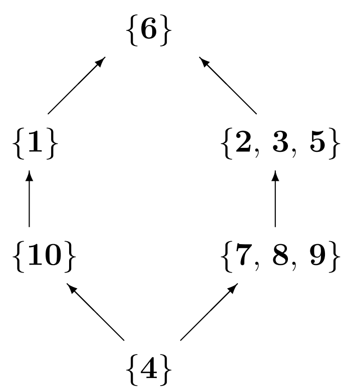
7.1.6 Another example

Here is a second (more elaborate) example. It proceeds from directly from a zero-pattern matrix through the same steps described above to arrive at the image matrix.

```
>> Z   % zero-pattern matrix

Z =
     0     0     0     0     0     1     0     0     0     0
     0     0     1     0     1     0     0     0     0     0
     0     1     0     0     1     1     0     0     0     0
     0     0     0     0     0     0     0     1     0     1
     0     1     1     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     1     0     0
     0     0     1     0     0     0     0     0     1     0
     0     0     0     0     0     0     1     0     0     0
     1     0     0     0     0     0     0     0     0     0

>> R = (eye(10) + Z)^9 > 0;   % reachability matrix

>> C = R & R';  % can-reach-and-be-reached-by matrix

>> U = unique(C, 'rows')   % communication classes

U =
     0     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     1     1     1     0
     0     0     0     0     0     1     0     0     0     0
     0     0     0     1     0     0     0     0     0     0
     0     1     1     0     1     0     0     0     0     0
     1     0     0     0     0     0     0     0     0     0

>> M = U * Z * U' > 0;
```

```
>> M = M & ~eye(6)   % image matrix

M =
     0     0     0     0     0     1
     0     0     0     0     1     0
     0     0     0     0     0     0
     1     1     0     0     0     0
     0     0     1     0     0     0
     0     0     1     0     0     0
```

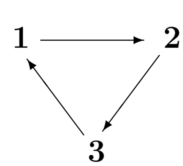
Note that Matlab has listed the communication classes in the order $\{\mathbf{10}\}$, $\{\mathbf{7}, \mathbf{8}, \mathbf{9}\}$, $\{\mathbf{6}\}$, $\{\mathbf{4}\}$, $\{\mathbf{2}, \mathbf{3}, \mathbf{5}\}$, and $\{\mathbf{1}\}$. Thus, the reduced transition diagram is depicted below.



It is apparent that class $\{\mathbf{6}\}$ is closed while all other classes are open.

7.2 Irreducibility and Centeredness

The preceding discussion of communication classes leads immediately to several definitions that are useful for understanding the long-run behavior of Markov chains and linear systems more generally. Given any square, non-negative matrix A (not necessarily a transition matrix), we may use its zero pattern to determine its communication classes. The matrix A is *irreducibile* when every state belongs to the same communication class. Equivalently, A is irreducibile when every state can reach every other state. As you might expect, any matrix that is not irreducible is said to be *reducible*. Although every primitive matrix is irreducible, some irreducible matrices are not primitive. For instance, consider a Markov chain with the following zero-pattern transition diagram.



It is apparent that this chain has a single communication class, and hence the transition matrix is irreducible. Alternatively, irreducibility can also be verified by computing the reachability matrix.

```
>> P1   % an irreducible matrix

P1 =
     0     1     0
     0     0     1
     1     0     0

>> (eye(3) + P1)^2 > 0   % reachability

ans =
     1     1     1
     1     1     1
     1     1     1
```

However, iterated multiplication of the transition matrix yields

```
>> P1^2

ans =
     0     0     1
     1     0     0
     0     1     0

>> P1^3

ans =
     1     0     0
     0     1     0
     0     0     1

>> P1^4

ans =
     0     1     0
     0     0     1
     1     0     0
```

and it becomes apparent that the zeros will never "fill in" regardless of the exponent. Thus, the transition matrix is not primitive. As indicated by the transition diagram, this Markov chain perpetually "cycles" from state 1 to 2 to 3 to 1 to 2 to 3 and so on. For this reason, any matrix that is irreducible but not primitive is called a *cyclic* matrix.

To offer one more definition: a square, non-negative matrix A is *centered* when it has only one closed class and the submatrix for this class is primitive. Every primitive matrix is centered because the "submatrix" for its sole communication class is the entire (primitive) matrix. However, some centered matrices are not primitive. For instance, consider the following transition matrix.

```
>> P2  % a centered matrix that is not primitive

P2 =
    0.8000    0.2000         0
         0         0    1.0000
         0    0.5000    0.5000
```

From the zero-pattern transition diagram,

$$\begin{tikzcd} \mathbf{1} \arrow[loop left] \arrow[r] & \mathbf{2} \arrow[r, leftrightarrow] & \mathbf{3} \arrow[loop right] \end{tikzcd}$$

it is easy to see that this matrix has two communication classes, with the reduced transition diagram given below.

$$\{\mathbf{1}\} \longrightarrow \{\mathbf{2}, \mathbf{3}\}$$

Because the process cannot transition from state 2 (or 3) to state 1 in any number of steps, this transition matrix is not irreducible and hence not primitive. However, it is easy to see that the submatrix for the closed class is primitive.

```
>> P2(2:3,2:3)   % submatrix for class {2,3}

ans =
         0    1.0000
    0.5000    0.5000

>> P2(2:3,2:3)^2   % to demonstrate primitivity

ans =

    0.5000    0.5000
    0.2500    0.7500
```

Before proceeding, perhaps it is useful to quickly review the connections between the definitions just introduced. Matrices can be reducible or irreducible. Reducible matrices can be centered (like `P2`) or not centered. Irreducibile matrices can be primitive or cyclic. Primitive matrices are always centered, while cyclic matrices (like `P1`) are never centered.

7.2.1 Implications for convergence

In Chapter 1, we saw that a Markov chain will reach a unique long-run equilibrium regardless of its initial condition if the transition matrix is primitive. More precisely, given the dynamics $\mathbf{x}_t = \mathbf{x}_0 P^t$, the probability distribution $\mathbf{x}_t$ converges to the unique limiting distribution $\mathbf{x}$ determined by the condition $\mathbf{x} = \mathbf{x}P$. Because a closed communication class acts as an "absorbing set" of states, it is easy see that this convergence result will also hold under the weaker condition that P is centered. Intuitively, because the chain must ultimately be absorbed into the sole closed class, the long-run probabilities will be zero for all states in any other classes. The probability distribution for states within the closed class can thus be determined using the (primitive) submatrix for this class. For the preceding example,

```
>> P2(2:3,2:3)^100   % long-run outcome within closed class

ans =
    0.3333    0.6667
    0.3333    0.6667

>> P2^100   % long-run outcome for all states

ans =
    0.0000    0.3333    0.6667
         0    0.3333    0.6667
         0    0.3333    0.6667
```

Consequently, for any initial condition, we obtain the limiting distribution given by any row of the preceding matrix.

In contrast, convergence is not guaranteed if the transition matrix is cyclic. To illustrate, consider the cyclic transition matrix given above, along with the following initial condition. Projecting ahead for several periods, it becomes obvious that the distribution will continue to cycle forever, never converging to a limiting distribution.

```
>> x0 = [.1 .3 .6]   % initial condition

x0 =
    0.1000    0.3000    0.6000

>> for t = 0:10; disp(x0*P1^t); end

    0.1000    0.3000    0.6000
    0.6000    0.1000    0.3000
    0.3000    0.6000    0.1000
    0.1000    0.3000    0.6000
    0.6000    0.1000    0.3000
    0.3000    0.6000    0.1000
    0.1000    0.3000    0.6000
    0.6000    0.1000    0.3000
```

```
    0.3000    0.6000    0.1000
    0.1000    0.3000    0.6000
    0.6000    0.1000    0.3000
```

While easily understood in the context of the preceding example, non-convergence might seem more surprising given that there is a version of Perron-Frobenius Theorem for irreducible matrices which is quite similar to the version that we encountered in Chapter 3.

Perron-Frobenius Theorem (for irreducible matrices). *If A is a non-negative, irreducible matrix then (i) one of its eigenvalues is positive and greater than or equal to (in absolute value) all other eigenvalues, and (ii) there is a positive eigenvector corresponding to that eigenvalue.*

Carefully comparing the two versions of theorem, note the addition of the crucial phrase "or equal to" in point (i). Thus, while primitivity of A implies there is one eigenvalue that is strictly larger than all others, cyclic matrices may have several eigenvalues that share the largest absolute value. To illustrate the consequences, consider the eigenvectors and eigenvalues of the `P1` matrix.

```
>> [eigvec, eigval] = eig(P1')

eigvec =
  -0.2887 - 0.5000i  -0.2887 + 0.5000i  -0.5774
  -0.2887 + 0.5000i  -0.2887 - 0.5000i  -0.5774
   0.5774             0.5774            -0.5774

eigval =
  -0.5000 + 0.8660i        0                  0
        0            -0.5000 - 0.8660i        0
        0                  0             1.0000

>> abs(diag(eigval))'

ans =

    1.0000    1.0000    1.0000
```

Computing the absolute value of each of these eigenvalues, we find that all three have the same absolute value. Because no one eigenvalue "dominates" all others, the distribution does not simply converge to the corresponding eigenvector, but demonstrates the more complicated (cyclic) behavior we saw above.

7.3 Reducibility of simultaneous equation systems

Given the conventional rendering of a transition diagram, the directed edges reflect possible transitions between states over time. That is, by following the arrows, we can visualize the transitions that might occur from one period to the next. While this convention seems quite natural, it is sometimes useful to reverse the direction of the edges of the transition diagram. Formally, given a zero-pattern matrix Z, its transition diagram contains an edge from i to j when $Z(i,j) = 1$, while its *influence diagram* contains an edge from i to j when $Z(j,i) = 1$. Put differently, the influence diagram for Z is the transition diagram for the *transpose* of Z. Influence diagrams are especially helpful for visualizing interdependence – whether one variable is "influenced by" another variable – in a simultaneous equation system. To illustrate, consider again the centered transition matrix given above. The dynamics for the Markov chain process are given by the matrix equation

$$\mathbf{x}_{t+1} \;=\; \mathbf{x}_t P$$

which (given the coefficients in the `P2` matrix) may be rewritten as

$$\begin{aligned}
\mathbf{x}_{t+1}(1) &= 0.8\ \mathbf{x}_t(1) \\
\mathbf{x}_{t+1}(2) &= 0.2\ \mathbf{x}_t(1) + 0.5\ \mathbf{x}_t(3) \\
\mathbf{x}_{t+1}(3) &= \mathbf{x}_t(2) + 0.5\ \mathbf{x}_t(3)
\end{aligned}$$

Based on these equations, we might now construct a directed graph, including an edge from node i to node j when $\mathbf{x}_{t+1}(i)$ is a function of $\mathbf{x}_t(j)$, as shown below.

$$\begin{tikzcd}
\mathbf{1} \arrow[loop left] & \mathbf{2} \arrow[l] \arrow[r, leftrightarrow] & \mathbf{3} \arrow[loop right]
\end{tikzcd}$$

Of course, this graph is simply the influence diagram for the zero pattern of `P2` (i.e., the transition diagram for the zero pattern of the transpose of `P2`). Pushing further, we might use the preceding diagram to determine its communication classes, obtaining the reduced influence diagram below.

$$\{\mathbf{1}\} \longleftarrow \{\mathbf{2},\mathbf{3}\}$$

This diagram simply reverses the direction of the edge on the reduced transition diagram given in section 7.2. More generally, transposition of a zero-pattern matrix never affects its communication classes, but merely reverses the direction of any edges between classes. Thus, for any zero-pattern matrix, we can obtain its reduced influence diagram simply by reversing the edges of its reduced transition diagram.[6]

Given this motivation for the reduced influence diagram, the mathematical rationale for the terms "irreducible" and "reducible" becomes more apparent. If this diagram reveals a single communication class, every variable depends (directly or

[6]To put this differently, if the reduced transition matrix is characterized by image matrix M, then the reduced influence matrix is characterized by the transpose of M.

indirectly) on every other variable, and the entire set of equations must be solved together. Hence, the problem (of solving the simultaneous-equation system) is "irreducible." But if there is more than one communication class, it is possible to "reduce" the overall problem to a series of (simpler) subproblems. In particular, we may begin by solving the set of equations corresponding to some closed class (because the variables in this set are not influenced by variables outside this set). And once we have solved for the variables within each closed set, we may then proceed sequentially to solve for variables in the open sets.

To illustrate, let's solve for the long-run distribution for our present example, determined by the following system of equations,

$$\begin{aligned}\mathbf{x}(1) &= 0.8\,\mathbf{x}(1)\\ \mathbf{x}(2) &= 0.2\,\mathbf{x}(1) + 0.5\,\mathbf{x}(3)\\ \mathbf{x}(3) &= \mathbf{x}(2) + 0.5\,\mathbf{x}(3)\end{aligned}$$

along with the requirement that the long-run distribution is a probability vector,

$$\mathbf{x}(1) + \mathbf{x}(2) + \mathbf{x}(3) = 1$$

From the reduced influence diagram, we see that communication class $\{\mathbf{1}\}$ is closed. We can thus solve that (one-equation) system first, obtaining $\mathbf{x}(1) = 0$. Given this solution, we may now consider the remaining (two-equation) system,

$$\begin{aligned}\mathbf{x}(2) &= 0.5\,\mathbf{x}(3)\\ \mathbf{x}(3) &= \mathbf{x}(2) + 0.5\,\mathbf{x}(3)\end{aligned}$$

which, along with the requirement

$$\mathbf{x}(2) + \mathbf{x}(3) = 1$$

leads to the result $\mathbf{x}(2) = 1/3$ and $\mathbf{x}(3) = 2/3$ already obtained in section 7.2. The following chapters on influence networks and demography will offer further illustrations of reducibile simultaneous-equation systems.

7.4 Further Reading

For the classic discussion of equivalence classes in communication networks, see Kemeny, Snell, and Thompson, *Introduction to Finite Mathematics*, Prentice-Hall, 1966. The standard reference for social network analysis is Wasserman and Faust, *Social Network Analysis: Methods and Applications*, Cambridge, 1994. The standard reference for graph theory is Harary, Norman, and Cartwright, *Structural Models: An Introduction to the Theory of Directed Graphs*, Wiley, 1965. The term "centered" seems to have been coined by Friedkin and Johnsen in their 1997 *Social Networks* paper, which we will address more directly in the next chapter. For further discussion and application of influence diagrams in a more general linear-systems framework, see Farina and Rinaldi, *Positive Linear Systems: Theory and Applications*, Wiley, 2000.