

Chapter 4: Random Walks on Graphs

Introduction: From randomization tests to random walks

Chapter 2 gave many examples of how you can use choosing at random from a set to test scientific hypotheses. First, based on the applied context, you define a null model and a test statistic. Abstractly, the null model is simply a list of possible data sets that are assumed to be equally likely. In an application, the null model corresponds to the hypothesis that “the data are just random.” Abstractly, the test statistic is just a way to assign a number to each random data set. In applications, it is an “index of suspicion” that assigns extreme values to data sets of the sort you would expect to see if the null model is false. Once you have a null model and test statistic, you compute the value of your test statistic for the actual data. Then you simulate: Use your null model to generate a large number of random data sets, and keep track of the percentage of times the test statistic is at least as extreme as the value for the observed data. If this fraction is tiny, you know that the null model almost never generates data like what you actually observed. Thus “the data are just random” is not a very believable explanation for the observed data.

To make this method work, you need to be able to create random data sets. If your null model is a simple one, generating the data sets will be relatively straightforward. For example, for the simplified version of the Martin data, it was not hard to draw random subsets of size 3 from {25, 33, 35, 38, 48, 55, 55, 55, 64}. You could choose the samples by drawing at random, quite literally, from a bag of numbered chips, or you could get the computer to simulate the random draws for you. The computer can do this easily because the “data sets” have a simple structure and there aren’t very many possibilities, only 120 in all. For the finch data, on the other hand, as for a great many real problems, there is no such simple method. The possible data sets have a complicated structure (13x17 matrices of 0s and 1s with fixed row and column sums), and the number of possibilities is huge. (Even if you wrote so small that you could fit 10 of these data sets on a page, listing them all would take a stack of paper 5 billion miles high.) How can we possibly draw at random from this set?

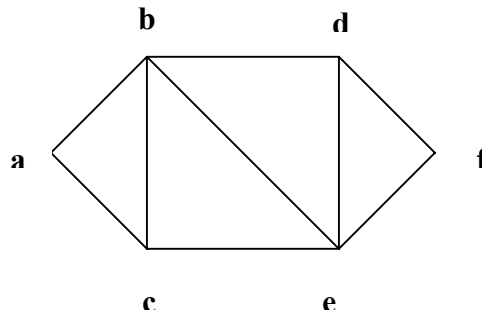
The answer is quite ingenious. In effect, we simply implement the strategy of the shortsighted astronomer from Chapter 1: Instead of asking a computer to draw at random from a list of possibilities, we’ll ask it to wander randomly from one possible data set to another. Our hope is that if you wander randomly enough, for long enough, the data set where you end up might just as well have been chosen at random. Up to this point, I’ve told you only one possible method for random wandering, due to Connor and Simberloff¹: Start with the actual data. Randomly choose a 2x2 sub-matrix, and swap the 0s and 1s, *provided* this swap leaves the row and column totals unchanged. Then pick another 2x2 sub-matrix, and if possible, swap again. If you keep doing this long enough, said C & S, and you will create a random data set. (All that work for *one* random data set! To estimate the p -value, you should have hundreds, at least.)

¹ Connor, E.F. and D. Simberloff (1979). “The assembly of species communities: chance or competition?” *Ecology* **60**, 1132-1140.

To see how well the Conner/Simberloff method works, and to prepare to improve on their original idea or extend it to other situations, you need to learn some mathematical theory about random wandering. For that, it would be much too ambitious to start with the finch data. We'll set the finches aside temporarily, in order to study vastly simpler situations.² However, keep the swap walk in the back of your mind as you read, and pause from time to time to think about how the swap walk of Chapter 1 relates to what you are reading about.

4.1 An introductory example: walking randomly on a graph.

Display 4.1 is an example of a **graph**.



Display 4.1 A graph with 6 vertices and 9 edges.

Imagine that the graph is a map of a small lily pond, with lily pads labeled $a - f$. Imagine also that a frog, having more energy than sense of direction, jumps randomly from pad to pad, always choosing with equal probabilities from among the pads joined by lines to the one where she is. To be concrete, and to introduce some technical terms in the process³, suppose our energetic but aimless frog starts at **vertex** a , and chooses from the two **adjacent** vertices (b and c) with equal probability. Suppose that her random choice takes her to b . For her next move, there are four **edges** leading to adjacent vertices (a , c , d and f). She chooses one of the four, again choosing with equal probabilities, and jumps to that vertex. She continues in this way, one **step** at a time, at each step choosing **uniformly at random** from the set of edges **incident** to her current vertex.

Exercise:

1. For the frog, each lily pad is a vertex. (a) What are the vertices for the shortsighted astronomer? What are the edges? (b) For the swap walk on a set of co-occurrence matrices, what are the vertices? What are the edges?

² This strategy, of replacing the original problem with a much simpler version, is very effective in mathematics. The Hungarian mathematician George Polya has said, "If there's a problem you can't solve, then there's a simpler problem you can't solve. Find it."

³ Fortunately, a lot of the basic terms of graph theory mean what you would think they ought to mean, based on the words themselves, and the context. Careful definitions of some terms may be useful later on, but for now, you really don't need them.

Three questions

The frog's path is an example of a **random walk on a graph**. In the applications to come, the lily pond will correspond to a null model, with each pad a data set. If the frog wanders randomly enough, for long enough, where she ends up will correspond – *we hope* – to a randomly chosen data set. For our purposes, three sets of questions will turn out to be particularly important. Answers to the first two are known.⁴ Questions in the third cluster are an active area of research.

Question 1: n-step transition probabilities.

If the frog starts at b , what is the probability that she is in c after exactly two steps? More generally, given a starting vertex u and an ending vertex v , what is the probability of moving from u to v in exactly n steps?

Question 2: limiting distribution.

Suppose the frog starts in a and takes a large number of steps, say n . Let $\hat{p}_{(n)}[a]$ be the fraction of steps $1 - n$ that she is in a , and consider the behavior of $\hat{p}_{(n)}[a]$ as the number of steps increases. Does $\hat{p}_{(n)}[a]$ converge? If so, what is the limit? More generally, given any starting vertex u and any vertex v of interest, does the fraction $\hat{p}_{(n)}[v]$ of steps at v have a limit? If so, what is its value? Does the value depend on the starting vertex? How do the answers to these questions depend on the structure of the graph?

Question 3: convergence rate.

If \hat{p} does converge to a limiting value p , how fast does it happen? In other words, what is the relationship between the “error” $|\hat{p} - p|$ and the number of steps? (Is convergence linear? logarithmic? geometric? power law? some other?) What is the relationship between the convergence rate and the structure of the graph?

A look ahead.

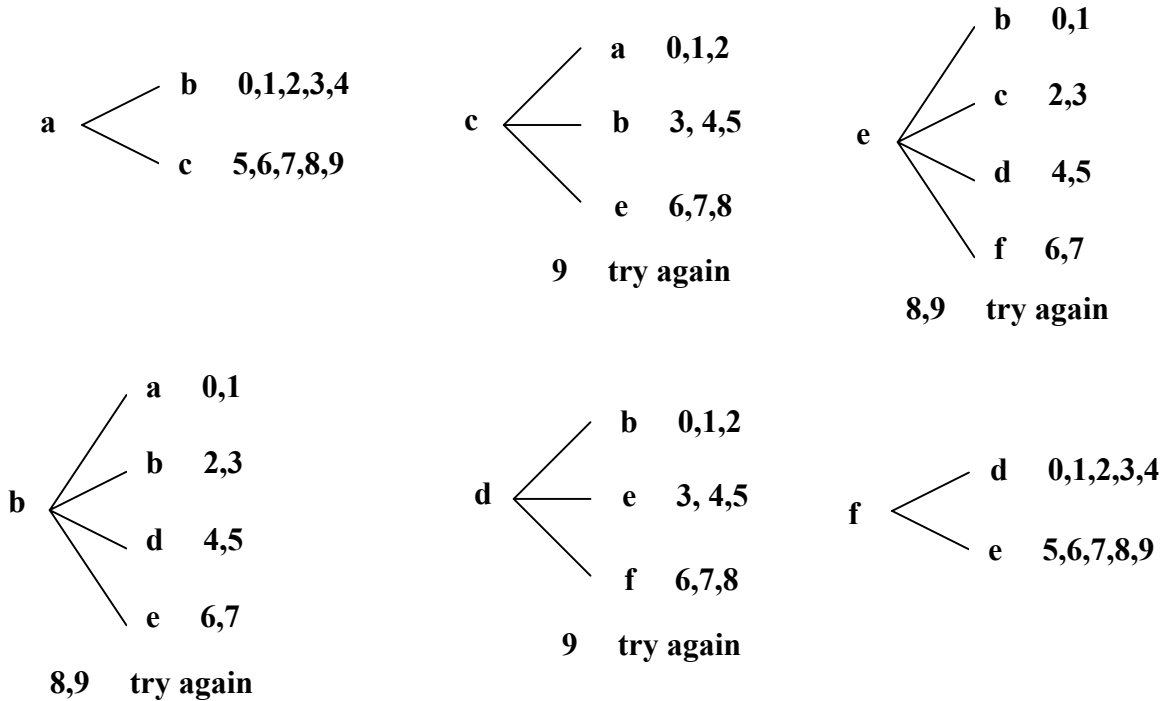
The remainder of this chapter is aimed at helping you find answers to these sets of questions. To get started, it may help you to build your intuition if you simulate a few random walks. Keep in mind that these simulations are a means to an end – an exercise for building understanding. As you carry out the simulations, observe the random behavior, and try to relate what you see to the three clusters of questions. Write down any observations that seem potentially useful, any hunches you have, and any questions that seem worth investigating.

Using random digits to simulate a random walk on a graph.

In a table of uniform random digits (or a computer algorithm for generating random digits), 0, 1, 2, ..., 9 occur with equal frequency. Display 4.3 contains such a table. To

⁴If you like to play with ideas, however, it is more fun to try to figure out the answers on your own. Rumor has it that Bradley Efron, a Stanford statistician who won one of the early MacArthur “genius” awards, doesn't really read research articles. Instead, he reads the abstract, and then tries to figure out on his own what the paper says.

use one for simulation, read the digits in sequence one at a time. To choose with equal probabilities between two outcomes, assign the digits 0, 1, ..., 4 to the first outcome, and 5, 6, ..., 9 to the second outcome. If the next random digit in your sequence is one of 0 – 4, the first outcome has “occurred”; if 5 – 9, the second outcome. To choose uniformly from among three outcomes, assign 0, 1, 2 to the first outcome, 3, 4, 5 to the second, and 6, 7, 8 to the third; if the random digit you get happens to be a 9, discard it and take the next random digit. To choose uniformly from four outcomes, assign pairs of digits: 0, 1 to the first outcome, 2, 3 to the second, 4, 5 to the third, and 6, 7 to the fourth; if your random digit is an 8 or 9, draw again, i.e., go on to the next random digit. Display 4.2 shows the assignments for a random walk on the graph in Display 4.1.



Display 4.2 Using random digits to simulate a random walk on the graph of Display 4.1

To illustrate, suppose you start your random walk at *a*, and your sequence of random digits is 8 3 1 4 0 7 7 3 3. Then your first step, from *a*, has two choices, either *b* (0, 1, 2, 3, 4) or *c* (5, 6, 7, 8, 9). The first random digit is 8, so you move to *c*. From *c*, you have three choices, *a* (0, 1, 2), *b* (3, 4, 5), or *c* (6, 7, 8). The second random digit is a 3: move to *b*. From *b* you have four choices, etc. Here is the entire walk:

8 3 1 4 0 7 7 3 3
a → *c* → *b* → *a* → *b* → *a* → *c* → *e* → *c* → *b*

Check that for the final transition, from *c* to *b*, the first random number is a 9, which we discard and replace by the next digit, a 3.

Exercise:

2. Use Display 4.2 and the random sequence 6 5 5 3 4 0 9 6 1 to generate a random walk on the graph in Display 4.1, starting from vertex *a*. (Partial answer: *c b d e ...*)

Row	A	B	C	D	E	F	G	H
1	0 2 7 0 5	9 0 4 1 6	8 0 5 2 1	3 1 0 7 2	4 6 7 1 7	1 3 4 1 2	7 4 0 0 1	9 5 6 6 6
2	5 0 4 4 3	3 8 0 7 4	0 1 6 7 0	7 0 7 4 5	6 5 2 3 9	4 9 1 0 6	4 3 4 5 2	2 0 5 1 2
3	2 7 3 3 4	8 8 7 8 4	8 8 2 2 1	1 0 6 6 3	4 7 4 5 0	2 9 8 8 4	6 7 0 5 4	3 0 3 8 6
4	3 7 0 8 8	3 9 1 1 6	2 9 3 5 1	1 5 2 3 9	9 3 2 8 9	9 0 2 4 6	0 7 7 7 7	2 0 5 5 1
5	4 3 7 3 4	9 7 4 1 4	3 4 7 2 6	9 8 0 8 4	9 3 8 6 6	2 2 1 1 4	2 8 5 3 9	5 3 3 9 9
6	1 6 1 7 2	8 7 3 4 5	8 1 1 2 4	6 7 7 8 7	3 9 5 2 8	5 6 7 7 3	1 1 2 2 7	8 0 3 3 5
7	2 4 6 4 7	7 3 9 4 2	2 6 8 4 0	1 8 2 4 2	2 7 1 9 1	3 9 0 8 2	9 0 3 8 2	8 0 5 2 8
8	2 2 8 9 4	2 3 0 0 4	9 5 9 4 8	3 4 5 7 5	4 1 7 6 0	9 3 3 4 6	7 0 8 2 6	0 3 5 6 0
9	4 3 8 1 7	2 5 4 5 2	8 2 1 3 0	5 2 7 9 2	1 6 7 3 1	9 1 7 0 8	8 5 1 6 3	7 9 8 5 6
10	4 9 7 7 9	4 7 2 9 3	5 8 7 1 6	6 5 6 8 1	3 3 5 8 6	8 5 1 7 7	0 5 9 9 0	0 7 5 3 2
11	0 8 2 4 0	6 1 2 7 6	3 5 7 9 6	2 7 4 2 9	7 5 3 1 8	9 2 5 8 1	2 3 8 7 0	4 7 5 4 3
12	9 6 4 8 2	2 9 6 0 2	1 9 1 0 6	2 8 8 2 9	2 3 5 5 1	2 2 2 4 1	2 1 4 6 1	2 4 9 6 8
13	2 0 6 6 1	4 4 5 2 8	4 1 2 8 2	5 7 2 7 9	0 1 0 7 1	0 3 8 0 3	4 9 4 4 5	7 5 0 1 6
14	4 0 6 8 4	1 4 8 0 5	9 5 9 2 9	0 8 6 0 5	2 4 8 1 2	0 6 7 8 2	5 2 6 8 8	2 6 3 5 1
15	8 8 7 4 9	0 9 6 8 0	1 9 6 3 8	8 6 6 5 3	4 8 0 1 7	1 8 7 9 8	0 5 3 5 1	7 0 0 1 5
16	6 4 6 7 0	8 7 1 5 6	0 1 7 4 5	9 7 8 7 0	8 5 6 0 4	7 5 5 4 5	8 6 2 9 0	2 7 7 7 8
17	7 5 4 0 7	1 7 7 9 0	4 9 0 7 4	8 8 0 2 4	6 6 9 3 4	3 8 4 8 4	1 7 8 7 0	6 4 3 2 6
18	1 8 5 0 3	7 9 1 0 3	4 0 4 7 2	3 0 5 2 1	1 3 8 9 8	3 4 7 0 4	7 2 9 7 2	7 0 7 3 4
19	0 9 9 6 4	1 9 4 3 0	0 7 6 9 3	2 5 7 7 0	4 4 6 1 3	2 2 7 9 7	7 4 0 4 2	5 7 4 1 3
20	3 7 3 3 4	1 2 3 7 0	4 3 9 9 4	9 2 1 0 5	9 6 8 0 1	8 4 7 8 0	8 6 0 7 6	7 7 5 6 3
21	7 6 7 3 5	6 3 4 6 6	9 8 4 2 3	9 4 1 0 3	1 5 7 3 5	0 3 0 1 5	0 1 7 4 2	8 1 2 5 3
22	9 6 8 5 8	4 7 2 1 1	6 5 2 1 1	2 1 0 4 0	1 2 5 9 0	4 4 0 3 6	8 0 5 4 2	1 8 3 4 3
23	3 4 9 5 1	2 8 7 0 4	2 9 9 9 5	6 3 3 1 9	9 3 4 8 9	9 5 2 2 6	5 6 3 0 3	5 8 5 1 2
24	7 9 6 9 3	5 4 9 4 4	6 9 2 6 6	4 5 4 2 1	6 7 5 5 9	0 6 4 0 1	2 2 4 1 4	3 7 1 7 3
25	8 9 6 6 5	1 2 6 7 4	7 1 5 0 4	6 5 0 6 0	4 9 8 3 0	8 3 8 3 9	1 3 7 8 6	1 6 0 2 5
26	3 9 0 5 9	7 6 6 7 4	6 0 1 2 1	9 4 4 1 0	4 6 0 6 3	6 5 3 0 8	2 2 3 3 1	1 0 1 4 4
27	1 9 0 6 1	8 8 5 2 4	1 2 1 4 3	8 7 0 5 3	5 3 0 7 9	9 3 5 5 8	4 3 0 9 5	4 6 1 9 1
28	9 3 2 3 0	1 1 7 0 2	3 2 8 4 9	5 6 8 5 6	1 2 9 5 6	0 0 6 5 9	6 1 4 6 6	5 8 2 2 4
29	8 5 8 7 1	7 2 9 2 3	7 1 0 4 0	4 0 4 8 7	4 6 4 8 7	8 9 2 3 7	0 5 8 4 3	0 2 7 0 7
30	9 0 1 6 6	7 4 7 8 8	1 9 6 2 3	2 3 9 4 5	3 5 7 6 3	9 8 8 2 2	7 4 1 8 7	1 8 3 2 7
31	9 1 4 3 3	6 0 2 0 6	8 6 5 4 5	3 7 6 3 2	4 9 7 5 3	4 8 0 7 8	2 7 5 9 9	4 0 1 4 8
32	9 7 2 6 2	8 4 7 7 0	8 2 3 6 2	1 1 7 0 8	7 8 1 2 2	9 5 2 0 4	4 8 8 5 3	5 0 2 0 2
33	4 2 1 2 9	8 8 5 0 9	6 7 5 4 5	0 2 1 3 7	3 0 0 9 1	9 1 2 1 1	6 7 3 8 5	5 8 7 7 7
34	7 9 3 3 2	8 7 3 8 7	6 3 2 1 0	4 2 5 5 8	6 2 4 7 3	6 2 1 3 2	9 1 8 1 1	4 4 9 6 1
35	2 7 8 5 5	9 1 6 8 7	8 3 5 8 7	8 2 6 5 7	7 4 8 2 0	5 2 9 7 0	0 1 9 2 1	8 5 2 8 8
36	4 6 9 2 0	1 2 3 4 8	3 5 7 5 1	7 1 1 4 4	6 6 1 1 4	0 5 2 5 2	3 2 7 3 0	6 3 5 3 0
37	9 6 7 6 7	1 3 3 0 4	4 4 3 7 6	2 8 2 8 8	8 0 4 8 6	5 0 3 8 3	7 0 4 6 5	2 9 2 4 0
38	5 9 2 8 5	1 3 8 2 4	9 4 3 3 6	7 8 9 8 8	6 8 3 0 9	2 7 4 2 4	9 7 9 9 7	8 3 3 4 5
39	8 0 2 9 6	5 5 0 4 0	4 2 9 6 1	3 9 4 6 2	1 3 3 5 2	9 8 3 3 5	2 3 0 5 2	8 4 4 2 4
40	8 7 7 2 8	2 2 9 1 5	0 3 2 7 4	4 2 8 8 3	1 9 2 1 4	5 7 7 9 6	6 5 1 5 8	7 6 3 9 0

Display 4.3 Table of uniform random digits

The purpose of the activity that follows is to help you think concretely about the three sets of questions.

Activity. Random walks on graphs.

Display 4.4 shows ten graphs, each one a **connected** graph of **order 5**. Pick one graph to use for this activity.

Simulation using a random number table

Step 1. Following the introductory example as a guide, make a diagram like Display 4.3 for your graph.

Step 2. Then use a random number table to simulate 5 steps of a random walk on your graph. Record the path, and compute the fraction of steps in at each vertex. Which vertex was most frequent? Least frequent? Is this pattern related to the structure of your graph?

Step 3. Now carry out an additional 15 steps, but first, based on the structure of your graph, predict which vertices will be most frequent and least frequent for the set of 20 steps taken together. Then do the simulation, and check your predictions.

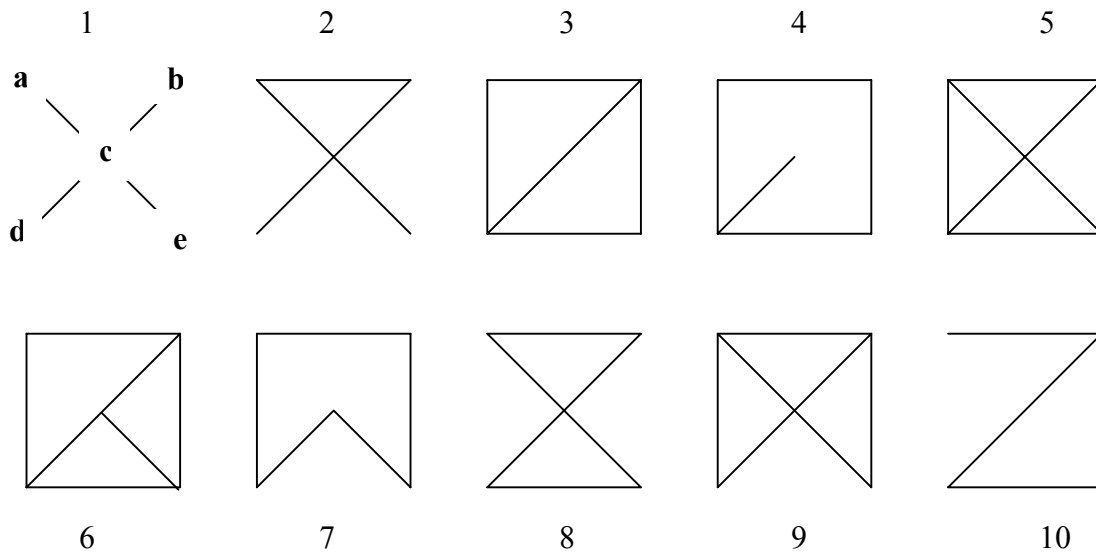
Computer simulation

Step 4. Display 4.5 shows S-Plus code for simulating a random walk on the graph shown in Display 4.1. Read through the code and predict what it will do; then run it to verify your predictions.

Step 5. Now alter the list `vert` in the S-Plus code so that it corresponds to your own graph from Display 4.3. This time, before you simulate, make a more elaborate prediction about the limiting distribution: Rank the vertices according to what you predict for their frequencies, from least frequent to most frequent. Now change `nSteps` to 10000, run the random walk, compute frequencies, and compare with your predictions.

Discussion question:

3. One of the graphs 1 – 10 in Display 4.4 corresponds to the swap walk on the 3x3 co-occurrence matrices whose row and column sums are 2,1,2. Which graph is it? How can you tell? (See Exercises 41-44 at the end of this chapter.)



Display 4.4 Ten connected graphs of order 5

```

v1 <- c(2,3)
v2 <- c(1,3,4,5)
v3 <- c(1,2,5)
v4 <- c(2,5,6)
v5 <- c(2,3,4,6)
v6 <- c(4,5)
vert <- list(v1,v2,v3,v4,v5,v6)

GraphWalk <- function(vert,start,NSteps){
  walk <- rep(0,NSteps)

  walk[1] <- start
  i <- 1
  while (i < NSteps) {
    walk[i+1] <- sample(vert[[walk[i]]],1)

    i <- i+1
  }
  return(walk)
}

NSteps <- 100
walk <- GraphWalk(vert,1,NSteps)
walk

prob <- rep(0,length(vert))
for (i in 1:length(vert)) {prob[i] <- length(walk[walk==i])/NSteps}
prob

```

Display 4.5 S-Plus code for random walks on graphs

4.2 Limiting distributions for random walks on graphs

Question: When (for which vertices of which graphs) does the fraction of steps at a given vertex converge to a limit as the number of steps increases? Find a formula for the limit based on the structure of the graph.

Background: looking back, looking ahead

Remind yourself of the goal from Chapter 1: to generate random data sets in order to estimate p -values. To get unbiased estimates, we need the data sets to be equally likely. If we are to generate our data sets using random walks on graphs, we need the limiting probabilities to be **uniform**, that is, equal to each other. This is the practical motivation for Question 2 of the introduction: Does the fraction of time spent at a vertex u converge to a limit as the number of steps increases? If so, what is the limit?

It will turn out that the Connor/Simberloff method of random swaps does *not* produce uniform limiting probabilities. (Using their method, some data sets are more likely than others.) Fortunately, in the process of coming to understand why this is true, you'll learn what you need to know in order to fix their algorithm so that it does give uniform probabilities. The method for fixing their approach, called the **Metropolis algorithm**, has a long and impressive history. It was invented half a century ago to study the diffusion of nuclear particles, in connection with the Manhattan Project, the US project to develop an atomic bomb. Since then, it has been used for a variety of less violent purposes, including research on the structure of large molecules in biochemistry and a host of statistical applications in economics.

A Million Frog March: another way to think about the limiting distribution

Turn back to the graph of Display 4.1, and imagine releasing one million frogs at vertex a . After one move, half will be at b , half at c . At move 2, one fourth of the 500,000 frogs at b jump back to a , one fourth jump to c , one fourth to d , and one-fourth to e . At the same time, one third of the 500,000 frogs at c jump to each of the three vertices adjacent to c : one third return to another third go to b , and the remaining go to e . At step 3, the frogs at each vertex jump in equal numbers to the adjacent vertices, and so on.

What happens to the proportions of frogs at the various vertices as time passes? Does the process settle down to an equilibrium distribution, for which the proportions of frogs at the various vertices remain the same from one step to the next?

Some notation.

Standard notation for random walks uses X_t for the outcome at time t . In this notation, $X_0 = a$ means the process starts at vertex a . By convention, we say the process is “in state a at time 0.” More notation:

$\{t: X_t = a\}$ the set of times (steps) when the process is in a

$\#\{t: X_t = a\}$ the number of steps when the process is in a

$\#\{t: X_t = a, 1 \leq t \leq n\}/n$ the fraction of steps 1 through n spent in a

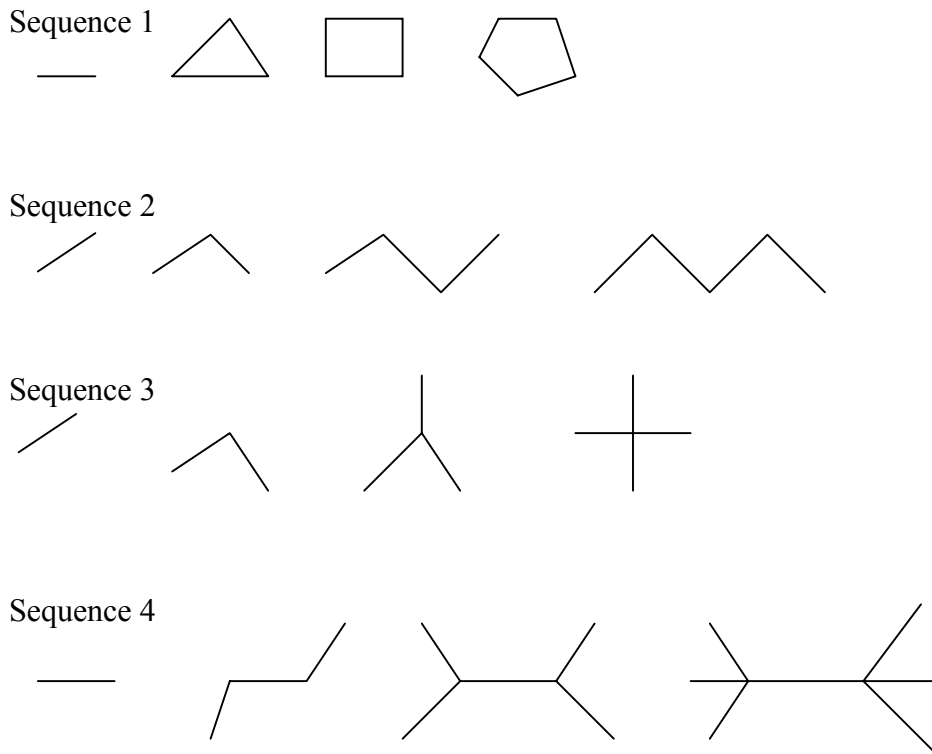
The last quantity, $(\# \text{ steps in } a)/(\# \text{ steps})$, is the focus of Investigation 4 below. Does this fraction converge to a limiting value as the number of steps increases? If so, let $\hat{p}_\infty[a]$ be the limiting value for a .

$$\hat{p}_\infty[a] = \lim_{n \rightarrow \infty} \frac{\#\{t: X_t = a, 1 \leq t \leq n\}}{n}$$

To show the starting vertex, say u , write $\hat{p}_\infty[a | u]$. Your goal is to determine when $\hat{p}_\infty[a | u]$ exists, and to find a formula for $\hat{p}_\infty[a | u]$ in terms of the structure of the graph.

Investigation

4. Using S-plus as needed, discover as much as you can about the relationship between the structure of a graph and the limiting fractions $\hat{p}_\infty[\cdot]$. Start with simple graphs for which you can guess the answer. (For example, if the graph has only one vertex, what is the limiting fraction? What if the graph has two vertices, connected by an edge?) One strategy is to invent sequences of graphs that will allow you to discover patterns. Display 4.6 shows some possible sequences to try.



Display 4.6 Four sequences of graphs

4.3 Convergence rates for random walks on graphs.

Question: (a) How many steps does it take to get to a “random” vertex?
 (b) How quickly does the fraction of time spent at a vertex converge to its limit, and what is the relation between the structure of the graph and the rate of convergence?

Background: relation to Chapter 1.

The “random swap” method for generating matrices uses a long sequence of swaps to generate a single random data matrix. How many swaps do you need to make before you can reasonably say you’ve created a random matrix? Notice that this question is very similar to a question from a much more familiar setting: How many times do you have to shuffle a deck of cards before you can be reasonably confident that the deck is thoroughly mixed, that is, that all possible orderings have approximately the same chance?

More generally, this section deals with the question, “How long does it take for a random walk on a graph to converge to its limiting behavior?” The question turns out to be more complicated than it might seem at first, because there are in fact two kinds of convergence involved.

Discussion: Two kinds of convergence.

5. Consider a walk on a two-point graph: a — b . Suppose the walk starts at a , that is, $X_0 = a$. Write out the first 10 steps of a random walk on the two-point graph.
6. What is the probability that the walk is in state a after 1 step? 2 steps? 3 steps? n steps?

Notation: Let p , with no hat, stand for a probability. Show the number of steps using a superscript in parentheses: $p^{(n)}$. (To see why the parentheses are needed, think about what p^n would mean.) Show the state in square brackets: $p^{(n)}[a]$. Thus, for the two-point walk, $p^{(0)}[a] = \Pr\{X_0 = a\} = 1$.

7. Does $p^{(n)}[a]$ converge to a limit as $n \rightarrow \infty$? Give a justification for your answer.
8. Use your results from (5) to compute, for $n = 1, 2, 3, \dots, 10$, the observed fraction of steps in state a , not counting the starting state.

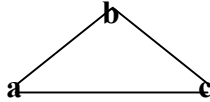
Notation. As before, we use \hat{p} for the observed fraction. We use a subscript in parentheses to tell the number of steps: $\hat{p}_{(n)}$. (The superscript in the previous notation refers to the n^{th} step. The subscript here refers to the first n steps.) Finally, as above, we use square brackets to indicate the state: $\hat{p}_{(n)}[a]$. Thus, for example, the observed fraction of steps 1 – 5 spent in state a would be $\hat{p}_{(5)}[a]$.

9. For the walk on the two-point graph, does $\hat{p}_{(n)}[a]$ converge to a limit?

Definition. The convergence of observed fractions $\hat{p}_{(n)}$, when it occurs, is called **ergodic** convergence. The convergence of probabilities $p^{(n)}$, when it occurs, is called **mixing**.

Exercises: Random walk on a triangle.

Consider the graph



10. Simulate 10 steps of a random walk starting at $X_0 = a$. (Note that at each step there are always two equally likely choices, so you can toss a coin to do the simulation.)

11. Use your path from Step 1 to compute $\hat{p}_{(n)}[a]$ for $n = 1, 2, \dots, 10$. Graph $\hat{p}_{(n)}[a]$ versus n , and compare your graph with others in the class.

12. Compute $p^{(1)}[a]$, $p^{(1)}[b]$, $p^{(1)}[c]$.

13. List all walks of length 2 that start from a . Convince yourself that these walks are equally likely, and so

$$p^{(2)}[a] = \frac{\# \text{ walks of length 2 that end at } a}{\text{total } \# \text{ walks of length 2}}.$$

Compute $p^{(2)}[a]$, $p^{(2)}[b]$, $p^{(2)}[c]$.

14. Extend the method of Step 4 to compute $p^{(3)}[a]$ and $p^{(3)}[a]$.

15. How many walks of length 2 are there? of length 3? of length n ?

16. How many walks of length n are there that agree on steps 0, 1, 2, ..., $n-1$?

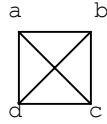
17. There are eight walks of length 3; two of them end at a , and six end at either b or c . Explain how to use this information, without listing walks, to determine the number of walks of length 4 that end at a .

Investigation

18. Use your reasoning from (17) to write a formula for $p^{(n)}[a]$ in terms of $p^{(n-1)}[a]$.

19. Use your formula to decide whether $p^{(n)}[a]$ converges to a limit. If it does converge, use your formula to investigate the rate of convergence (mixing rate).

A **complete** graph is one with all possible edges. The complete four point graph is shown below:

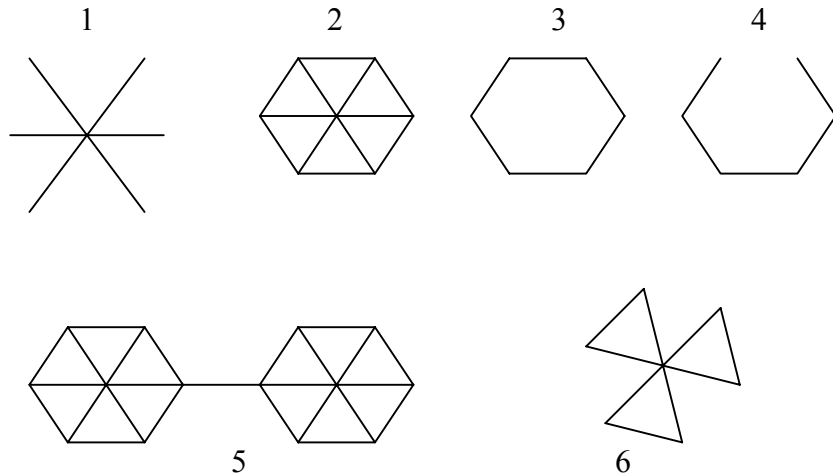


20. Find a formula for $p^{(n)}[a]$ for the walk on the complete four point graph, and use your formula to investigate the mixing rate.

21. Generalize to the walk on a complete graph with k vertices.

Exercise. Mixing rates and graph structure: a look ahead.

23. Display 4.7 shows several graphs. Imagine a random walk on each graph; think through an imaginary sequence of random steps. (a) Which random walk will get to its limiting distribution fastest? slowest? (b) Rank order the graphs from fastest to slowest. (c) Explain the reasoning you used to get your ranking.



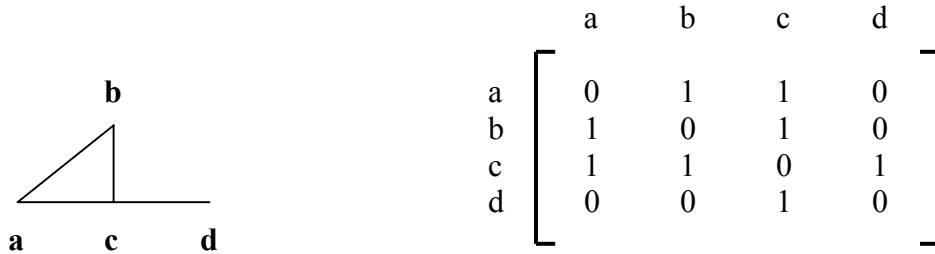
Display 4.7 Six graphs for comparing convergence rates

4.4 Adjacency matrices for graphs

Your work on convergence rates in the last section took advantage of a simple special case: for complete graphs, all walks of the same length are equally likely. This means that to compute probabilities, “all” you have to do is list and count walks. It also means that the same method of listing and counting won’t work for graphs whose vertices have differing numbers of neighbors. Generalizing to all graph walks will require a new, more general way to compute probabilities for these walks. That is the topic of the next two sections, and all of the next chapter as well.

Definition. The **adjacency matrix** of a graph has one row and one column for each vertex. If u and v are vertices and a_{uv} is the entry in row u , column v , then $a_{uv} = 1$ if u and v are adjacent (joined by an edge), and $a_{uv} = 0$ otherwise.

Example:



Drill

23. Find the adjacency matrix for the graph in Display 4.1.
24. Find the adjacency matrix for all connected graphs of order 4 (i.e., with 4 vertices).
25. There are sixteen different 2x2 matrices of 0s and 1s. Find all the ones that can be adjacency matrices for graphs of order 2, and draw the corresponding graphs.
26. List all the adjacency matrices for graphs of order 3. For each, draw the corresponding graph.
27. True or false, and explain:
 - a. For all pairs of vertices in a graph, $a_{uv} = a_{vu}$. Thus $A = A^T$, where A^T is the transpose of A .
 - b. For every vertex u , $a_{uu} = 0$.

Discussion:

28. How can you tell from the adjacency matrix whether a graph is connected?

29. Let A be the adjacency matrix for a graph.
- What do the elements of A^2 tell you?
 - Explain why $A^2A = AA^2$.
 - What do the elements of A^3 tell you?

Exercises

30. Let A be the adjacency matrix for a graph. Tell how to find:
- The degree of a vertex
 - The number of walks of length 2 from u to v .
 - The number of **cycles** of length 3. (A cycle is a walk of length 3 or more that ends at its starting vertex.)
31. Let $C = \{c_{ij}\}$ be a **co-occurrence matrix** for the finch data, like the one in Display 1.1. Thus for row i and column j , $c_{ij} = 1$ if species i occurs on island j ; otherwise $c_{ij} = 0$. For each of the matrix products below (i) tell whether the product exists; (ii) if not, explain why not; if so, explain what the (i,j) element of the product tells you.
- C^2
 - $C^T C$
 - CC^T
 - $(C^T)^2$
32. Given two adjacency matrices, how can you tell if the corresponding graphs are isomorphic?

Definition. Informally, two graphs are isomorphic if you can turn one into the other by moving its vertices. Formally, the two graphs are defined by their vertex and edge sets: $G_1 = \{E_1, V_1\}$ and $G_2 = \{E_2, V_2\}$. The two graphs are **isomorphic** if and only if there is a one-one function $f: V_1 \rightarrow V_2$ that preserves edges, i.e., for which

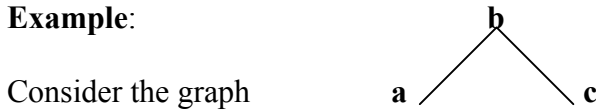
$$(u, v) \in E_1 \Leftrightarrow (f(u), f(v)) \in E_2.$$

4.5 Two-stage transition probabilities

Skill-building goals:

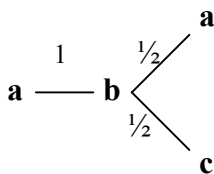
Learn to draw tree diagrams for two-step transitions, use them to compute two-step transition probabilities, and begin to develop intuition about how these probabilities are related to graph structure.

Example:



From a , the only edge leads to b , so the one-step transition probability from a to b is 1. (Notation: $P(a, b) = 1$, or $p_{ab} = 1$.) Similarly, from c there is only one edge, and $P(c, b) = 1$, or $p_{cb} = 1$. From b , there are two equally likely edges, and $p_{ba} = p_{bc} = 1/2$.

What about two-step transitions? We can show these in a tree diagram. For example, starting from a , there are two paths of length two:



Consider the probability of going from a back to a in two steps. There is only one **2-cycle** that does this, the one that corresponds to the upper part of the tree diagram: $a \rightarrow b \rightarrow a$.⁵ The first step has probability 1 (100% of the time, a frog in a jumps to b) and the second step has probability .5 (50% of the time, a frog in b jumps back to a). To find the probability for the two steps together, multiply their probabilities: $P(a \rightarrow b \rightarrow a) = P(a \rightarrow b)P(b \rightarrow a) = p_{ab}p_{ba} = (1)(1/2) = 1/2$. So a frog in a will jump to b and then back to a again 50% of 100%, or 50% of the time.

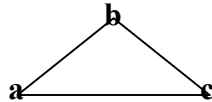
By the same reasoning, this time applied to the path $a \rightarrow b \rightarrow c$, you can check that $p_{ac} = 1/2$. Finally, as you can convince yourself from the graph, there is no way to get from a to b in two steps, so $p_{bb} = 0$.

⁵ A **walk** is any connected sequence of vertices and edges. A **path** is a walk with no repeated vertices. A **cycle** is a walk of length 3 or more that starts and ends at the same vertex, but has no other repeated vertices. The **length** of a walk is the number of edges, i.e., the number of steps.

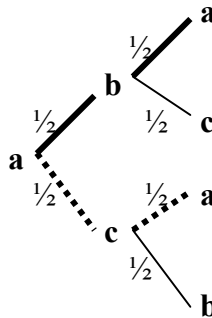
The two-step transition probabilities can be recorded in a matrix:

$$\begin{array}{c} \text{From:} \\ a \\ b \\ c \end{array} \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ \end{bmatrix} = P^{(2)}$$

The drill exercises will ask you to complete the matrix. First, though, consider another example. Suppose the graph had been the **complete** graph of order 3, that is, the one containing all possible edges:



What is the probability that a frog at *a* returns to *a* after two steps? Notice that for this example, there are two different cycles from *a* back to *a* in two steps: *a*—*b*—*a* and *a*—*c*—*a*. These cycles correspond to the first and third endpoints on the right in the tree diagram:



Using the same logic as for the first example, we get walk probabilities by *multiplying*. Thus $P(a \rightarrow b \rightarrow a) = P(a \rightarrow b)P(b \rightarrow a) = p_{ab}p_{ba} = (\frac{1}{2})(\frac{1}{2}) = \frac{1}{4}$. (50% of the time, a frog in *a* jumps to *b*, and for 50% of those times that she gets to *b*, she jumps back to *a*, so 25% of her walks from *a* go to *b* and then back to *a* again.) Similarly, the other walk (via *c*) leading back to *a* has probability $\frac{1}{4}$: $P(a \rightarrow c \rightarrow a) = (\frac{1}{2})(\frac{1}{2}) = \frac{1}{4}$. How do we combine our two $\frac{1}{4}$ s? Whereas before, we multiplied, here we *add*:

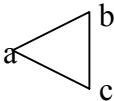
$$P(a \rightarrow a \text{ in two steps}) = P^{(2)}(a,a) = P(a \rightarrow b \rightarrow a) + P(a \rightarrow c \rightarrow a) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}.$$

How is this time different? Think of the tree diagram as a system of pipes, with large pipes that can accommodate larger flow, and small pipes that can accommodate less. At each branch of the tree, the flow divides, with part going one way, part another. Whenever this happens, only a fraction of the flow coming into a vertex gets passed along each outgoing pipe. (In our examples, wherever there are two branches, half the flow goes each way.) Multiplying the fractions keeps track of what percentage of the original flow goes along any particular sequence of pipes. Now consider the total flow

into a after two steps. Some of it comes from $a \rightarrow b \rightarrow a$, and some comes from $a \rightarrow c \rightarrow a$. The combined flow into a is the total of the two components of flow, so we add.⁶

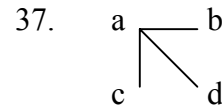
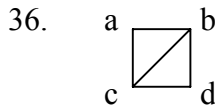
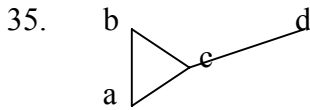
Drill Exercises:

33. Work with the graph $a \text{---} b \text{---} c$
- Use tree diagrams to show two-step transitions starting in c ; then compute the two-step transition probabilities.
 - Repeat part (a), this time for two-step transitions starting in b .
 - Use your transition probabilities to fill in the rest of the matrix $P^{(2)}$ above.

34. Work with the graph 

Show all two-step transitions using tree diagrams. Then compute all two-step transition probabilities and record them in a matrix.

Additional graphs for practice:



Investigation. Two-step transition probabilities

38. Draw all the connected graphs of order 4. How many are there? (Consider two graphs to be the same if you can turn one into the other by rearranging, reordering and/or renaming vertices.)

For the next two questions, try to answer by inspecting the structure of the graphs, rather than by doing lots of computations.

39. Find the graph, from among those in (38), which has
- the largest two-step transition probability from a vertex back to itself;
 - the smallest two-step transition probability from a vertex back to itself;

⁶ It will turn out that there is a simple, quick way to compute two-step transition probabilities *without* drawing tree diagrams, and we'll get to it before too much longer. However (as with many shortcuts in mathematics), getting to the shortcut prematurely has a cost. Shortcuts save time precisely because they jump over a number of steps, and it is often these very steps that you need to follow several times in order to build your intuition. That's why the exercises that follow ask you to find two-stage transition probabilities "the long way," and why the investigations ask you to think intuitively rather than computationally.

- c. the largest two-step transition probability from one vertex to a different vertex;
 - d. the smallest two-step transition probability from a vertex to a different vertex.
40. For a connected graph of order 4, what is the largest possible number of vertex pairs (u,v) for which $p^{(2)}(u,v) = 0$?

Exercises: The Finch Connection

41. List all five of the 3×3 matrices of 0s and 1s with row sums 2, 1, 2 and column sums 2, 1, 2. Suppose you use the Conner/Simberloff method to generate random matrices. Would you predict that over the long run their method will produce the five different matrices with equal frequencies?
42. Label the five matrices a, b, c, d, e . Divide up the matrices among the members of the class. For each matrix, find all the swappable 2×2 sub-matrices. Then, for each possible swap, find the matrix that results. (Since the swaps leave row and column sums unchanged, the resulting matrix should always be one of $a - e$.)
43. Combine all the class results into a single diagram that shows all the moves that result from 2×2 swaps: List each matrix once, and show each move from one matrix to another as a line from the first matrix to the second. (Is it necessary to indicate the direction of the move?)
44. What does your work here tell you about the C/S method?