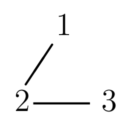
last revised: March 2008

WARNING for Soc 376 students: This draft adopts the demography convention for transition matrices (i.e., transitions from column to row).

6 Evolution of Networks

6.1 Strategic network formation

We begin by introducing some terminology and notation used in the economics literature on network formation. A social tie ("link") between actors i and j is denoted by ij. In this chapter, links are always undirected, and thus ij is equivalent to ji. A social network ("graph") can be described by the set of links between actors. We let g denote a particular graph, and let G denote the set of all possible graphs on the set of actors N. To illustrate, the graph $g = \{12, 23\}$ on the set of actors $N = \{1, 2, 3\}$ is depicted below.



Given 3 actors, there are 8 possible graphs. More precisely,

$$G = \{\emptyset, \{12\}, \{13\}, \{23\}, \{12, 13\}, \{12, 23\}, \{13, 23\}, \{12, 13, 23\}\}$$

where $\emptyset$ denotes the empty graph (which contains no links).

6.1.1 Connections Model

A crucial component of any strategic network formation model is the *value function* which describes, for each possible graph, the payoff received by each actor. Let $u_i(g)$ denote the payoff received by actor $i \in N$ given graph $g \in G$. To illustrate, we consider the "connections model" (introduced by Jackson and Wolinsky 1996). In this model, the value function is given by

$$u_i(g) = -c\ d_i + \sum_{j \neq i} \delta^{\tau_{ij}}$$

where $c \geq 0$ is the cost of maintaining a link (borne by both linked actors), d_i is the number of links held by actor i, and τ_{ij} is the length of the shortest path from actor

i to actor j in graph g.[1] Intuitively, the parameter $\delta \in (0, 1)$ reflects "information decay." Thus, in the connections model, actors face a tradeoff between "high-quality" information through direct links (at cost c) and "lower quality" information through indirect links (at no cost).

6.1.2 Pairwise stability

While the value function captures the incentives of actors to form or break links, we also need an *equilibrium concept* to determine whether a particular graph is stable.[2] Here, we adopt here the concept of pairwise stability (also due to Jackson and Wolinsky 1996). Formally, a graph g is *pairwise stable* when

$$\begin{array}{ll} \text{(i) for all } ij \in g, & u_i(g) \geq u_i(g - ij) \text{ and } u_j(g) \geq u_j(g - ij) \\ \text{(ii) for all } ij \notin g, & u_i(g) \geq u_i(g + ij) \;\text{ or }\; u_j(g) \geq u_j(g + ij) \end{array}$$

where $g + ij$ denotes graph g with link ij added, and $g - ij$ denotes graph g with link ij removed. In words, graph g is pairwise stable when

(i) for any link ij in g, both i and j prefer not to remove the link
(ii) for any link ij not in g, either i or j prefers not to add the link

Thus, the pairwise stability concept presumes that actors can unilaterally break links, while pairs of actors can add links.

Reflection might suggest several shortcomings of pairwise stability as an equilibrium concept. For instance, given graph g, an actor might be able increase her payoff by breaking several links simultaneously. But if the removal of any one link would decrease her payoff, then g may still be pairwise stable. Alternatively, given graph g, two actors might prefer to form a new link between themselves if they can also sever some existing links to third parties. But if these actors would not prefer to form the new link with the existing links in place, then g may again be pairwise stable.

On the other hand, pairwise stability might be understood as a necessary (if not sufficient) condition for "true" stability. That is, the pairwise stability condition might lead us to judge too many graphs as stable. But if a graph does not pass the pairwise stability test, we can be sure that it is not "truly" stable. Even more importantly, the pairwise stability concept leads naturally to a process model of network evolution that we develop in the next section.

[1] Adopting graph-theoretic terminology, d_i is the *degree* of node i, the shortest path between a pair of nodes is called a *geodesic*, and τ_{ij} is the *distance* between nodes i and j.

[2] Recalling our discussion of the Nash equilibrium condition in Chapter 5, "stability" should be understood here in the game-theoretic sense of "robust to deviations by self-interested actors." The next section will bring together this (non-process) conception of stability with the (process) conception of stability within Markov chain models.

6.1.3 Pairwise stability in the connections model

To illustrate the pairwise stability condition, we return to the connections model, fixing the set of actors $N = \{1, 2, 3, 4, 5\}$ and the parameter values $\delta = 0.5$ and $c = 0.55$. Consider the graph $g = \{12, 13, 14, 15\}$. It is easy to see that this "star" graph is not pairwise stable. The payoff for actor 1 (who is the center of the star) is

$$u_1(g) = 4(d_1 - c) = -0.2$$

But severing any of her existing links (say link 12), actor 1's payoff rises to

$$u_1(g - 12) = 3(d_1 - c) = -0.15$$

and thus the graph g is not pairwise stable.[3] Intuitively, given our maintained assumption that $c > \delta$, the cost of maintaining a link exceeds the benefits of a direct connection, and thus no individual would be willing to serve as the center of a star graph.[4]

In contrast, the graph $g = \{12, 23, 34, 45, 51\}$ is pairwise stable for the assumed parameter values. In this "cycle" graph, all actors hold similar positions, and payoffs are

$$u_i(g) = 2\delta + 2\delta^2 - 2c = 0.4$$

for all $i \in N$. To establish the pairwise stability of this graph, we must first show that no actor would want to sever an existing link. For instance, suppose link 12 was removed from g. This implies

$$u_1(g - 12) = u_2(g - 12) = \delta + \delta^2 + \delta^3 + \delta^4 - c = 0.3875$$

and hence neither actor 1 or 2 would prefer to break this link. Given that the removal of any other edge ij in g has the same consequence for actors i and j, we see that no actor wishes to sever an existing link. Second, we must show that no pair of actors would want to add a link. For instance, suppose that link 13 was added to g. This implies

$$u_1(g + 13) = u_3(g + 13) = 3\delta + \delta^2 - 3c = 0.1$$

and hence both actors 1 and 3 would prefer not to add this link. Given that the addition of any new link ij to the cycle graph would have the same consequences for actors i and j, we see that no pair of actors would prefer to add a new link. Thus, the graph g is pairwise stable.

[3] Note that a single violation of the pairwise stability condition is sufficient to demonstrate the negative result that g is not pairwise stable. As we'll see in the next paragraph, it takes more work to establish the positive result that a graph is pairwise stable.

[4] This result might suggest the possibility for "side payments" from actors on the periphery of the star to the actor in the center. See Jackson and Wolinsky (1996) for development of the connections model with transferrable utility.

Moving beyond these two graphs, Jackson and Wolinsky (1996) provide the following characterization of pairwise stable graphs in the connections model. For $c < \delta - \delta^2$, the complete graph (with links between every pair of actors) is the unique pairwise stable network. For $c \in [\delta - \delta^2, \delta]$, the star graph is pairwise stable, but is not always the unique pairwise stable graph. Finally, for $c > \delta$, the star graph is not pairwise stable. Intuitively, as c becomes very large, no actor is ever willing to maintain links, and only the empty graph is pairwise stable.

6.2 A process model

Like the standard game-theoretic analysis of coordination games discussed in Chapter 5, the preceding game-theoretic analysis of network stability is essentially static. However, the pairwise stability condition naturally suggests a Markov chain process model of network evolution. In this model, like the bystander model discussed in Chapter 4, graphs are viewed as the states of the process. Thus, G is the set of all states, and each $g \in G$ is a particular state. Letting g_t denote the state of the chain in period t, the process may be summarized as follows.

(1) given graph g_t

- (a) randomly choose some pair of actors (i, j)
- (b) if (g_t does not contain link ij) and (both i and j prefer to add link) then $g_{t+1} = g_t + ij$
- (c) elseif (g_t contains link ij) and (either i or j prefers to remove link) then $g_{t+1} = g_t - ij$
- (d) else $g_{t+1} = g_t$

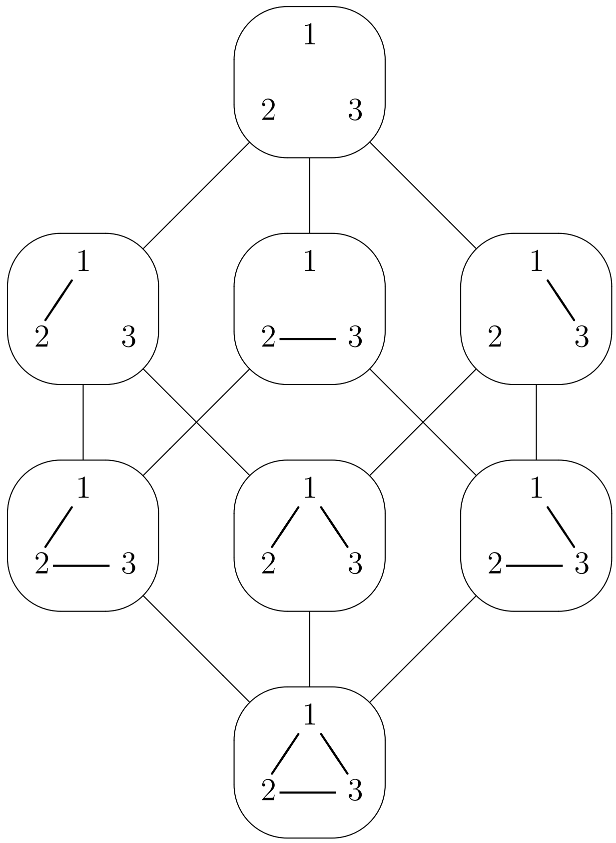
(2) increment t and repeat

Starting from some initial state g_0, the chain $(g_0, g_1, g_2, \ldots)$ will eventually converge to an absorbing state, or perhaps a set of states comprising a closed communication class.[5] If we interpret points (b) and (c) to require at least one of the actors (i or j) to *strictly* prefer any change in the state, then state g is absorbing if and only if g is a pairwise stable graph. In this way, pairwise stability can be understood as "stability" in both a game-theoretic and stochastic-process sense.

Following Jackson and Watts (2002), we say that two states g and g' are *adjacent* when $g' = g + ij$ or $g' = g - ij$ for some ij. To illustrate, Figure 1 depicts adjacencies between all states $g \in G$ for $N = \{1, 2, 3\}$. Note that the edges on this diagram are undirected due to the symmetry of the adjacency relation. Given a value function, we could further assign a direction to each edge, with an arrow from state g to $g + ij$ if both actors i and j prefer to maintain link ij, and an arrow in the reverse direction

[5] See Jackson and Watts (2002) for an example illustrating the latter possibility.

Figure 1: Adjacent states $g \in G$ for $N = \{1, 2, 3\}$



if either actor i or j does not prefer to maintain this link.[6] In this way, we obtain (except for the loops) the transition diagram for the zero pattern of the transition matrix. Jackson and Watts (2002) refer to the paths in this diagram as *improving paths* because they reflect improvements in the payoffs of the actors involved the transitions.

6.2.1 Network evolution in the connections model

To illustrate further, we return again to the connections model. If we fix the set of actors $N = \{1, 2, 3, 4\}$, there are 6 possible links $\{12, 13, 14, 23, 24, 34\}$, and thus G contains $2^6 = 64$ different graphs. It might thus appear that our process model must have 64 states. However, because we have specified an "anonymous" version of the

[6]If we require at least one of the actors to strictly prefer a change in the state, then it is possible that transitions occur in neither direction (when both actors are indifferent between the pair of states), but transitions can never occur in both directions.

connections model (in which the identities of the actors do not affect payoffs), we may "collapse" these 64 states into the 11 states shown in Figure 2.[7] This diagram also indicates the direction of possible transitions between each pair of adjacent states under the assumption that

$$\delta - \delta^2 < c < \delta - \delta^3$$

Intuitively, $\delta - \delta^\tau$ is the increase in "information" generated by replacing an indirect (length τ) path by a direct (length 1) path. Thus, Figure 2 reflects the assumption that actor i is willing to add a direct link to actor j if and only if the distance from i to j is greater than 2.

While somewhat tedious, it is not difficult to verify the direction of each the transitions indicated in Figure 2. For instance, when the process is in state 1 (the empty graph), all actors receive payoff 0. Adding a link between any pair of actors, so that the process is now in state 2, the payoff of both actors rises to $\delta - c$. Thus, both actors would prefer to add this link. This same computation also implies that neither of the linked actors in state 2 would wish to sever the link. Thus, in Figure 2, transitions occur from state 1 to 2 (and not from state 2 to 1). To give one more example, suppose the process is currently in state 2. If one of the connected actors (let's call him i) and one of the unconnected actors (let's call her j) form a new link (ij), the process would transition to state 3. This transition causes i's payoff to rise from $\delta - c$ to $2(\delta - c)$, while j's payoff rises from 0 to $\delta + \delta^2 - c$. Thus, transition occur from state 2 to 3 (but not from state 3 to 2). The direction of every other transition indicated on this diagram can be verified in similar fashion.

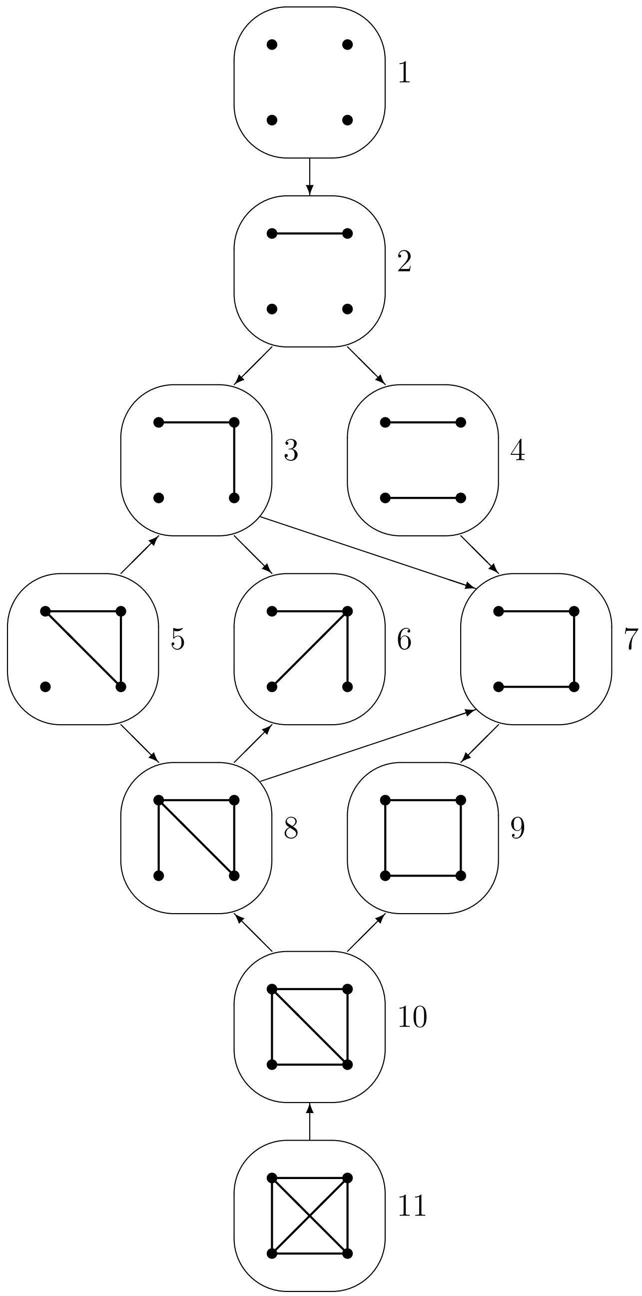
Given the direction of each transition, a state g is absorbing when no edges leave g. Thus, inspection of Figure 2 reveals that state 6 (star) and state 9 (cycle) are absorbing. Having fixed the parameter values so that $\delta - \delta^2 < c < \delta - \delta^3$, it may be interesting to briefly consider the long-run outcome(s) under other scenarios. For instance, given $\delta - \delta^3 < c < \delta$, the reader may verify that the edge from state 7 to state 9 reverses direction, while all other edges retain the same direction shown on Figure 2. Thus, if the cost parameter c lies in that (higher) range, the absorbing states of the process are state 6 and state 7. But given the general result from Jackson and Wolinsky (1996) stated above, we know that state 6 is not absorbing if we increase the cost parameter even further, into the range where $c > \delta$.

6.2.2 Transition probabilities

Returning to the case where $\delta - \delta^2 < c < \delta - \delta^3$, we now assign precise transition probabilities to each of the directed edges in Figure 2. To begin, recall that one pair of actors is drawn randomly in each time period, and that (given 4 actors) there are 6 pairs of actors from which to choose. For each state g, we may thus determine the number of pairs x (between 0 and 6) associated with each directed edge leaving g,

[7] Formally, these 11 states are equivalence classes of isomorphic graphs. Recall the similar tactic adopted in our analysis of the bystander model in Chapter 4.

Figure 2: Connections model with 4 actors and $\delta - \delta^2 < c < \delta - \delta^3$



and then assign probability $x/6$ to this edge. If the sum of the probabilities over all edges leaving g is less than 1, this indicates that some pair(s) would not prefer to leave g, and the residual probability is assigned to an (undepicted) loop at g.

To illustrate, consider state 2. The directed edge from state 2 to 3 is associated with 4 pairs (those with one connected actor and one unconnected actor), while the directed edge from state 2 to 4 is associated with the one pair in which both actors are unconnected. Thus, we assign probability 2/3 to the edge from state 2 to 4, and probability 1/6 to the edge from state 2 to 4. The residual probability is associated with the one pair in which both actors are connected. Thus, we assign probability 1/6 to the undepicted loop reflecting a transition from state 2 to itself. Proceeding in this same manner for each of the 11 states, we obtain the transition matrix below.

```
>> P = connectionsmodel(0)   % input 0 indicates no mistakes
P =
         0         0         0         0         0         0         0         0         0         0         0
    1.0000    0.1667         0         0         0         0         0         0         0         0         0
         0    0.6667    0.5000         0    0.5000         0         0         0         0         0         0
         0    0.1667         0    0.3333         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0
         0         0    0.1667         0         0    1.0000         0    0.1667         0         0         0
         0         0    0.3333    0.6667         0         0    0.8333    0.3333         0         0         0
         0         0         0         0    0.5000         0         0    0.5000         0    0.6667         0
         0         0         0         0         0         0    0.1667         0    1.0000    0.1667         0
         0         0         0         0         0         0         0         0         0    0.1667    1.0000
         0         0         0         0         0         0         0         0         0         0         0
```

The m-file `connectionsmodel` will be discussed further in the next section. For now, given this transition matrix, we can easily determine the long-run outcome of the Markov chain process by raising it to a sufficiently high power.

```
>> P^10000   % long-run outcome
ans =
         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0
    0.2667    0.2667    0.3333         0    0.3333    1.0000         0    0.3333         0    0.2667    0.2667
         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0
    0.7333    0.7333    0.6667    1.0000    0.6667         0    1.0000    0.6667    1.0000    0.7333    0.7333
         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0
```

We already knew from inspection of Figure 2 that states 6 are 9 are absorbing, and that there is zero probability the process will occupy any other state in the long run. But the preceding computation gives the precise probability of absorption into state 6 or 9 from each non-absorbing state.

6.3 Introducing mistakes into the process model

In the preceding chapter, we introduced "mistakes" into the evolution of social conventions both to capture the bounded rationality of actors and to permit analysis of stochastic stability. Jackson and Watts (2002) have taken this same approach in their analysis of network evolution. Generalizing the process model so that mistakes occur with probability $\epsilon \geq 0$, we obtain the modified version outlined below.

(1) given graph g_t

  (a) randomly choose some pair of actors (i, j)

  (b) if (g_t does not contain ij) and (i and j prefer link)
  then $g_{t+1} = \begin{cases} g_t + ij \text{ with probability } 1-\epsilon \\ g_t \text{ with probability } \epsilon \end{cases}$

  (c) elseif (g_t does not contain ij) and (i or j does not prefer link)
  then $g_{t+1} = \begin{cases} g_t \text{ with probability } 1-\epsilon \\ g_t + ij \text{ with probability } \epsilon \end{cases}$

  (d) elseif (g_t contains ij) and (i and j prefer link)
  then $g_{t+1} = \begin{cases} g_t \text{ with probability } 1-\epsilon \\ g_t - ij \text{ with probability } \epsilon \end{cases}$

  (e) elseif (g_t contains ij) and (i or j does not prefer link)
  then $g_{t+1} = \begin{cases} g_t - ij \text{ with probability } 1-\epsilon \\ g_t \text{ with probability } \epsilon \end{cases}$

(2) increment t and repeat

Thus, a "mistake" may reflect the addition (or retention) of a link that was not desired by at least one actor, or it may reflect the failure to form (or maintain) a link that was desired by both actors.

6.3.1 Mistakes in the connections model

To see how mistakes affect the network-formation process, we proceed directly to the specification of the transition matrix $\mathbf{P}$ for the 11-state connections model. Following the same strategy as in the preceding section, we may again determine the transition probabilities from each state g (that is, each column g of the $\mathbf{P}$ matrix) by considering in turn each of the 6 pairs of actors that could be drawn randomly. Conditional on being drawn (with probability 1/6), each pair makes the desired transition with probability $1 - \epsilon$, and the alternative (mistaken) transition with probability ϵ.

To illustrate, consider state 2 once again. For the four pairs composed of one connected and one unconnected actor, both actors would prefer to form a link, creating a transition from state 2 to 3 with probability $\mathbf{P}(3, 2) = (4/6)(1 - \epsilon)$. For the one pair in which both actors are unconnected, both actors would prefer to form a link, creating a transition from state 2 to 4 with probability $\mathbf{P}(4,2) = (1/6)(1-\epsilon)$. Finally, for the one pair already linked, both actors prefer to maintain this link. But given the possibility of mistakes, this link may be severed anyway, and we obtain a transition from state 2 to 1 with probability $\mathbf{P}(1,2) = \epsilon$. The residual probability is assigned to the transition from state 2 to itself, so that $\mathbf{P}(2,2) = (1/6) + (4/6)\epsilon$.

Proceeding in this manner for all 11 states, we obtain the full transition matrix, given by the m-file `connectionsmodel` in the Appendix. Setting the probability of mistakes at $\epsilon = 0.01$, we obtain the transition matrix below.

```
>> P = connectionsmodel(.01)   % mistakes occur with probability .01
P =
    0.0100    0.0017         0         0         0         0         0         0         0         0         0
    0.9900    0.1733    0.0033    0.0033         0         0         0         0         0         0         0
         0    0.6600    0.5000         0    0.4950    0.0050    0.0033         0         0         0         0
         0    0.1650         0    0.3367         0         0    0.0017         0         0         0         0
         0         0    0.0017         0    0.0100         0         0    0.0017         0         0         0
         0         0    0.1650         0         0    0.9900         0    0.1650         0         0         0
         0         0    0.3300    0.6600         0         0    0.8267    0.3300    0.0067         0         0
         0         0         0         0    0.4950    0.0050    0.0033    0.5000         0    0.6600         0
         0         0         0         0         0         0    0.1650         0    0.9900    0.1650         0
         0         0         0         0         0         0         0    0.0033    0.0033    0.1733    0.9900
         0         0         0         0         0         0         0         0         0    0.0017    0.0100
```

Given the possibility of mistakes, every state can reach (directly or indirectly) every other state, and hence the transition matrix is irreducible. The positive probabilities along the main diagonal of $\mathbf{P}$ (loops on the transition diagram) further ensure that $\mathbf{P}$ is primitive. Thus, we know that the probability distribution over states will converge to the unique probability vector $\mathbf{x}$ determined by the equation $\mathbf{x} = \mathbf{P}\mathbf{x}$. To find this limiting distribution, we may raise the transition matrix to a sufficiently high power.

```
>> P^10000
ans =
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0016    0.0016    0.0016    0.0016    0.0016    0.0016    0.0016    0.0016    0.0016    0.0016    0.0016
    0.0001    0.0001    0.0001    0.0001    0.0001    0.0001    0.0001    0.0001    0.0001    0.0001    0.0001
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.1243    0.1243    0.1243    0.1243    0.1243    0.1243    0.1243    0.1243    0.1243    0.1243    0.1243
    0.0463    0.0463    0.0463    0.0463    0.0463    0.0463    0.0463    0.0463    0.0463    0.0463    0.0463
    0.0060    0.0060    0.0060    0.0060    0.0060    0.0060    0.0060    0.0060    0.0060    0.0060    0.0060
    0.8184    0.8184    0.8184    0.8184    0.8184    0.8184    0.8184    0.8184    0.8184    0.8184    0.8184
    0.0033    0.0033    0.0033    0.0033    0.0033    0.0033    0.0033    0.0033    0.0033    0.0033    0.0033
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
```

Thus, in the long run, the chain occupies state 9 about 81.84% of the time and state 6 about 12.43% of the time. While the Perron-Frobenius Theorem guarantees that all of the other probabilities are strictly positive, we see that some are very small (less than 0.0001).

To determine whether both states 6 and 9 are stochastically stable, we decrease the probability of mistakes even further (to, say, $\epsilon = 10^{-6}$). Because ϵ remains strictly positive, we know that the probability distribution over states must again converge in the long run. However, because ϵ is now very small, we find that the current state of the chain remains dependent on the initial state even after 10,000 periods. A much larger number of periods (closer to 10^8) is required to find the limiting distribution.

```
>> P = connectionsmodel(10^-6);   % mistakes occur with probability .000001

>> P^10000   % after 10^4 periods
ans =
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.2655    0.2655    0.3317    0.0009    0.3317    0.9934    0.0009    0.3317    0.0009    0.2655    0.2655
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.7344    0.7344    0.6683    0.9991    0.6683    0.0066    0.9991    0.6683    0.9991    0.7344    0.7344
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

>> ans^10000   % after 10^8 periods
ans =
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.1176    0.1176    0.1176    0.1176    0.1176    0.1176    0.1176    0.1176    0.1176    0.1176    0.1176
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.8823    0.8823    0.8823    0.8823    0.8823    0.8823    0.8823    0.8823    0.8823    0.8823    0.8823
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
```

In the long run, the chain occupies state 6 about 12% of the time, and state 9 about 88% of the time. Because both of the long-run probabilities are strictly positive, it appears that both states are stochastically stable.

6.3.2 A graphical method for determining stochastic stability

For the present example, it is also possible (perhaps even easier) to determine stochastic stability directly from Figure 2. In the absence of mistakes, the process would follow an improving path, eventually reaching one of the absorbing states. If a mistake occurs, this mistake may be viewed as a "backward" transition because the process moves contrary to the direction indicated by some arrow. For instance, if the process is currently in state 9 and a new link is added (against the preference of at least one actor involved), this mistake constitutes a backward transition from state 9 to 10. Figure 2 reveals that, if this mistake did occur, the process could then transition from state 10 to 8 to 6 via an improving path. Thus, only one mistake – one backward transition – is required for the process to be "bumped" from state 9 to state 6. Now suppose the process is currently in state 6. If one of the links is severed (against the preferences of both actors involved), this mistake constitutes a backward transition from state 6 to 3. If this mistake did occur, the process could then transition from state 3 to 7 to 9 via an improving path. Thus, we find that one backward transition is also required for the process to be "bumped" from state 6 to 9.[8] Because the same number of mistakes is required to "bump" the process in either direction, both states 6 and 9 are stochatically stable.

To proceed more formally, we may define the *resistance* of state g as the smallest number of arrows that must be reversed on the zero-pattern transition diagram (for the mistake-free process) such that g can be reached (directly or indirectly) by every other state $g' \in G$. Given this definition, the following result was established by Jackson and Watts (2002). Letting $r(g)$ denote the resistance of state g, state g is stochastically stable when

$$r(g) = \min_{g' \in G} r(g')$$

That is, g is stochastically stable when no other state g' has strictly lower resistance. For our example, states 6 and 9 are stochastically stable because $r(6) = r(9) = 1$ while every other state has strictly higher resistance.[9] Of course, for other examples, the minimum resistance level may be greater than 1, implying that a longer chain of mistakes is necessary to "bump" the process away from each stochastically stable state. Moreover, not every absorbing state will be stochastically stable.[10]

6.4 Further reading

Jackson and Wolinsky (J Econ Theory 1996) developed the concept of pairwise stability and introduced the connections model. Jackson and Watts (J Econ Theory 2002) developed the evolutionary approach.

[8]Note that the backward transition might also have occurred from state 6 to 8. The process could then have followed an improving path from state 8 to 7 to 9. Either of the paths (6,3,7,9) or (6,8,7,9) involves a single backward transition.

[9]It is easy to see that state g is stochastically stable only if g is contained within a closed communication class. Any state g' contained in an open class must lie on an improving path to some state g in a closed class. Once we have reversed the minimum number of arrows such that all states can reach g', we can "unreverse" the arrows along this improving path to establish that $r(g)$ must be less than $r(g')$.

[10]In models with only two absorbing states, g and g', graphical determination of stochastic stability is relatively straightforward, since we would need merely to compute the pairwise resistances $r(g, g')$ and $r(g, g')$ as defined by Jackson and Watts (2002). More complicated cases with 3 or more absorbing states would require us to search over "restricted g-trees" (again see Jackson and Watts 2002) which seems a computationally more complex problem.

6.5 Appendix

6.5.1 Connectionsmodel m-file

```
function P = connectionsmodel(e)
% P = connectionsmodel(e)
% output P is transition matrix for connections model with 4 actors (11 states)
%   and parameter values d - d^2 < c < d - d^3
% input e is the probability that pairs make "mistakes"
P = zeros(11);
P(2,1) = (6/6)*(1-e);
P(1,2) = (1/6)*e; P(3,2) = (4/6)*(1-e); P(4,2) = (1/6)*(1-e);
P(2,3) = (2/6)*e; P(5,3) = (1/6)*e; P(6,3) = (1/6)*(1-e); P(7,3) = (2/6)*(1-e);
P(2,4) = (2/6)*e; P(7,4) = (4/6)*(1-e);
P(3,5) = (3/6)*(1-e); P(8,5) = (3/6)*(1-e);
P(3,6) = (3/6)*e; P(8,6) = (3/6)*e;
P(3,7) = (2/6)*e; P(4,7) = (1/6)*e; P(8,7) = (2/6)*e; P(9,7) = (1/6)*(1-e);
P(5,8) = (1/6)*e; P(6,8) = (1/6)*(1-e); P(7,8) = (2/6)*(1-e); P(10,8) = (2/6)*e;
P(7,9) = (4/6)*e; P(10,9) = (2/6)*e;
P(8,10) = (4/6)*(1-e); P(9,10) = (1/6)*(1-e); P(11,10) = (1/6)*e;
P(10,11) = (6/6)*(1-e);
P = P + diag(1-sum(P));
```