Probability and Computing

Randomization and Probabilistic Techniques in Algorithms and Data Analysis

Second Edition

Michael Mitzenmacher Eli Upfal

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

4843/24, 2nd Floor, Ansari Road, Daryaganj, Delhi - 110002, India

79 Anson Road, #06-04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781107154889
10.1017/9781316651124

First published 2017

Printed in the United Kingdom by Clays, St Ives plc

A catalogue record for this publication is available from the British Library.

Library of Congress Cataloging in Publication Data
Names: Mitzenmacher, Michael, 1969– author. | Upfal, Eli, 1954– author.
Title: Probability and computing / Michael Mitzenmacher Eli Upfal.
Description: Second edition. | Cambridge, United Kingdom ;
New York, NY, USA : Cambridge University Press, [2017] |
Includes bibliographical references and index.
Identifiers: LCCN 2016041654 | ISBN 9781107154889
Subjects: LCSH: Algorithms. | Probabilities. | Stochastic analysis.
Classification: LCC QA274.M574 2017 | DDC 518/.1 – dc23
LC record available at https://lccn.loc.gov/2016041654

ISBN 978-1-107-15488-9 Hardback

Additional resources for this publication at www.cambridge.org/Mitzenmacher.

CHAPTER SEVEN

Markov Chains and Random Walks

Markov chains provide a simple but powerful framework for modeling random processes. We start this chapter with the basic definitions related to Markov chains and then show how Markov chains can be used to analyze simple randomized algorithms for the 2-SAT and 3-SAT problems. Next we study the long-term behavior of Markov chains, explaining the classifications of states and conditions for convergence to a stationary distribution. We apply these techniques to analyzing simple gambling schemes and a discrete version of a Markovian queue. Of special interest is the limiting behavior of random walks on graphs. We prove bounds on the covering time of a graph and use this bound to develop a simple randomized algorithm for the *s–t* connectivity problem. Finally, we apply Markov chain techniques to resolve a subtle probability problem known as Parrondo's paradox.

7.1. Markov Chains: Definitions and Representations

A *stochastic process* $\mathbf{X} = \{X(t) : t \in T\}$ is a collection of random variables. The index t often represents time, and in that case the process $\mathbf{X}$ models the value of a random variable X that changes over time.

We call $X(t)$ the *state* of the process at time t. In what follows, we use X_t interchangeably with $X(t)$. If, for all t, X_t assumes values from a countably infinite set, then we say that $\mathbf{X}$ is a *discrete space* process. If X_t assumes values from a finite set then the process is *finite*. If T is a countably infinite set we say that $\mathbf{X}$ is a *discrete time* process.

In this chapter we focus on a special type of discrete time, discrete space stochastic process $X_0, X_1, X_2, \ldots$ in which the value of X_t depends on the value of X_{t-1} but *not* on the sequence of states that led the system to that value.

Definition 7.1: *A discrete time stochastic process $X_0, X_1, X_2, \ldots$ is a Markov chain if*[1]

$$\begin{aligned}\Pr(X_t = a_t \mid X_{t-1} = a_{t-1}, X_{t-2} = a_{t-2}, \ldots, X_0 = a_0) &= \Pr(X_t = a_t \mid X_{t-1} = a_{t-1})\\ &= P_{a_{t-1},a_t}.\end{aligned}$$

[1] Strictly speaking, this is a time-homogeneous Markov chain; this will be the only type we study in this book.

This definition expresses that the state X_t depends on the previous state X_{t-1} but is independent of the particular history of how the process arrived at state X_{t-1}. This is called the *Markov property* or *memoryless property*, and it is what we mean when we say that a chain is *Markovian*. It is important to note that the Markov property does not imply that X_t is independent of the random variables $X_0, X_1, \ldots, X_{t-2}$; it just implies that any dependency of X_t on the past is captured in the value of X_{t-1}.

Without loss of generality, we can assume that the discrete state space of the Markov chain is $\{0, 1, 2, \ldots, n\}$ (or $\{0, 1, 2, \ldots\}$ if it is countably infinite). The transition probability

$$P_{i,j} = \Pr(X_t = j \mid X_{t-1} = i)$$

is the probability that the process moves from i to j in one step. The Markov property implies that the Markov chain is uniquely defined by the one-step *transition matrix*:

$$\mathbf{P} = \begin{pmatrix} P_{0,0} & P_{0,1} & \cdots & P_{0,j} & \cdots \\ P_{1,0} & P_{1,1} & \cdots & P_{1,j} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ P_{i,0} & P_{i,1} & \cdots & P_{i,j} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix}.$$

That is, the entry in the ith row and jth column is the transition probability $P_{i,j}$. It follows that, for all i, $\sum_{j\geq 0} P_{i,j} = 1$.

This transition matrix representation of a Markov chain is convenient for computing the distribution of future states of the process. Let $p_i(t)$ denote the probability that the process is at state i at time t. Let $\bar{p}(t) = (p_0(t), p_1(t), p_2(t), \ldots)$ be the vector giving the distribution of the state of the chain at time t. Summing over all possible states at time $t-1$, we have

$$p_i(t) = \sum_{j\geq 0} p_j(t-1)P_{j,i}$$

or[2]

$$\bar{p}(t) = \bar{p}(t-1)\mathbf{P}.$$

We represent the probability distribution as a row vector and multiply $\bar{p}\mathbf{P}$ instead of $\mathbf{P}\bar{p}$ to conform with the interpretation that starting with a distribution $\bar{p}(t-1)$ and applying the operand $\mathbf{P}$, we arrive at the distribution $\bar{p}(t)$.

For any $m \geq 0$, we define the m-step transition probability

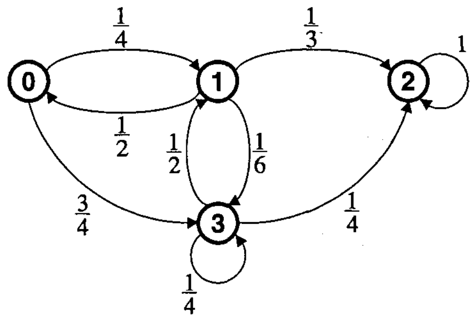
$$P^m_{i,j} = \Pr(X_{t+m} = j \mid X_t = i)$$

as the probability that the chain moves from state i to state j in exactly m steps.

Conditioning on the first transition from i, we have

$$P^m_{i,j} = \sum_{k\geq 0} P_{i,k}P^{m-1}_{k,j}. \tag{7.1}$$

[2] Operations on vectors are generalized to a countable number of elements in the natural way.



$$\mathbf{P} = \begin{bmatrix} 0 & 1/4 & 0 & 3/4 \\ 1/2 & 0 & 1/3 & 1/6 \\ 0 & 0 & 1 & 0 \\ 0 & 1/2 & 1/4 & 1/4 \end{bmatrix}$$

Figure 7.1: A Markov chain (*left*) and the corresponding transition matrix (*right*).

Let $\mathbf{P^{(m)}}$ be the matrix whose entries are the m-step transition probabilities, so that the entry in the ith row and jth column is $P^m_{i,j}$. Then, applying Eqn. (7.1) yields

$$\mathbf{P^{(m)}} = \mathbf{P} \cdot \mathbf{P^{(m-1)}};$$

by induction on m,

$$\mathbf{P^{(m)}} = \mathbf{P^m}.$$

Thus, for any $t \geq 0$ and $m \geq 1$,

$$\bar{p}(t+m) = \bar{p}(t)\mathbf{P^m}.$$

Another useful representation of a Markov chain is by a directed, weighted graph $D = (V, E, w)$. The set of vertices of the graph is the set of states of the chain. There is a directed edge $(i, j) \in E$ if and only if $P_{i,j} > 0$, in which case the weight $w(i, j)$ of the edge (i, j) is given by $w(i, j) = P_{i,j}$. Self-loops, where an edge starts and ends at the same vertex, are allowed. Again, for each i we require that $\sum_{j:(i,j)\in E} w(i, j) = 1$. A sequence of states visited by the process is represented by a directed path on the graph. The probability that the process follows this path is the product of the weights of the path's edges.

Figure 7.1 gives an example of a Markov chain and the correspondence between the two representations. Let us consider how we might calculate with each representation the probability of going from state 0 to state 3 in exactly three steps. With the graph, we consider all the paths that go from state 0 to state 3 in exactly three steps. There are only four such paths: 0–1–0–3, 0–1–3–3, 0–3–1–3, and 0–3–3–3. The probabilities that the process follows each of these paths are 3/32, 1/96, 1/16, and 3/64, respectively. Summing these probabilities, we find that the total probability is 41/192. Alternatively, we can simply compute

$$\mathbf{P}^3 = \begin{bmatrix} 3/16 & 7/48 & 29/64 & 41/192 \\ 5/48 & 5/24 & 79/144 & 5/36 \\ 0 & 0 & 1 & 0 \\ 1/16 & 13/96 & 107/192 & 47/192 \end{bmatrix}.$$

2-SAT Algorithm:

1. Start with an arbitrary truth assignment.
2. Repeat up to $2mn^2$ times, terminating if all clauses are satisfied:
   (a) Choose an arbitrary clause that is not satisfied.
   (b) Choose uniformly at random one of the literals in the clause and switch the value of its variable.
3. If a valid truth assignment has been found, return it.
4. Otherwise, return that the formula is unsatisfiable.

Algorithm 7.1: 2-SAT algorithm.

The entry $P^3_{0,3} = 41/192$ gives the correct answer. The matrix is also helpful if we want to know the probability of ending in state 3 after three steps when we begin in a state chosen uniformly at random from the four states. This can be computed by calculating

$$(1/4, 1/4, 1/4, 1/4)\mathbf{P}^3 = (17/192, 47/384, 737/1152, 43/288);$$

here the last entry, 43/288, is the required answer.

7.1.1. *Application: A Randomized Algorithm for 2-Satisfiability*

Recall from Section 6.2.2 that an input to the general satisfiability (SAT) problem is a Boolean formula given as the conjunction (AND) of a set of clauses, where each clause is the disjunction (OR) of literals and where a literal is a Boolean variable or the negation of a Boolean variable. A solution to an instance of a SAT formula is an assignment of the variables to the values True (T) and False (F) such that all the clauses are satisfied. The general SAT problem is NP-hard. We analyze here a simple randomized algorithm for 2-SAT, a restricted case of the problem that is solvable in polynomial time.

For the k-satisfiability (k-SAT) problem, the satisfiability formula is restricted so that each clause has exactly k literals. Hence an input for 2-SAT has exactly two literals per clause. The following expression is an instance of 2-SAT:

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_1 \vee x_2) \wedge (x_4 \vee \overline{x_3}) \wedge (x_4 \vee \overline{x_1}). \tag{7.2}$$

One natural approach to finding a solution for a 2-SAT formula is to start with an assignment, look for a clause that is not satisfied, and change the assignment so that the clause becomes satisfied. If there are two literals in the clause, then there are two possible changes to the assignment that will satisfy the clause. Our 2-SAT algorithm (Algorithm 7.1) decides which of these changes to try randomly. In the algorithm, n denotes the number of variables in the formula and m is an integer parameter that determines the probability that the algorithm terminates with a correct answer.

In the instance given in (7.2), if we begin with all variables set to False then the clause $(x_1 \vee x_2)$ is not satisfied. The algorithm might therefore choose this clause and then select x_1 to be set to True. In this case the clause $(x_4 \vee \overline{x_1})$ would be unsatisfied and the algorithm might switch the value of a variable in that clause, and so on.

If the algorithm terminates with a truth assignment, it clearly returns a correct answer. The case where the algorithm does not find a truth assignment requires some care, and we will return to this point later. Assume for now that the formula is satisfiable and that the algorithm will actually run as long as necessary to find a satisfying truth assignment.

We are mainly interested in the number of iterations of the while-loop executed by the algorithm. We refer to each time the algorithm changes a truth assignment as a *step*. Since a 2-SAT formula has $O(n^2)$ distinct clauses, each step can be executed in $O(n^2)$ time. Faster implementations are possible but we do not consider them here. Let S represent a satisfying assignment for the n variables and let A_i represent the variable assignment after the ith step of the algorithm. Let X_i denote the number of variables in the current assignment A_i that have the same value as in the satisfying assignment S. When $X_i = n$, the algorithm terminates with a satisfying assignment. In fact, the algorithm could terminate before X_i reaches n if it finds another satisfying assignment, but for our analysis the worst case is that the algorithm only stops when $X_i = n$. Starting with $X_i < n$, we consider how X_i evolves over time, and in particular how long it takes before X_i reaches n.

First, if $X_i = 0$ then, for any change in variable value on the next step, we have $X_{i+1} = 1$. Hence

$$\Pr(X_{i+1} = 1 \mid X_i = 0) = 1.$$

Suppose now that $1 \le X_i \le n-1$. At each step, we choose a clause that is unsatisfied. Since S satisfies the clause, that means that A_i and S disagree on the value of at least one of the variables in this clause. Because the clause has no more than two variables, the probability that we increase the number of matches is at least $1/2$; the probability that we increase the number of matches could be 1 if we are in the case where A_i and S disagree on the value of both variables in this clause. It follows that the probability that we decrease the number of matches is at most $1/2$. Hence, for $1 \le j \le n-1$,

$$\Pr(X_{i+1} = j+1 \mid X_i = j) \ge 1/2;$$
$$\Pr(X_{i+1} = j-1 \mid X_i = j) \le 1/2.$$

The stochastic process $X_0, X_1, X_2, \ldots$ is not necessarily a Markov chain, since the probability that X_i increases could depend on whether A_i and S disagree on one or two variables in the unsatisfied clause the algorithm chooses at that step. This, in turn, might depend on the clauses that have been considered in the past. However, consider the following Markov chain $Y_0, Y_1, Y_2, \ldots$:

$$Y_0 = X_0;$$
$$\Pr(Y_{i+1} = 1 \mid Y_i = 0) = 1;$$
$$\Pr(Y_{i+1} = j+1 \mid Y_i = j) = 1/2;$$
$$\Pr(Y_{i+1} = j-1 \mid Y_i = j) = 1/2.$$

The Markov chain $Y_0, Y_1, Y_2, \ldots$ is a pessimistic version of the stochastic process $X_0, X_1, X_2, \ldots$ in that, whereas X_i increases at the next step with probability at least

$1/2$, Y_i increases with probability exactly $1/2$. It is therefore clear that the expected time to reach n starting from any point is larger for the Markov chain Y than for the process X, and we use this fact hereafter. (A stronger formal framework for such ideas is developed in Chapter 12.)

This Markov chain models a random walk on an undirected graph G. (We elaborate further on random walks in Section 7.4.) The vertices of G are the integers $0, \ldots, n$ and, for $1 \le i \le n-1$, node i is connected to node $i-1$ and node $i+1$. Let h_j be the expected number of steps to reach n when starting from j. For the 2-SAT algorithm, h_j is an upper bound on the expected number of steps to fully match S when starting from a truth assignment that matches S in j locations.

Clearly, $h_n = 0$ and $h_0 = h_1 + 1$, since from h_0 we always move to h_1 in one step. We use linearity of expectations to find an expression for other values of h_j. Let Z_j be a random variable representing the number of steps to reach n from state j. Now consider starting from state j, where $1 \le j \le n-1$. With probability $1/2$, the next state is $j-1$, and in this case $Z_j = 1 + Z_{j-1}$. With probability $1/2$, the next step is $j+1$, and in this case $Z_j = 1 + Z_{j+1}$. Hence

$$\mathbf{E}[Z_j] = \mathbf{E}\left[\frac{1}{2}(1 + Z_{j-1}) + \frac{1}{2}(1 + Z_{j+1})\right].$$

But $\mathbf{E}[Z_j] = h_j$ and so, by applying the linearity of expectations, we obtain

$$h_j = \frac{h_{j-1} + 1}{2} + \frac{h_{j+1} + 1}{2} = \frac{h_{j-1}}{2} + \frac{h_{j+1}}{2} + 1.$$

We therefore have the following system of equations:

$$\begin{aligned} h_n &= 0; \\ h_j &= \frac{h_{j-1}}{2} + \frac{h_{j+1}}{2} + 1, \quad 1 \le j \le n-1; \\ h_0 &= h_1 + 1. \end{aligned}$$

We can show inductively that, for $0 \le j \le n-1$,

$$h_j = h_{j+1} + 2j + 1.$$

It is true when $j = 0$, since $h_1 = h_0 - 1$. For other values of j, we use the equation

$$h_j = \frac{h_{j-1}}{2} + \frac{h_{j+1}}{2} + 1$$

to obtain

$$\begin{aligned} h_{j+1} &= 2h_j - h_{j-1} - 2 \\ &= 2h_j - (h_j + 2(j-1) + 1) - 2 \\ &= h_j - 2j - 1, \end{aligned}$$

using the induction hypothesis in the second line. We can conclude that

$$h_0 = h_1 + 1 = h_2 + 1 + 3 = \cdots = \sum_{i=0}^{n-1}(2i + 1) = n^2.$$

An alternative approach for solving the system of equations for the h_j is to guess and verify the solution $h_j = n^2 - j^2$. The system has $n + 1$ linearly independent equations and $n + 1$ unknowns, and hence there is a unique solution for each value of n. Therefore, if this solution satisfies the foregoing equations then it must be correct. We have $h_n = 0$. For $1 \le j \le n - 1$, we check

$$\begin{aligned} h_j &= \frac{n^2 - (j-1)^2}{2} + \frac{n^2 - (j+1)^2}{2} + 1 \\ &= n^2 - j^2 \end{aligned}$$

and

$$\begin{aligned} h_0 &= (n^2 - 1) + 1 \\ &= n^2. \end{aligned}$$

Thus we have proven the following fact.

Lemma 7.1: *Assume that a 2-SAT formula with n variables has a satisfying assignment and that the 2-SAT algorithm is allowed to run until it finds a satisfying assignment. Then the expected number of steps until the algorithm finds an assignment is at most n^2.*

We now return to the issue of dealing with unsatisfiable formulas by forcing the algorithm to stop after a fixed number of steps.

Theorem 7.2: *The 2-SAT algorithm always returns a correct answer if the formula is unsatisfiable. If the formula is satisfiable, then with probability at least $1 - 2^{-m}$ the algorithm returns a satisfying assignment. Otherwise, it incorrectly returns that the formula is unsatisfiable.*

Proof: It is clear that if there is no satisfying assignment then the algorithm correctly returns that the formula is unsatisfiable. Suppose the formula is satisfiable. Divide the execution of the algorithm into segments of $2n^2$ steps each. Given that no satisfying assignment was found in the first $i - 1$ segments, what is the conditional probability that the algorithm did not find a satisfying assignment in the ith segment? By Lemma 7.1, the expected time to find a satisfying assignment, regardless of its starting position, is bounded by n^2. Let Z be the number of steps from the start of segment i until the algorithm finds a satisfying assignment. Applying Markov's inequality,

$$\Pr(Z > 2n^2) \le \frac{n^2}{2n^2} = \frac{1}{2}.$$

Thus the probability that the algorithm fails to find a satisfying assignment after m segments is bounded above by $(1/2)^m$. ■

7.1.2. *Application: A Randomized Algorithm for 3-Satisfiability*

We now generalize the technique used to develop an algorithm for 2-SAT to obtain a randomized algorithm for 3-SAT. This problem is NP-complete, so it would be rather

3-SAT Algorithm:

1. Start with an arbitrary truth assignment.
2. Repeat up to m times, terminating if all clauses are satisfied:
   (a) Choose an arbitrary clause that is not satisfied.
   (b) Choose one of the literals uniformly at random, and change the value of the variable in the current truth assignment.
3. If a valid truth assignment has been found, return it.
4. Otherwise, return that the formula is unsatisfiable.

Algorithm 7.2: 3-SAT algorithm.

surprising if a randomized algorithm could solve the problem in expected time polynomial in n.[3] We present a randomized 3-SAT algorithm that solves 3-SAT in expected time that is exponential in n, but it is much more efficient than the naïve approach of trying all possible truth assignments for the variables.

Let us first consider the performance of a variant of the randomized 2-SAT algorithm when applied to a 3-SAT problem. The basic approach is the same as in the previous section; see Algorithm 7.2. In the algorithm, m is a parameter that controls the probability of success of the algorithm. We focus on bounding the expected time to reach a satisfying assignment (assuming one exists), as the argument of Theorem 7.2 can be extended once such a bound is found.

As in the analysis of the 2-SAT algorithm, assume that the formula is satisfiable and let S be a satisfying assignment. Let the assignment after i steps of the process be A_i, and let X_i be the number of variables in the current assignment A_i that match S. It follows from the same reasoning as for the 2-SAT algorithm that, for $1 \le j \le n-1$,

$$\Pr(X_{i+1} = j+1 \mid X_i = j) \ge 1/3;$$
$$\Pr(X_{i+1} = j-1 \mid X_i = j) \le 2/3.$$

These equations follow because at each step we choose an unsatisfied clause, so A_i and S must disagree on at least one variable in this clause. With probability at least $1/3$, we increase the number of matches between the current truth assignment and S. Again we can obtain an upper bound on the expected number of steps until $X_i = n$ by analyzing a Markov chain $Y_0, Y_1, \ldots$ such that $Y_0 = X_0$ and

$$\Pr(Y_{i+1} = 1 \mid Y_i = 0) = 1,$$
$$\Pr(Y_{i+1} = j+1 \mid Y_i = j) = 1/3,$$
$$\Pr(Y_{i+1} = j-1 \mid Y_i = j) = 2/3.$$

In this case, the chain is more likely to go down than up. If we let h_j be the expected number of steps to reach n when starting from j, then the following equations hold

[3] Technically, this would not settle the P = NP question, since we would be using a randomized algorithm and not a deterministic algorithm to solve an NP-hard problem. It would, however, have similar far-reaching implications about the ability to solve all NP-complete problems.

for h_j:

$$h_n = 0;$$
$$h_j = \frac{2h_{j-1}}{3} + \frac{h_{j+1}}{3} + 1, \quad 1 \le j \le n-1;$$
$$h_0 = h_1 + 1.$$

Again, these equations have a unique solution, which is given by

$$h_j = 2^{n+2} - 2^{j+2} - 3(n-j).$$

Alternatively, the solution can be found by using induction to prove the relationship

$$h_j = h_{j+1} + 2^{j+2} - 3.$$

We leave it as an exercise to verify that this solution indeed satisfies the foregoing equations.

The algorithm just described takes $\Theta(2^n)$ steps on average to find a satisfying assignment. This result is not very compelling, since there are only 2^n truth assignments to try! With some insight, however, we can significantly improve the process. There are two key observations.

1. If we choose an initial truth assignment *uniformly at random*, then the number of variables that match S has a binomial distribution with expectation $n/2$. With an exponentially small but nonnegligible probability, the process starts with an initial assignment that matches S in significantly more than $n/2$ variables.
2. Once the algorithm starts, it is more likely to move toward 0 than toward n. The longer we run the process, the more likely it has moved toward 0. Therefore, we are better off restarting the process with many randomly chosen initial assignments and running the process each time for a small number of steps, rather than running the process for many steps on the same initial assignment.

Based on these ideas, we consider the modified procedure of Algorithm 7.3. The modified algorithm has up to $3n$ steps to reach a satisfying assignment starting from a random assignment. If it fails to find a satisfying assignment in $3n$ steps, it restarts the search with a new randomly chosen assignment. We now determine how many times the process needs to restart before it reaches a satisfying assignment.

Let q represent the probability that the modified process reaches S (or some other satisfying assignment) in $3n$ steps starting with a truth assignment chosen uniformly at random. Let q_j be a lower bound on the probability that our modified algorithm reaches S (or some other satisfying assignment) when it starts with a truth assignment that includes exactly j variables that do not agree with S. Consider a particle moving on the integer line, with probability $1/3$ of moving up by one and probability $2/3$ of moving down by one. Notice that

$$\binom{j+2k}{k}\left(\frac{2}{3}\right)^k\left(\frac{1}{3}\right)^{j+k}$$

is the probability of exactly k moves down and $k+j$ moves up in a sequence of $j+2k$ moves. It is therefore a lower bound on the probability that the algorithm reaches a

Modified 3-SAT Algorithm:

1. Repeat up to m times, terminating if all clauses are satisfied:
   (a) Start with a truth assignment chosen uniformly at random.
   (b) Repeat the following up to $3n$ times, terminating if a satisfying assignment is found:
      i. Choose an arbitrary clause that is not satisfied.
      ii. Choose one of the literals uniformly at random, and change the value of the variable in the current truth assignment.
2. If a valid truth assignment has been found, return it.
3. Otherwise, return that the formula is unsatisfiable.

Algorithm 7.3: Modified 3-SAT algorithm.

satisfying assignment within $j + 2k \leq 3n$ steps, starting with an assignment that has exactly j variables that did not agree with S. That is,

$$q_j \geq \max_{k=0,\ldots,j} \binom{j+2k}{k} \left(\frac{2}{3}\right)^k \left(\frac{1}{3}\right)^{j+k}.$$

In particular, consider the case where $k = j$. In that case we have

$$q_j \geq \binom{3j}{j} \left(\frac{2}{3}\right)^j \left(\frac{1}{3}\right)^{2j}.$$

In order to approximate $\binom{3j}{j}$ we use Stirling's formula, which is similar to the bound of Eqn. (5.5) we have previously proven for factorials. Stirling's formula is tighter, which proves useful for this application. We use the following loose form.

Lemma 7.3 [Stirling's Formula]: *For $m > 0$,*

$$m! = \sqrt{2\pi m}\left(\frac{m}{e}\right)^m (1 \pm o(1)).$$

In particular, for $m > 0$,

$$\sqrt{2\pi m}\left(\frac{m}{\mathrm{e}}\right)^m \leq m! \leq 2\sqrt{2\pi m}\left(\frac{m}{\mathrm{e}}\right)^m.$$

Hence, when $j > 0$,

$$\begin{aligned}
\binom{3j}{j} &= \frac{(3j)!}{j!\,(2j)!} \\
&\geq \frac{\sqrt{2\pi(3j)}}{4\sqrt{2\pi j}\sqrt{2\pi(2j)}}\left(\frac{3j}{\mathrm{e}}\right)^{3j}\left(\frac{\mathrm{e}}{2j}\right)^{2j}\left(\frac{\mathrm{e}}{j}\right)^{j} \\
&= \frac{\sqrt{3}}{8\sqrt{\pi j}}\left(\frac{27}{4}\right)^j \\
&= \frac{c}{\sqrt{j}}\left(\frac{27}{4}\right)^j
\end{aligned}$$

for a constant $c = \sqrt{3}/8\sqrt{\pi}$. Thus, when $j > 0$,

$$\begin{aligned} q_j &\geq \binom{3j}{j}\left(\frac{2}{3}\right)^j\left(\frac{1}{3}\right)^{2j} \\ &\geq \frac{c}{\sqrt{j}}\left(\frac{27}{4}\right)^j\left(\frac{2}{3}\right)^j\left(\frac{1}{3}\right)^{2j} \\ &\geq \frac{c}{\sqrt{j}}\frac{1}{2^j}. \end{aligned}$$

Also, $q_0 = 1$.

Having established a lower bound for q_j, we can now derive a lower bound for q, the probability that the process reaches a satisfying assignment in $3n$ steps when starting with a random assignment:

$$\begin{aligned} q &\geq \sum_{j=0}^{n} \Pr(\text{a random assignment has } j \text{ mismatches with } S) \cdot q_j \\ &\geq \frac{1}{2^n} + \sum_{j=1}^{n}\binom{n}{j}\left(\frac{1}{2}\right)^n\frac{c}{\sqrt{j}}\frac{1}{2^j} \\ &\geq \frac{c}{\sqrt{n}}\left(\frac{1}{2}\right)^n\sum_{j=0}^{n}\binom{n}{j}\left(\frac{1}{2}\right)^j(1)^{n-j} \qquad (7.3) \\ &= \frac{c}{\sqrt{n}}\left(\frac{1}{2}\right)^n\left(\frac{3}{2}\right)^n \\ &= \frac{c}{\sqrt{n}}\left(\frac{3}{4}\right)^n, \end{aligned}$$

where in (7.3) we used $\sum_{j=0}^{n}\binom{n}{j}\left(\frac{1}{2}\right)^j(1)^{n-j} = \left(1+\frac{1}{2}\right)^n$.

Assuming that a satisfying assignment exists, the number of random assignments the process tries before finding a satisfying assignment is a geometric random variable with parameter q. The expected number of assignments tried is $1/q$, and for each assignment the algorithm uses at most $3n$ steps. Thus, the expected number of steps until a solution is found is bounded by $O(n^{3/2}(4/3)^n)$. As in the case of 2-SAT (Theorem 7.2), the modified 3-SAT algorithm (Algorithm 7.3) yields a Monte Carlo algorithm for the 3-SAT problem. If the expected number of steps until a satisfying solution is found is bounded above by a and if m is set to $2ab$, then the probability that no assignment is found when the formula is satisfiable is bounded above by 2^{-b}.

7.2. Classification of States

A first step in analyzing the long-term behavior of a Markov chain is to classify its states. In the case of a finite Markov chain, this is equivalent to analyzing the connectivity structure of the directed graph representing the Markov chain.

Definition 7.2: *State j is* accessible *from state i if, for some integer $n \geq 0$, $P^n_{i,j} > 0$. If two states i and j are accessible from each other, we say that they communicate and we write $i \leftrightarrow j$.*

In the graph representation of a chain, $i \leftrightarrow j$ if and only if there are directed paths connecting i to j and j to i.

The communicating relation defines an equivalence relation. That is, the communicating relation is

1. *reflexive* – for any state i, $i \leftrightarrow i$;
2. *symmetric* – if $i \leftrightarrow j$ then $j \leftrightarrow i$; and
3. *transitive* – if $i \leftrightarrow j$ and $j \leftrightarrow k$, then $i \leftrightarrow k$.

Proving this is left as Exercise 7.4. Thus, the communication relation partitions the states into disjoint equivalence classes, which we refer to as *communicating classes*. It might be possible to move from one class to another, but in that case it is impossible to return to the first class.

Definition 7.3: *A Markov chain is* irreducible *if all states belong to one communicating class.*

In other words, a Markov chain is irreducible if, for every pair of states, there is a nonzero probability that the first state can reach the second. We thus have the following lemma.

Lemma 7.4: *A finite Markov chain is irreducible if and only if its graph representation is a strongly connected graph.*

Next we distinguish between transient and recurrent states. Let $r^t_{i,j}$ denote the probability that, starting at state i, the first transition to state j occurs at time t; that is,

$$r^t_{i,j} = \Pr(X_t = j \text{ and, for } 1 \leq s \leq t-1,\ X_s \neq j \mid X_0 = i).$$

Definition 7.4: *A state is* recurrent *if $\sum_{t\geq 1} r^t_{i,i} = 1$, and it is* transient *if $\sum_{t\geq 1} r^t_{i,i} < 1$. A Markov chain is recurrent if every state in the chain is recurrent.*

If state i is recurrent then, once the chain visits that state, it will (with probability 1) eventually return to that state. Hence the chain will visit state i over and over again, infinitely often. On the other hand, if state i is transient then, starting at i, the chain will return to i with some fixed probability $p = \sum_{t\geq 1} r^t_{i,i}$. In this case, the number of times the chain visits i when starting at i is given by a geometric random variable. If one state in a communicating class is transient (respectively, recurrent) then all states in that class are transient (respectively, recurrent); proving this is left as Exercise 7.5.

We denote the expected time to return to state i when starting at state i by $h_{i,i} = \sum_{t\geq 1} t \cdot r^t_{i,i}$. Similarly, for any pair of states i and j, we denote by $h_{i,j} = \sum_{t\geq 1} t \cdot r^t_{i,j}$ the expected time to first reach j from state i. It may seem that if a chain is recurrent, so that we visit a state i infinitely often, then $h_{i,i}$ should be finite. This is not the case, which leads us to the following definition.

Definition 7.5: *A recurrent state i is* positive recurrent *if $h_{i,i} < \infty$. Otherwise, it is* null recurrent.

To give an example of a Markov chain that has null recurrent states, consider a chain whose states are the positive integers. From state i, the probability of going to state $i+1$ is $i/(i+1)$. With probability $1/(i+1)$, the chain returns to state 1. Starting at state 1, the probability of not having returned to state 1 within the first t steps is thus

$$\prod_{j=1}^{t} \frac{j}{j+1} = \frac{1}{t+1}.$$

Hence the probability of never returning to state 1 from state 1 is 0, and state 1 is recurrent. It follows that

$$r_{1,1}^{t} = \frac{1}{t(t+1)}.$$

However, the expected number of steps until the first return to state 1 from state 1 is

$$h_{1,1} = \sum_{t=1}^{\infty} t \cdot r_{1,1}^{t} = \sum_{t=1}^{\infty} \frac{1}{t+1},$$

which is unbounded.

In the foregoing example the Markov chain had an infinite number of states. This is necessary for null recurrent states to exist. The proof of the following important lemma is left as Exercise 7.16.

Lemma 7.5: *In a finite Markov chain:*

1. *at least one state is recurrent; and*
2. *all recurrent states are positive recurrent.*

Finally, for our later study of limiting distributions of Markov chains we will need to define what it means for a state to be aperiodic. As an example of periodicity, consider a random walk whose states are the positive integers. When at state i, with probability $1/2$ the chain moves to $i+1$ and with probability $1/2$ the chain moves to $i-1$. If the chain starts at state 0, then it can be at an even-numbered state only after an even number of moves, and it can be at an odd-numbered state only after an odd number of moves. This is an example of periodic behavior.

Definition 7.6: *A state j in a discrete time Markov chain is* periodic *if there exists an integer $\Delta > 1$ such that $\Pr(X_{t+s} = j \mid X_t = j) = 0$ unless s is divisible by Δ. A discrete time Markov chain is periodic if any state in the chain is periodic. A state or chain that is not periodic is* aperiodic.

In our example, every state in the Markov chain is periodic because, for every state j, $\Pr(X_{t+s} = j \mid X_t = j) = 0$ unless s is divisible by 2.

We end this section with an important corollary about the behavior of finite Markov chains.

Definition 7.7: *An aperiodic, positive recurrent state is an* ergodic *state. A Markov chain is ergodic if all its states are ergodic.*

Corollary 7.6: *Any finite, irreducible, and aperiodic Markov chain is an ergodic chain.*

Proof: A finite chain has at least one recurrent state by Lemma 7.5, and if the chain is irreducible then all of its states are recurrent. In a finite chain, all recurrent states are positive recurrent by Lemma 7.5 and thus all the states of the chain are positive recurrent and aperiodic. The chain is therefore ergodic. ∎

7.2.1. *Example: The Gambler's Ruin*

When a Markov chain has more than one class of recurrent states, we are often interested in the probability that the process will enter and thus be *absorbed* by a given communicating class.

For example, consider a sequence of independent, fair gambling games between two players. In each round a player wins a dollar with probability $1/2$ or loses a dollar with probability $1/2$. The state of the system at time t is the number of dollars won by player 1. If player 1 has lost money, this number is negative. The initial state is 0.

It is reasonable to assume that there are numbers ℓ_1 and ℓ_2 such that player i cannot lose more than ℓ_i dollars, and thus the game ends when it reaches one of the two states $-\ell_1$ or ℓ_2. At this point, one of the gamblers is ruined; that is, he has lost all his money. To conform with the formalization of a Markov chain, we assume that for each of these two end states there is only one transition out and that it goes back to the same state. This gives us a Markov chain with two absorbing, recurrent states.

What is the probability that player 1 wins ℓ_2 dollars before losing ℓ_1 dollars? If $\ell_2 = \ell_1$, then by symmetry this probability must be $1/2$. We provide a simple argument for the general case using the classification of the states.

Clearly $-\ell_1$ and ℓ_2 are recurrent states. All other states are transient, since there is a nonzero probability of moving from each of these states to either state $-\ell_1$ or state ℓ_2.

Let P_i^t be the probability that, after t steps, the chain is at state i. For $-\ell_1 < i < \ell_2$, state i is transient and so $\lim_{t\to\infty} P_i^t = 0$.

Let q be the probability that the game ends with player 1 winning ℓ_2 dollars, so that the chain was absorbed into state ℓ_2. Then $1 - q$ is the probability the chain was absorbed into state $-\ell_1$. By definition,

$$\lim_{t\to\infty} P_{\ell_2}^t = q.$$

Since each round of the gambling game is fair, the expected gain of player 1 in each step is 0. Let W^t be the gain of player 1 after t steps. Then $\mathrm{E}[W^t] = 0$ for any t by induction. Thus,

$$\mathrm{E}[W^t] = \sum_{i=-\ell_1}^{\ell_2} iP_i^t = 0$$

and

$$\lim_{t\to\infty} \mathbf{E}[W^t] = \ell_2 q - \ell_1(1-q)$$
$$= 0.$$

Thus,

$$q = \frac{\ell_1}{\ell_1 + \ell_2}.$$

That is, the probability of winning (or losing) is proportional to the amount of money a player is willing to lose (or win).

Another approach that yields the same answer is to let q_j represent the probability that player 1 wins ℓ_2 dollars before losing ℓ_1 dollars when having won j dollars for $-\ell_1 \le j \le \ell_2$. Clearly, $q_{-\ell_1} = 0$ and $q_{\ell_2} = 1$. For $-\ell_1 < j < \ell_2$, we compute by considering the outcome of the first game:

$$q_j = \frac{q_{j-1}}{2} + \frac{q_{j+1}}{2}.$$

We have $\ell_2 + \ell_1 - 2$ linearly independent equations and $\ell_2 + \ell_1 - 2$ unknowns, so there is a unique solution to this set of equations. It is easy to verify that $q_j = (\ell_1 + j)/(\ell_1 + \ell_2)$ satisfies the given equations.

In Exercise 7.20, we consider the question of what happens if, as is generally the case in real life, one player is at a disadvantage and so is slightly more likely to lose than to win any single game.

7.3. Stationary Distributions

Recall that if $\mathbf{P}$ is the one-step transition probability matrix of a Markov chain and if $\bar{p}(t)$ is the probability distribution of the state of the chain at time t, then

$$\bar{p}(t+1) = \bar{p}(t)\mathbf{P}.$$

Of particular interest are state probability distributions that do not change after a transition.

Definition 7.8: *A* stationary distribution *(also called an* equilibrium distribution*) of a Markov chain is a probability distribution $\bar{\pi}$ such that*

$$\bar{\pi} = \bar{\pi}\mathbf{P}.$$

If a chain ever reaches a stationary distribution then it maintains that distribution for all future time, and thus a stationary distribution represents a steady state or an equilibrium in the chain's behavior. Stationary distributions play a key role in analyzing Markov chains. The fundamental theorem of Markov chains characterizes chains that converge to stationary distributions.

We discuss first the case of finite chains and then extend the results to any discrete space chain. Without loss of generality, assume that the finite set of states of the Markov chain is $\{0, 1, \ldots, n\}$.

Theorem 7.7: *Any finite, irreducible, and ergodic Markov chain has the following properties:*

1. *the chain has a unique stationary distribution* $\bar{\pi} = (\pi_0, \pi_1, \ldots, \pi_n)$*;*
2. *for all j and i, the limit* $\lim_{t\to\infty} P^t_{j,i}$ *exists and it is independent of j;*
3. $\pi_i = \lim_{t\to\infty} P^t_{j,i} = 1/h_{i,i}$.

Under the conditions of this theorem, the stationary distribution $\bar{\pi}$ has two interpretations. First, π_i is the limiting probability that the Markov chain will be in state i infinitely far out in the future, and this probability is independent of the initial state. In other words, if we run the chain long enough, the initial state of the chain is almost forgotten and the probability of being in state i converges to π_i. Second, π_i is the inverse of $h_{i,i} = \sum_{t=1}^{\infty} t \cdot r^t_{i,i}$, the expected number of steps for a chain starting in state i to return to i. This stands to reason; if the average time to return to state i from i is $h_{i,i}$, then we expect to be in state i for $1/h_{i,i}$ of the time and thus, in the limit, we must have $\pi_i = 1/h_{i,i}$.

Proof of Theorem 7.7: We prove the theorem using the following result, which we state without proof. ■

Lemma 7.8: *For any irreducible, ergodic Markov chain and for any state i, the limit* $\lim_{t\to\infty} P^t_{i,i}$ *exists and*

$$\lim_{t\to\infty} P^t_{i,i} = \frac{1}{h_{i,i}}.$$

This lemma is a corollary of a basic result in renewal theory. We give an informal justification for Lemma 7.8: the expected time between visits to i is $h_{i,i}$, and therefore state i is visited $1/h_{i,i}$ of the time. Thus $\lim_{t\to\infty} P^t_{i,i}$, which represents the probability a state chosen far in the future is at state i when the chain starts at state i, must be $1/h_{i,i}$.

Using the fact that $\lim_{t\to\infty} P^t_{i,i}$ exists, we now show that, for any j and i,

$$\lim_{t\to\infty} P^t_{j,i} = \lim_{t\to\infty} P^t_{i,i} = \frac{1}{h_{i,i}};$$

that is, these limits exist and are independent of the starting state j.

Recall that $r^t_{j,i}$ is the probability that starting at j, the chain first visits i at time t. Since the chain is irreducible we have that $\sum_{t=1}^{\infty} r^t_{j,i} = 1$, and for any $\varepsilon > 0$ there exists (a finite) $t_1 = t_1(\varepsilon)$ such that $\sum_{t=1}^{t_1} r^t_{j,i} \geq 1 - \varepsilon$.

For $j \neq i$, we have

$$P^t_{j,i} = \sum_{k=1}^{t} r^k_{j,i} P^{t-k}_{i,i}.$$

For $t \geq t_1$,

$$\sum_{k=1}^{t_1} r^k_{j,i} P^{t-k}_{i,i} \leq \sum_{k=1}^{t} r^k_{j,i} P^{t-k}_{i,i} = P^t_{j,i}.$$

Using the facts that $\lim_{t\to\infty} P^t_{i,i}$ exists and t_1 is finite, we have

$$\begin{aligned}
\lim_{t\to\infty} P^t_{j,i} &\geq \lim_{t\to\infty} \sum_{k=1}^{t_1} r^k_{j,i} P^{t-k}_{i,i} \\
&= \sum_{k=1}^{t_1} r^k_{j,i} \lim_{t\to\infty} P^t_{i,i} \\
&= \lim_{t\to\infty} P^t_{i,i} \sum_{k=1}^{t_1} r^k_{j,i} \\
&\geq (1-\varepsilon) \lim_{t\to\infty} P^t_{i,i}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
P^t_{j,i} &= \sum_{k=1}^{t} r^k_{j,i} P^{t-k}_{i,i} \\
&\leq \sum_{k=1}^{t_1} r^k_{j,i} P^{t-k}_{i,i} + \varepsilon,
\end{aligned}$$

from which we can deduce that

$$\begin{aligned}
\lim_{t\to\infty} P^t_{j,i} &\leq \lim_{t\to\infty} \left(\sum_{k=1}^{t_1} r^k_{j,i} P^{t-k}_{i,i} + \varepsilon \right) \\
&= \sum_{k=1}^{t_1} r^k_{j,i} \lim_{t\to\infty} P^{t-k}_{i,i} + \varepsilon \\
&\leq \lim_{t\to\infty} P^t_{i,i} + \varepsilon.
\end{aligned}$$

Letting ε approach 0, we have proven that, for any pair i and j,

$$\lim_{t\to\infty} P^t_{j,i} = \lim_{t\to\infty} P^t_{i,i} = \frac{1}{h_{i,i}}.$$

Now let

$$\pi_i = \lim_{t\to\infty} P^t_{j,i} = \frac{1}{h_{i,i}}.$$

We show that $\bar{\pi} = (\pi_0, \pi_1, \dots)$ forms a stationary distribution.

For every $t \geq 0$, we have $P^t_{i,i} \geq 0$ and thus $\pi_i \geq 0$. For any $t \geq 0$, $\sum_{i=0}^{n} P^t_{j,i} = 1$ and thus

$$\lim_{t\to\infty} \sum_{i=0}^{n} P^t_{j,i} = \sum_{i=0}^{n} \lim_{t\to\infty} P^t_{j,i} = \sum_{i=0}^{n} \pi_i = 1,$$

and $\bar{\pi}$ is a proper distribution. Now,

$$P^{t+1}_{j,i} = \sum_{k=0}^{n} P^t_{j,k} P_{k,i}.$$

Letting $t \to \infty$, we have

$$\pi_i = \sum_{k=0}^{n} \pi_k P_{k,i},$$

proving that $\bar{\pi}$ is a stationary distribution.

Suppose there were another stationary distribution $\bar{\phi}$. Then by the same argument we would have

$$\phi_i = \sum_{k=0}^{n} \phi_k P^t_{k,i},$$

and taking the limit as $t \to \infty$ yields

$$\phi_i = \sum_{k=0}^{n} \phi_k \pi_i = \pi_i \sum_{k=0}^{n} \phi_k.$$

Since $\sum_{k=0}^{n} \phi_k = 1$ it follows that $\phi_i = \pi_i$ for all i, or $\bar{\phi} = \bar{\pi}$. ■

It is worth making a few remarks about Theorem 7.7. First, the requirement that the Markov chain be aperiodic is not necessary for the existence of a stationary distribution. In fact, any finite Markov chain has a stationary distribution; but in the case of a periodic state i, the stationary probability π_i is not the limiting probability of being in i but instead just the long-term frequency of visiting state i. Second, any finite chain has at least one component that is recurrent. Once the chain reaches a recurrent component, it cannot leave that component. Thus, the subchain that corresponds to that component is irreducible and recurrent, and the limit theorem applies to any aperiodic recurrent component of the chain.

One way to compute the stationary distribution of a finite Markov chain is to solve the system of linear equations

$$\bar{\pi}\mathbf{P} = \bar{\pi}.$$

This is particularly useful if one is given a specific chain. For example, given the transition matrix

$$\mathbf{P} = \begin{bmatrix} 0 & 1/4 & 0 & 3/4 \\ 1/2 & 0 & 1/3 & 1/6 \\ 1/4 & 1/4 & 1/2 & 0 \\ 0 & 1/2 & 1/4 & 1/4 \end{bmatrix},$$

we have five equations for the four unknowns π_0, π_1, π_2, and π_3 given by $\bar{\pi}\mathbf{P} = \bar{\pi}$ and $\sum_{i=0}^{3} \pi_i = 1$. The equations have a unique solution.

Another useful technique is to study the cut-sets of the Markov chain. For any state i of the chain,

$$\sum_{j=0}^{n} \pi_j P_{j,i} = \pi_i = \pi_i \sum_{j=0}^{n} P_{i,j}$$

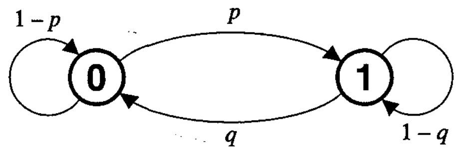


Figure 7.2: A simple Markov chain used to represent bursty behavior.

or

$$\sum_{j \neq i} \pi_j P_{j,i} = \sum_{j \neq i} \pi_i P_{i,j}.$$

That is, in the stationary distribution the probability that a chain leaves a state equals the probability that it enters the state. This observation can be generalized to sets of states as follows.

Theorem 7.9: *Let S be a set of states of a finite, irreducible, aperiodic Markov chain. In the stationary distribution, the probability that the chain leaves the set S equals the probability that it enters S.*

In other words, if C is a cut-set in the graph representation of the chain, then in the stationary distribution the probability of crossing the cut-set in one direction is equal to the probability of crossing the cut-set in the other direction.

A basic but useful Markov chain that serves as an example of cut-sets is given in Figure 7.2. The chain has only two states. From state 0, you move to state 1 with probability p and stay at state 0 with probability $1 - p$. Similarly, from state 1 you move to state 0 with probability q and remain in state 1 with probability $1 - q$. This Markov chain is often used to represent bursty behavior. For example, when bits are corrupted in transmissions they are often corrupted in large blocks, since the errors are often caused by an external phenomenon of some duration. In this setting, being in state 0 after t steps represents that the tth bit was sent successfully, while being in state 1 represents that the bit was corrupted. Blocks of successfully sent bits and corrupted bits both have lengths that follow a geometric distribution. When p and q are small, state changes are rare, and the bursty behavior is modeled.

The transition matrix is

$$\mathbf{P} = \begin{bmatrix} 1-p & p \\ q & 1-q \end{bmatrix}.$$

Solving $\bar{\pi}\mathbf{P} = \bar{\pi}$ corresponds to solving the following system of three equations:

$$\pi_0(1 - p) + \pi_1 q = \pi_0;$$
$$\pi_0 p + \pi_1(1 - q) = \pi_1;$$
$$\pi_0 + \pi_1 = 1.$$

The second equation is redundant, and the solution is $\pi_0 = q/(p+q)$ and $\pi_1 = p/(p+q)$. For example, with the natural parameters $p = 0.005$ and $q = 0.1$, in the stationary distribution more than 95% of the bits are received uncorrupted.

Using the cut-set formulation, we have that in the stationary distribution the probability of leaving state 0 must equal the probability of entering state 0, or

$$\pi_0 p = \pi_1 q.$$

Again, now using $\pi_0 + \pi_1 = 1$ yields $\pi_0 = q/(p+q)$ and $\pi_1 = p/(p+q)$.

Finally, for some Markov chains the stationary distribution is easy to compute by means of the following theorem.

Theorem 7.10: *Consider a finite, irreducible, and ergodic Markov chain with transition matrix* **P**. *If there are nonnegative numbers $\bar{\pi} = (\pi_0, \dots, \pi_n)$ such that $\sum_{i=0}^{n} \pi_i = 1$ and if, for any pair of states i, j,*

$$\pi_i P_{i,j} = \pi_j P_{j,i},$$

then $\bar{\pi}$ is the stationary distribution corresponding to **P**.

Proof: Consider the jth entry of $\bar{\pi}\mathbf{P}$. Using the assumption of the theorem, we find that it equals

$$\sum_{i=0}^{n} \pi_i P_{i,j} = \sum_{i=0}^{n} \pi_j P_{j,i} = \pi_j.$$

Thus $\bar{\pi}$ satisfies $\bar{\pi} = \bar{\pi}\mathbf{P}$. Since $\sum_{i=0}^{n} \pi_i = 1$, it follows from Theorem 7.7 that $\bar{\pi}$ must be the unique stationary distribution of the Markov chain. ■

Chains that satisfy the condition

$$\pi_i P_{i,j} = \pi_j P_{j,i}$$

are called *time reversible*; Exercise 7.13 helps explain why. You may check that the chain of Figure 7.2 is time reversible.

We turn now to the convergence of Markov chains with countably infinite state spaces. Using essentially the same technique as in the proof of Theorem 7.7, one can prove the next result.

Theorem 7.11: *Any irreducible aperiodic Markov chain belongs to one of the following two categories:*

1. *the chain is ergodic – for any pair of states i and j, the limit $\lim_{t\to\infty} P^t_{j,i}$ exists and is independent of j, and the chain has a unique stationary distribution $\pi_i = \lim_{t\to\infty} P^t_{j,i} > 0$; or*
2. *no state is positive recurrent – for all i and j, $\lim_{t\to\infty} P^t_{j,i} = 0$, and the chain has no stationary distribution.*

Cut-sets and the property of time reversibility can also be used to find the stationary distribution for Markov chains with countably infinite state spaces.

7.3.1. *Example: A Simple Queue*

A *queue* is a line where customers wait for service. We examine a model for a bounded queue where time is divided into steps of equal length. At each time step, exactly one of the following occurs.

- If the queue has fewer than n customers, then with probability λ a new customer joins the queue.
- If the queue is not empty, then with probability μ the head of the line is served and leaves the queue.
- With the remaining probability, the queue is unchanged.

If X_t is the number of customers in the queue at time t, then under the foregoing rules the X_t yield a finite-state Markov chain. Its transition matrix has the following nonzero entries:

$$\begin{aligned}
P_{i,i+1} &= \lambda \quad \text{if } i < n;\\
P_{i,i-1} &= \mu \quad \text{if } i > 0;\\
P_{i,i} &= \begin{cases} 1-\lambda & \text{if } i = 0,\\ 1-\lambda-\mu & \text{if } 1 \le i \le n-1,\\ 1-\mu & \text{if } i = n. \end{cases}
\end{aligned}$$

The Markov chain is irreducible, finite, and aperiodic, so it has a unique stationary distribution $\bar{\pi}$. We use $\bar{\pi} = \bar{\pi}\mathbf{P}$ to write

$$\begin{aligned}
\pi_0 &= (1-\lambda)\pi_0 + \mu\pi_1,\\
\pi_i &= \lambda\pi_{i-1} + (1-\lambda-\mu)\pi_i + \mu\pi_{i+1}, \quad 1 \le i \le n-1,\\
\pi_n &= \lambda\pi_{n-1} + (1-\mu)\pi_n.
\end{aligned}$$

It is easy to verify that

$$\pi_i = \pi_0 \left(\frac{\lambda}{\mu}\right)^i$$

is a solution to the preceding system of equations. Adding the requirement $\sum_{i=0}^{n} \pi_i = 1$, we have

$$\sum_{i=0}^{n} \pi_i = \sum_{i=0}^{n} \pi_0 \left(\frac{\lambda}{\mu}\right)^i = 1$$

or

$$\pi_0 = \frac{1}{\sum_{i=0}^{n} (\lambda/\mu)^i}.$$

For all $0 \le i \le n$,

$$\pi_i = \frac{(\lambda/\mu)^i}{\sum_{i=0}^{n} (\lambda/\mu)^i}. \tag{7.4}$$

Another way to compute the stationary probability in this case is to use cut-sets. For any i, the transitions $i \to i+1$ and $i+1 \to i$ constitute a cut-set of the graph representing the Markov chain. Thus, in the stationary distribution, the probability of moving from state i to state $i+1$ must be equal to the probability of moving from state $i+1$ to i, or

$$\lambda\pi_i = \mu\pi_{i+1}.$$

A simple induction now yields

$$\pi_i = \pi_0\left(\frac{\lambda}{\mu}\right)^i.$$

In the case where there is no upper limit n on the number of customers in a queue, the Markov chain is no longer finite. The Markov chain has a countably infinite state space. Applying Theorem 7.11, the Markov chain has a stationary distribution if and only if the following set of linear equations has a solution with all $\pi_i > 0$:

$$\begin{aligned} \pi_0 &= (1-\lambda)\pi_0 + \mu\pi_1; \\ \pi_i &= \lambda\pi_{i-1} + (1-\lambda-\mu)\pi_i + \mu\pi_{i+1}, \quad i \geq 1. \end{aligned} \tag{7.5}$$

It is easy to verify that

$$\pi_i = \frac{(\lambda/\mu)^i}{\sum_{i=0}^{\infty}(\lambda/\mu)^i} = \left(\frac{\lambda}{\mu}\right)^i\left(1-\frac{\lambda}{\mu}\right)$$

is a solution of the system of equations (7.5). This naturally generalizes the solution to the case where there is an upper bound n on the number of the customers in the system given in Eqn. (7.4). All of the π_i are greater than 0 if and only if $\lambda < \mu$, which corresponds to the situation when the rate at which customers arrive is lower than the rate at which they are served. If $\lambda > \mu$, then the rate at which customers arrive is higher than the rate at which they depart. Hence there is no stationary distribution, and the queue length will become arbitrarily long. In this case, each state in the Markov chain is transient. The case of $\lambda = \mu$ is more subtle. Again, there is no stationary distribution and the queue length will become arbitrarily long, but now the states are null recurrent. (See the related Exercise 7.17.)

7.4. Random Walks on Undirected Graphs

A random walk on an undirected graph is a special type of Markov chain that is often used in analyzing algorithms. Let $G = (V, E)$ be a finite, undirected, and connected graph.

Definition 7.9: *A* random walk *on G is a Markov chain defined by the sequence of moves of a particle between vertices of G. In this process, the place of the particle at a given time step is the state of the system. If the particle is at vertex i and if i has $d(i)$ outgoing edges, then the probability that the particle follows the edge (i, j) and moves to a neighbor j is $1/d(i)$.*

We have already seen an example of such a walk when we analyzed the randomized 2-SAT algorithm.

For a random walk on an undirected graph, we have a simple criterion for aperiodicity as follows.

Lemma 7.12: *A random walk on an undirected graph G is aperiodic if and only if G is not bipartite.*

Proof: A graph is bipartite if and only if it does not have cycles with an odd number of edges. In an undirected graph, there is always a path of length 2 from a vertex to itself. If the graph is bipartite then the random walk is periodic with period $d = 2$. If the graph is not bipartite then it has an odd cycle, and by traversing that cycle we have an odd-length path from any vertex to itself. It follows that the Markov chain is aperiodic. ■

For the remainder of this section we assume that G is not bipartite. A random walk on a finite, undirected, connected, and non-bipartite graph G satisfies the conditions of Theorem 7.7, and hence the random walk converges to a stationary distribution. We show that this distribution depends only on the degree sequence of the graph.

Theorem 7.13: *A random walk on G converges to a stationary distribution $\bar{\pi}$, where*

$$\pi_v = \frac{d(v)}{2|E|}.$$

Proof: Since $\sum_{v\in V} d(v) = 2|E|$, it follows that

$$\sum_{v\in V} \pi_v = \sum_{v\in V} \frac{d(v)}{2|E|} = 1,$$

and $\bar{\pi}$ is a proper distribution over $v \in V$.

Let $\mathbf{P}$ be the transition probability matrix of the Markov chain. Let $N(v)$ represent the neighbors of v. The relation $\bar{\pi} = \bar{\pi}\mathbf{P}$ is equivalent to

$$\pi_v = \sum_{u\in N(v)} \frac{d(u)}{2|E|}\frac{1}{d(u)} = \frac{d(v)}{2|E|},$$

and the theorem follows. ■

Recall that we have used $h_{u,v}$ to denote the expected time to reach state v when starting at state u. The value $h_{u,v}$ is often referred to as the *hitting time* from u to v, or just the hitting time where the meaning is clear. Another value related to the hitting time is the *commute time* between u and v, given by $h_{u,v} + h_{v,u}$. Unlike the hitting time, the commute time is symmetric; it represents the time to go from u to v and back to u, and this is the same as the time to go from v to u and back to v. Finally, for random walks on graphs, we are also interested in a quantity called the cover time.

Definition 7.10: *The* cover time *of a graph $G = (V, E)$ is the maximum over all vertices $v \in V$ of the expected time to visit all of the nodes in the graph by a random walk starting from v.*

We consider here some basic bounds on the commute time and the cover time for standard random walks on a finite, undirected, connected graph $G = (V, E)$.

Lemma 7.14: *If $(u, v) \in E$, the commute time $h_{u,v} + h_{v,u}$ is at most $2|E|$.*

Proof: Let D be a set of directed edges such that for every edge $(u, v) \in E$ we have the two directed edges $u \to v$ and $v \to u$ in D. We can view the random walk on G as a Markov chain with state space D, where the state of the Markov chain at time t is the directed edge taken by the random walk in its tth transition. The Markov chain has $2|E|$ states and it is easy to verify that it has a uniform stationary distribution. (This is left as Exercise 7.29.) Since the stationary probability of being in state $u \to v$ is $1/2|E|$, once the original random walk traverses the directed edge $u \to v$ the expected time to traverse that directed edge again is $2|E|$. Because the random walk is memoryless, once it reaches vertex v we can "forget" that it reached it through the edge $u \to v$, and therefore the expected time starting at v to reach u and then traverse the edge $u \to v$ back to v is bounded above by $2|E|$. As this is only one of the possible ways to go from v to u and back to v, we have shown that $h_{v,u} + h_{u,v} \le 2|E|$. ∎

Lemma 7.15: *The cover time of $G = (V, E)$ is bounded above by $2|E|(|V| - 1)$.*

Proof: Choose a spanning tree T of G; that is, choose any subset of the edges that gives an acyclic graph connecting all the vertices of G. Starting from any vertex v, there exists a cyclic (Eulerian) tour on the spanning tree in which every edge is traversed once in each direction; for example, such a tour can be found by considering the sequence of vertices passed through by a depth first search. The maximum expected time to go through the vertices in the tour, where the maximum is over the choice of starting vertex, is an upper bound on the cover time. Let $v_0, v_1, \ldots, v_{2|V|-2}$ be the sequence of vertices in the tour starting from $v_0 = v$. Then the expected time to go through all the vertices in sequence order is

$$\sum_{i=0}^{2|V|-3} h_{v_i, v_{i+1}} = \sum_{(x,y)\in T} (h_{x,y} + h_{y,x}) \le 2|E|(|V| - 1).$$

In words, the commute time for every pair of adjacent vertices in the tree is bounded above by $2|E|$, and there are $|V| - 1$ pairs of adjacent vertices. ∎

The following result is known as Matthews' theorem, which relates the cover time of a graph to the hitting time. Recall that we use $H(n)$ to denote the harmonic number $\sum_{i=1}^{n} 1/i \approx \ln n$.

Lemma 7.16: *The cover time C_G of $G = (V, E)$ with n vertices is bounded by*

$$C_G \le H(n-1) \max_{u,v\in V:u\neq v} h_{u,v}.$$

Proof: For convenience let $B = \max_{u,v\in V:u\neq v} h_{u,v}$. Consider a random walk starting from a vertex u. We choose an ordering of the vertices according to a uniform permutation; let $Z_1, Z_2, \ldots, Z_n$ be the ordering. Let T_j be the first time when all of the first j vertices in the order, $Z_1, Z_2, \ldots, Z_j$, have been visited, and let A_j be the last vertex from the set $\{Z_1, \ldots, Z_j\}$ that was visited. Following the spirit of the coupon collector's analysis, we consider the successive time intervals $T_j - T_{j-1}$. If the chain's history is given by $X_1, X_2, \ldots$, then in particular for $j \geq 2$ we consider

$$Y_j = \mathbf{E}[T_j - T_{j-1} \mid Z_1, \ldots, Z_j; X_1, \ldots, X_{T_{j-1}}].$$

The expected time to cover the graph starting from u is

$$\sum_{j=2}^{n} Y_j + \mathbf{E}[T_1].$$

If Z_1 is chosen to be u, which happens with probability $1/n$, then T_1 is 0. Otherwise, $\mathbf{E}[T_1 \mid Z_1] = h_{u,Z_1} \leq B$. Hence $\mathbf{E}[T_1] \leq (1 - 1/n)B$.

For the Y_j, there are two cases to consider. If Z_j is not the last vertex seen from the set $\{Z_1, Z_2, \ldots, Z_j\}$, then Y_j is 0, since $T_j = T_{j-1}$ in that case. If Z_j is the last vertex seen from this set, then, regardless of the rest of the history of the chain, $Y_j \leq B$, since Y_j is the hitting time h_{Z_k,Z_j} for the Z_k that was visited last out of $\{Z_1, Z_2, \ldots, Z_{j-1}\}$. As the Z_j were chosen according to a random permutation, independent of the random walk, we have Z_j is last out of the set $\{Z_1, Z_2, \ldots, Z_j\}$ with probability $1/j$. It follows that

$$\begin{aligned}
\sum_{j=2}^{n} Y_j + \mathbf{E}[T_1] &\leq \sum_{j=2}^{n} \frac{1}{j}B + \left(1 - \frac{1}{n}\right)B \\
&= \left(1 + \sum_{j=2}^{n} \frac{1}{j}\right)B - \frac{1}{n}B \\
&= H(n-1)B.
\end{aligned}$$

Since this holds for every starting vertex u, the lemma is proven. ■

One can similarly obtain lower bounds using the same technique. A natural lower bound is

$$C_G \geq H(n-1) \min_{u,v \in V : u \neq v} h_{u,v}.$$

However, the minimum hitting time can be very small for some graphs, making this bound less useful. In some cases, the lower bound can be made stronger by considering a subset of vertices $V' \subset V$. In this case, the proof can be modified to give

$$C_G \geq H(|V'| - 1) \min_{u,v \in V' : u \neq v} h_{u,v}.$$

The term from the harmonic series is smaller, but the minimum hitting time used in the bound may correspondingly be larger.

7.4.1. *Application: An s–t Connectivity Algorithm*

Suppose we are given an undirected graph $G = (V, E)$ and two vertices s and t in G. Let $n = |V|$ and $m = |E|$. We want to determine if there is a path connecting s and t. This is easily done in linear time using a standard breadth-first search or depth-first search. Such algorithms, however, require $\Omega(n)$ space.

s–t Connectivity Algorithm:

1. Start a random walk from s.
2. If the walk reaches t within $2n^3$ steps, return that there is a path. Otherwise, return that there is no path.

Algorithm 7.4: s–t Connectivity algorithm.

Here we develop a randomized algorithm that works with only $O(\log n)$ bits of memory. This could be even less than the number of bits required to write the path between s and t. The algorithm is simple: perform a random walk on G for enough steps so that a path from s to t is likely to be found. We use the cover time result (Lemma 7.16) to bound the number of steps that the random walk has to run. For convenience, assume that the graph G has no bipartite connected components, so that the results of Theorem 7.13 apply to any connected component of G. (The results can be made to apply to bipartite graphs with some additional technical work.)

Theorem 7.17: *The s–t connectivity algorithm (Algorithm 7.4) returns the correct answer with probability $1/2$, and it only errs by returning that there is no path from s to t when there is such a path.*

Proof: If there is no path then the algorithm returns the correct answer. If there is a path, the algorithm errs if it does not find the path within $2n^3$ steps of the walk. The expected time to reach t from s (if there is a path) is bounded from above by the cover time of their shared component, which by Lemma 7.15 is at most $2nm < n^3$. By Markov's inequality, the probability that a walk takes more than $2n^3$ steps to reach t from s is at most $1/2$. ∎

The algorithm must keep track of its current position, which takes $O(\log n)$ bits, as well as the number of steps taken in the random walk, which also takes only $O(\log n)$ bits (since we count up only to $2n^3$). As long as there is some mechanism for choosing a random neighbor from each vertex, this is all the memory required.

7.5. Parrondo's Paradox

Parrondo's paradox provides an interesting example of the analysis of Markov chains while also demonstrating a subtlety in dealing with probabilities. The paradox appears to contradict the old saying that two wrongs don't make a right, showing that two losing games can be combined to make a winning game. Because Parrondo's paradox can be analyzed in many different ways, we will go over several approaches to the problem.

First, consider game A, in which we repeatedly flip a biased coin (call it coin a) that comes up heads with probability $p_a < 1/2$ and tails with probability $1 - p_a$. You win a dollar if the coin comes up heads and lose a dollar if it comes up tails. Clearly, this is a losing game for you. For example, if $p_a = 0.49$, then your expected loss is two cents per game.

In game B, we also repeatedly flip coins, but the coin that is flipped depends on how you have been doing so far in the game. Let w be the number of your wins so far and ℓ the number of your losses. Each round we bet one dollar, so $w - \ell$ represents your winnings; if it is negative, you have lost money. Game B uses two biased coins, coin b and coin c. If your winnings in dollars are a multiple of 3, then you flip coin b, which comes up heads with probability p_b and tails with probability $1 - p_b$. Otherwise, you flip coin c, which comes up heads with probability p_c and tails with probability $1 - p_c$. Again, you win a dollar if the coin comes up heads and lose a dollar if it comes up tails.

This game is more complicated, so let us consider a specific example. Suppose coin b comes up heads with probability $p_b = 0.09$ and tails with probability 0.91 and that coin c comes up heads with probability $p_c = 0.74$ and tails with probability 0.26. At first glance, it might seem that game B is in your favor. If we use coin b for the 1/3 of the time that your winnings are a multiple of 3 and use coin c the other 2/3 of the time, then your probability w of winning is

$$w = \frac{1}{3}\frac{9}{100} + \frac{2}{3}\frac{74}{100} = \frac{157}{300} > \frac{1}{2}.$$

The problem with this line of reasoning is that coin b is not necessarily used 1/3 of the time! To see this intuitively, consider what happens when you first start the game, when your winnings are 0. You use coin b and most likely lose, after which you use coin c and most likely win. You may spend a great deal of time going back and forth between having lost one dollar and breaking even before either winning one dollar or losing two dollars, so you may use coin b more than 1/3 of the time.

In fact, the specific example for game B is a losing game for you. One way to show this is to suppose that we start playing game B when your winnings are 0, continuing until you either lose three dollars or win three dollars. If you are more likely to lose than win in this case, by symmetry you are more likely to lose three dollars than win three dollars whenever your winnings are a multiple of 3. On average, then, you would obviously lose money on the game.

One way to determine if you are more likely to lose than win is to analyze the absorbing states. Consider the Markov chain on the state space consisting of the integers $\{-3, \dots, 3\}$, where the states represent your winnings. We want to know, when you start at 0, whether or not you are more likely to reach -3 before reaching 3. We can determine this by setting up a system of equations. Let z_i represent the probability you will end up having lost three dollars before having won three dollars when your current winnings are i dollars. We calculate all the probabilities $z_{-3}, z_{-2}, z_{-1}, z_0, z_1, z_2$, and z_3, although what we are really interested in is z_0. If $z_0 > 1/2$, then we are more likely to lose three dollars than win three dollars starting from 0. Here $z_{-3} = 1$ and $z_3 = 0$; these are boundary conditions. We also have the following equations:

$$\begin{aligned}
z_{-2} &= (1 - p_c)z_{-3} + p_c z_{-1},\\
z_{-1} &= (1 - p_c)z_{-2} + p_c z_0,\\
z_0 &= (1 - p_b)z_{-1} + p_b z_1,\\
z_1 &= (1 - p_c)z_0 + p_c z_2,\\
z_2 &= (1 - p_c)z_1 + p_c z_3.
\end{aligned}$$

This is a system of five equations with five unknowns, and hence it can be solved easily. The general solution for z_0 is

$$z_0 = \frac{(1-p_b)(1-p_c)^2}{(1-p_b)(1-p_c)^2 + p_b p_c^2}.$$

For the specific example here, the solution yields $z_0 = 15{,}379/27{,}700 \approx 0.555$, showing that one is much more likely to lose than win playing this game over the long run.

Instead of solving these equations directly, there is a simpler way of determining the relative probability of reaching -3 or 3 first. Consider any sequence of moves that starts at 0 and ends at 3 before reaching -3. For example, a possible sequence is

$$s = 0, 1, 2, 1, 2, 1, 0, -1, -2, -1, 0, 1, 2, 1, 2, 3.$$

We create a one-to-one and onto mapping of such sequences with the sequences that start at 0 and end at -3 before reaching 3 by negating every number starting from the last 0 in the sequence. In this example, s maps to $f(s)$, where

$$f(s) = 0, 1, 2, 1, 2, 1, 0, -1, -2, -1, 0, -1, -2, -1, -2, -3.$$

It is simple to check that this is a one-to-one mapping of the relevant sequences.

The following lemma provides a useful relationship between s and $f(s)$.

Lemma 7.18: *For any sequence s of moves that starts at 0 and ends at 3 before reaching -3, we have*

$$\frac{\Pr(s \text{ occurs})}{\Pr(f(s) \text{ occurs})} = \frac{p_b p_c^2}{(1-p_b)(1-p_c)^2}.$$

Proof: For any given sequence s satisfying the properties of the lemma, let t_1 be the number of transitions from 0 to 1; t_2, the number of transitions from 0 to -1; t_3, the sum of the number of transitions from -2 to -1, -1 to 0, 1 to 2, and 2 to 3; and t_4, the sum of the number of transitions from 2 to 1, 1 to 0, -1 to -2, and -2 to -3. Then the probability that the sequence s occurs is $p_b^{t_1}(1-p_b)^{t_2} p_c^{t_3}(1-p_c)^{t_4}$.

Now consider what happens when we transform s to $f(s)$. We change one transition from 0 to 1 into a transition from 0 to -1. After this point, in s the total number of transitions that move up 1 is two more than the number of transitions that move down 1, since the sequence ends at 3. In $f(s)$, then, the total number of transitions that move down 1 is two more than the number of transitions that move up 1. It follows that the probability that the sequence $f(s)$ occurs is $p_b^{t_1-1}(1-p_b)^{t_2+1} p_c^{t_3-2}(1-p_c)^{t_4+2}$. The lemma follows. ∎

By letting S be the set of all sequences of moves that start at 0 and end at 3 before reaching -3, it immediately follows that

$$\frac{\Pr(3 \text{ is reached before } -3)}{\Pr(-3 \text{ is reached before } 3)} = \frac{\sum_{s \in S} \Pr(s \text{ occurs})}{\sum_{s \in S} \Pr(f(s) \text{ occurs})} = \frac{p_b p_c^2}{(1-p_b)(1-p_c)^2}.$$

If this ratio is less than 1, then you are more likely to lose than win. In our specific example, this ratio is $12{,}321/15{,}379 < 1$.

Finally, yet another way to analyze the problem is to use the stationary distribution. Consider the Markov chain on the states $\{0, 1, 2\}$, where here the states represent the remainder when our winnings are divided by 3. (That is, the state keeps track of $w - \ell \bmod 3$.) Let π_i be the stationary probability of this chain. The probability that we win a dollar in the stationary distribution, which is the limiting probability that we win a dollar if we play long enough, is then

$$\begin{aligned} p_b\pi_0 + p_c\pi_1 + p_c\pi_2 &= p_b\pi_0 + p_c(1 - \pi_0) \\ &= p_c - (p_c - p_b)\pi_0. \end{aligned}$$

Again, we want to know if this is greater than or less than 1/2.

The equations for the stationary distribution are easy to write:

$$\begin{aligned} \pi_0 + \pi_1 + \pi_2 &= 1, \\ p_b\pi_0 + (1 - p_c)\pi_2 &= \pi_1, \\ p_c\pi_1 + (1 - p_b)\pi_0 &= \pi_2, \\ p_c\pi_2 + (1 - p_c)\pi_1 &= \pi_0. \end{aligned}$$

Indeed, since there are four equations and only three unknowns, one of these equations is actually redundant. The system is easily solved to find

$$\begin{aligned} \pi_0 &= \frac{1 - p_c + p_c^2}{3 - 2p_c - p_b + 2p_bp_c + p_c^2}, \\ \pi_1 &= \frac{p_bp_c - p_c + 1}{3 - 2p_c - p_b + 2p_bp_c + p_c^2}, \\ \pi_2 &= \frac{p_bp_c - p_b + 1}{3 - 2p_c - p_b + 2p_bp_c + p_c^2}. \end{aligned}$$

Recall that you lose if the probability of winning in the stationary distribution is less than 1/2 or, equivalently, if $p_c - (p_c - p_b)\pi_0 < 1/2$. In our specific example, $\pi_0 = 673/1759 \approx 0.3826\ldots$, and

$$p_c - (p_c - p_b)\pi_0 = \frac{86{,}421}{175{,}900} < \frac{1}{2}.$$

Again, we find that game B is a losing game in the long run.

We have now completely analyzed game A and game B. Next let us consider what happens when we try to combine these two games. In game C, we repeatedly perform the following bet. We start by flipping a fair coin, call it coin d. If coin d is heads, we proceed as in game A: we flip coin a, and if the coin is heads, you win. If coin d is tails, we then proceed to game B: if your current winnings are a multiple of 3, we flip coin b; otherwise, we flip coin c, and if the coin is heads then you win. It would seem that this must be a losing game for you. After all, game A and game B are both losing games, and this game just flips a coin to decide which of the two games to play.

In fact, game C is exactly like game B, except the probabilities are slightly different. If your winnings are a multiple of 3, then the probability that you win is $p_b^* = \frac{1}{2}p_a + \frac{1}{2}p_b$. Otherwise, the probability that you win is $p_c^* = \frac{1}{2}p_a + \frac{1}{2}p_c$. Using p_b^* and p_c^* in place of p_b and p_c, we can repeat any of the foregoing analyses we used for game B.

For example: if the ratio

$$\frac{p_b^*(p_c^*)^2}{(1-p_b^*)(1-p_c^*)^2} < 1,$$

then the game is a losing game for you; if the ratio is larger than 1, it is a winning game. In our specific example the ratio is $438{,}741/420{,}959 > 1$, so game C appears to be a winning game.

This seems somewhat odd, so let us recheck by using our other approach of considering the stationary distribution. The game is a losing game if $p_c^* - (p_c^* - p_b^*)\pi_0 < 1/2$ and a winning game if $p_c^* - (p_c^* - p_b^*)\pi_0 > 1/2$, where π_0 is now the stationary distribution for the chain corresponding to game C. In our specific example, $\pi_0 = 30{,}529/88{,}597$, and

$$p_c^* - (p_c^* - p_b^*)\pi_0 = \frac{4{,}456{,}523}{8{,}859{,}700} > \frac{1}{2},$$

so game C again appears to be a winning game.

How can randomly combining two losing games yield a winning game? The key is that game B was a losing game because it had a very specific structure. You were likely to lose the next round in game B if your winnings were divisible by 3, but if you managed to get over that initial barrier you were likely to win the next few games as well. The strength of that barrier made game B a losing game. By combining the games that barrier was weakened, because now when your winnings are divisible by 3 you sometimes get to play game A, which is close to a fair game. Although game A is biased against you, the bias is small, so it becomes easier to overcome that initial barrier. The combined game no longer has the specific structure required to make it a losing game.

You may be concerned that this seems to violate the law of linearity of expectations. If the winnings from a round of game A, B, and C are X_A, X_B, and X_C (respectively), then it seems that

$$\mathbf{E}[X_C] = \mathbf{E}\left[\frac{1}{2}X_A + \frac{1}{2}X_B\right] = \frac{1}{2}\mathbf{E}[X_A] + \frac{1}{2}\mathbf{E}[X_B],$$

so if $\mathbf{E}[X_A]$ and $\mathbf{E}[X_B]$ are negative then $\mathbf{E}[X_C]$ should also be negative. The problem is that this equation does not make sense, because we cannot talk about the expected winnings of a round of games B and C without reference to the current winnings. We have described a Markov chain on the states $\{0, 1, 2\}$ for games B and C. Let s represent the current state. We have

$$\begin{aligned}\mathbf{E}[X_C \mid s] &= \mathbf{E}\left[\frac{1}{2}(X_A + X_B) \mid s\right] \\ &= \frac{1}{2}\mathbf{E}[X_A \mid s] + \frac{1}{2}\mathbf{E}[X_B \mid s].\end{aligned}$$

Linearity of expectations holds for any given step, but we must condition on the current state. By combining the games we have changed how often the chain spends in each state, allowing the two losing games to become a winning game.

7.6. Exercises

Exercise 7.1: Consider a Markov chain with state space $\{0, 1, 2, 3\}$ and a transition matrix

$$\mathbf{P} = \begin{bmatrix} 0 & 3/10 & 1/10 & 3/5 \\ 1/10 & 1/10 & 7/10 & 1/10 \\ 1/10 & 7/10 & 1/10 & 1/10 \\ 9/10 & 1/10 & 0 & 0 \end{bmatrix},$$

so $P_{0,3} = 3/5$ is the probability of moving from state 0 to state 3.

(a) Find the stationary distribution of the Markov chain.
(b) Find the probability of being in state 3 after 32 steps if the chain begins at state 0.
(c) Find the probability of being in state 3 after 128 steps if the chain begins at a state chosen uniformly at random from the four states.
(d) Suppose that the chain begins in state 0. What is the smallest value of t for which $\max_s |P^t_{0,s} - \pi_s| \leq 0.01$? Here $\bar{\pi}$ is the stationary distribution. What is the smallest value of t for which $\max_s |P^t_{0,s} - \pi_s| \leq 0.001$?

Exercise 7.2: Consider the two-state Markov chain with the following transition matrix

$$\mathbf{P} = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix}.$$

Find a simple expression for $P^t_{0,0}$.

Exercise 7.3: Consider a process $X_0, X_1, X_2, \ldots$ with two states, 0 and 1. The process is governed by two matrices, $\mathbf{P}$ and $\mathbf{Q}$. If k is even, the values $P_{i,j}$ give the probability of going from state i to state j on the step from X_k to X_{k+1}. Likewise, if k is odd then the values $Q_{i,j}$ give the probability of going from state i to state j on the step from X_k to X_{k+1}. Explain why this process does not satisfy Definition 7.1 of a (time-homogeneous) Markov chain. Then give a process with a larger state space that is equivalent to this process and satisfies Definition 7.1.

Exercise 7.4: Prove that the communicating relation defines an equivalence relation.

Exercise 7.5: Prove that if one state in a communicating class is transient (respectively, recurrent) then all states in that class are transient (respectively, recurrent).

Exercise 7.6: In studying the 2-SAT algorithm, we considered a 1-dimensional random walk with a completely reflecting boundary at 0. That is, whenever position 0 is reached, with probability 1 the walk moves to position 1 at the next step. Consider now a random walk with a partially reflecting boundary at 0. Whenever position 0 is reached, with probability $1/2$ the walk moves to position 1 and with probability $1/2$ the walk stays at 0. Everywhere else the random walk moves either up or down 1, each with probability $1/2$. Find the expected number of moves to reach n, starting from position i and using a random walk with a partially reflecting boundary.

Exercise 7.7: Suppose that the 2-SAT Algorithm 7.1 starts with an assignment chosen uniformly at random. How does this affect the expected time until a satisfying assignment is found?

Exercise 7.8: Generalize the randomized algorithm for 3-SAT to k-SAT. What is the expected time of the algorithm as a function of k?

Exercise 7.9: In the analysis of the randomized algorithm for 3-SAT, we made the pessimistic assumption that the current assignment A_i and the truth assignment S differ on just one variable in the clause chosen at each step. Suppose instead that, independently at each step, the two assignments disagree on one variable in the clause with probability p and at least two variables with probability $1 - p$. What is the largest value of p for which you can prove that the expected number of steps before Algorithm 7.2 terminates is polynomial in the number of variables n? Give a proof for this value of p and give an upper bound on the expected number of steps in this case.

Exercise 7.10: A *coloring* of a graph is an assignment of a color to each of its vertices. A graph is k-colorable if there is a coloring of the graph with k colors such that no two adjacent vertices have the same color. Let G be a 3-colorable graph.

(a) Show that there exists a coloring of the graph with two colors such that no triangle is monochromatic. (A triangle of a graph G is a subgraph of G with three vertices, which are all adjacent to each other.)

(b) Consider the following algorithm for coloring the vertices of G with two colors so that no triangle is monochromatic. The algorithm begins with an arbitrary 2-coloring of G. While there are any monochromatic triangles in G, the algorithm chooses one such triangle and changes the color of a randomly chosen vertex of that triangle. Derive an upper bound on the expected number of such recoloring steps before the algorithm finds a 2-coloring with the desired property.

Exercise 7.11: An $n \times n$ matrix $\mathbf{P}$ with entries $P_{i,j}$ is called stochastic if all entries are nonnegative and if the sum of the entries in each row is 1. It is called doubly stochastic if, additionally, the sum of the entries in each column is 1. Show that the uniform distribution is a stationary distribution for any Markov chain represented by a doubly stochastic matrix.

Exercise 7.12: Let X_n be the sum of n independent rolls of a fair die. Show that, for any $k \geq 2$,

$$\lim_{n\to\infty} \Pr(X_n \text{ is divisible by } k) = \frac{1}{k}.$$

Exercise 7.13: Consider a finite Markov chain on n states with stationary distribution $\bar{\pi}$ and transition probabilities $P_{i,j}$. Imagine starting the chain at time 0 and running it for m steps, obtaining the sequence of states $X_0, X_1, \ldots, X_m$. Consider the states in reverse order, $X_m, X_{m-1}, \ldots, X_0$.

(a) Argue that given X_{k+1}, the state X_k is independent of $X_{k+2}, X_{k+3}, \ldots, X_m$. Thus the reverse sequence is Markovian.

(b) Argue that for the reverse sequence, the transition probabilities $Q_{i,j}$ are given by

$$Q_{i,j} = \frac{\pi_j P_{j,i}}{\pi_i}.$$

(c) Prove that if the original Markov chain is time reversible, so that $\pi_i P_{i,j} = \pi_j P_{j,i}$, then $Q_{i,j} = P_{i,j}$. That is, the states follow the same transition probabilities whether viewed in forward order or reverse order.

Exercise 7.14: Prove that the Markov chain corresponding to a random walk on an undirected, non-bipartite graph that consists of one component is time reversible.

Exercise 7.15: Let $P_{i,i}^t$ be the probability that a Markov chain returns to state i when started in state i after t steps. Prove that

$$\sum_{t=1}^{\infty} P_{i,i}^t$$

is unbounded if and only if state i is recurrent.

Exercise 7.16: Prove Lemma 7.5.

Exercise 7.17: Consider the following Markov chain, which is similar to the 1-dimensional random walk with a completely reflecting boundary at 0. Whenever position 0 is reached, with probability 1 the walk moves to position 1 at the next step. Otherwise, the walk moves from i to $i+1$ with probability p and from i to $i-1$ with probability $1-p$. Prove that:

(a) if $p < 1/2$, each state is positive recurrent;
(b) if $p = 1/2$, each state is null recurrent;
(c) if $p > 1/2$, each state is transient.

Exercise 7.18: **(a)** Consider a random walk on the 2-dimensional integer lattice, where each point has four neighbors (up, down, left, and right). Is each state transient, null recurrent, or positive recurrent? Give an argument.

(b) Answer the problem in (a) for the 3-dimensional integer lattice.

Exercise 7.19: Consider the gambler's ruin problem, where a player plays until she lose ℓ_1 dollars or win ℓ_2 dollars. Prove that the expected number of games played is $\ell_1 \ell_2$.

Exercise 7.20: We have considered the gambler's ruin problem in the case where the game is fair. Consider the case where the game is not fair; instead, the probability of losing a dollar each game is $2/3$ and the probability of winning a dollar each game is $1/3$. Suppose that you start with i dollars and finish either when you reach n or lose it all. Let W_t be the amount you have gained after t rounds of play.

(a) Show that $\mathbf{E}[2^{W_{t+1}}] = \mathbf{E}[2^{W_t}]$.
(b) Use part (a) to determine the probability of finishing with 0 dollars and the probability of finishing with n dollars when starting at position i.
(c) Generalize the preceding argument to the case where the probability of losing is $p > 1/2$. (*Hint:* Try considering $\mathbf{E}[c^{W_t}]$ for some constant c.)

Exercise 7.21: Consider a Markov chain on the states $\{0, 1, \ldots, n\}$, where for $i < n$ we have $P_{i,i+1} = 1/2$ and $P_{i,0} = 1/2$. Also, $P_{n,n} = 1/2$ and $P_{n,0} = 1/2$. This process can be viewed as a random walk on a directed graph with vertices $\{0, 1, \ldots, n\}$, where each vertex has two directed edges: one that returns to 0 and one that moves to the vertex with the next higher number (with a self-loop at vertex n). Find the stationary distribution of this chain. (This example shows that random walks on directed graphs are very different than random walks on undirected graphs.)

Exercise 7.22: A cat and a mouse each independently take a random walk on a connected, undirected, non-bipartite graph G. They start at the same time on different nodes, and each makes one transition at each time step. The cat eats the mouse if they are ever at the same node at some time step. Let n and m denote, respectively, the number of vertices and edges of G. Show an upper bound of $O(m^2 n)$ on the expected time before the cat eats the mouse. (*Hint:* Consider a Markov chain whose states are the ordered pairs (a, b), where a is the position of the cat and b is the position of the mouse.)

Exercise 7.23: One way of spreading information on a network uses a rumor-spreading paradigm. Suppose that there are n hosts currently on the network. Initially, one host begins with a message. Each round, every host that has the message contacts another host chosen independently and uniformly at random from the other $n - 1$ hosts, and sends that host the message. We would like to know how many rounds are necessary before all hosts have received the message with probability 0.9999.

(a) Explain how this problem can be viewed in terms of Markov chains.
(b) Determine a method for computing the probability that j hosts have received the message after round k given that i hosts have received the message after round $k - 1$. (*Hint:* There are various ways of doing this. One approach is to let $P(i, j, c)$ be the probability that j hosts have the message after the first c of the i hosts have made their choices in a round; then find a recurrence for P.)
(c) As a computational exercise, write a program to determine the number of rounds required for a message starting at one host to reach all other hosts with probability 0.9999 when $n = 128$.

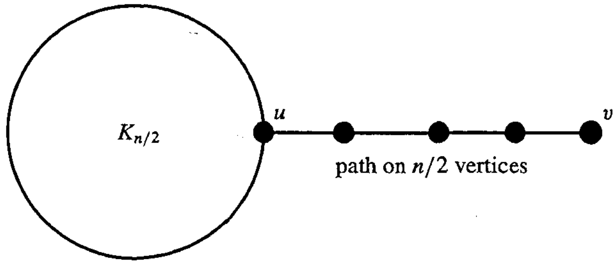


Figure 7.3: Lollipop graph.

Exercise 7.24: The *lollipop* graph on n vertices is a clique on $n/2$ vertices connected to a path on $n/2$ vertices, as shown in Figure 7.3. The node u is a part of both the clique and the path. Let v denote the other end of the path.

(a) Show that the expected covering time of a random walk starting at v is $\Theta(n^2)$.
(b) Show that the expected covering time for a random walk starting at u is $\Theta(n^3)$.

Exercise 7.25: The following is a variation of a simple children's board game. A player starts at position 0. On a player's turn, she rolls a standard six-sided die. If her old position was the positive integer x and her roll is y, then her new position is $x + y$, except in two cases:

- if $x + y$ is divisible by 6 and less than 36, her new position is $x + y - 6$;
- if $x + y$ is greater than 36, the player remains at x.

The game ends when a player reaches the goal position, 36.

(a) Let X_i be a random variable representing the number of rolls needed to get to 36 from position i for $0 \leq i \leq 35$. Give a set of equations that characterize $\mathbf{E}[X_i]$.
(b) Using a program that can solve systems of linear equations, find $\mathbf{E}[X_i]$ for $0 \leq i \leq 35$.

Exercise 7.26: Let n equidistant points be marked on a circle. Without loss of generality, we think of the points as being labeled clockwise from 0 to $n - 1$. Initially, a wolf begins at 0 and there is one sheep at each of the remaining $n - 1$ points. The wolf takes a random walk on the circle. For each step, it moves with probability $1/2$ to one neighboring point and with probability $1/2$ to the other neighboring point. At the first visit to a point, the wolf eats a sheep if there is still one there. Which sheep is most likely to be the last eaten?

Exercise 7.27: Suppose that we are given n records, $R_1, R_2, \ldots, R_n$. The records are kept in some order. The cost of accessing the jth record in the order is j. Thus, if we had four records ordered as R_2, R_4, R_3, R_1, then the cost of accessing R_4 would be 2 and the cost of accessing R_1 would be 4.

Suppose further that, at each step, record R_j is accessed with probability p_j, with each step being independent of other steps. If we knew the values of the p_j in advance, we would keep the R_j in decreasing order with respect to p_j. But if we don't know the p_j in advance, we might use the "move to front" heuristic: at each step, put the record that was accessed at the front of the list. We assume that moving the record can be done with no cost and that all other records remain in the same order. For example, if the order was R_2, R_4, R_3, R_1 before R_3 was accessed, then the order at the next step would be R_3, R_2, R_4, R_1.

In this setting, the order of the records can be thought of as the state of a Markov chain. Give the stationary distribution of this chain. Also, let X_k be the cost for accessing the kth requested record. Determine an expression for $\lim_{k\to\infty} \mathbf{E}[X_k]$. Your expression should be easily computable in time that is polynomial in n, given the p_j.

Exercise 7.28: Consider the following variation of the discrete time queue. Time is divided into fixed-length steps. At the beginning of each time step, a customer arrives with probability λ. At the end of each time step, if the queue is nonempty then the customer at the front of the line completes service with probability μ.

(a) Explain how the number of customers in the queue at the beginning of each time step forms a Markov chain, and determine the corresponding transition probabilities.
(b) Explain under what conditions you would expect a stationary distribution $\bar{\pi}$ to exist.
(c) If a stationary distribution exists, then what should be the value of π_0, the probability that no customers are in the queue at the beginning of the time step? (*Hint:* Consider that, in the long run, the rate at which customers enter the queue and the rate at which customers leave the queue must be equal.)
(d) Determine the stationary distribution and explain how it corresponds to your conditions from part (b).
(e) Now consider the variation where we change the order of incoming arrivals and service. That is: at the *beginning* of each time step, if the queue is nonempty then a customer is served with probability μ; and at the *end* of a time step a customer arrives with probability λ. How does this change your answers to parts (a)–(d)?

Exercise 7.29: Prove that the Markov chain from Lemma 7.14 where the states are the $2|E|$ directed edges of the graph has a uniform stationary distribution.

Exercise 7.30: We consider the covering time for the standard random walk on a hypercube with $N = 2^n$ nodes. (See Definition 4.3 if needed to recall the definition of a hypercube.) Let (u, v) be an edge in the hypercube.

(a) Prove that the expected time between traversals of the edge (u, v) from u to v is Nn.
(b) We consider the time between transitions from u to v in a different way. After moving from u to v, the walk must first return to u. When it returns to u, the walk might next move to v, or it might move to another neighbor of u, in which case it must return to u again before moving to v for there to be a transition from u to v.

Use symmetry and the above description to prove the following recurrence:

$$Nn = \sum_{i=1}^{\infty} \frac{1}{n} \left(\frac{n-1}{n} \right)^{i-1} (i(h_{u,v} + 1)) = n(h_{u,v} + 1).$$

(c) Conclude from the above that $h_{u,v} = N - 1$.

(d) Using the result on the hitting time of adjacent vertices and Matthews' theorem, show that the cover time is $O(N \log^2 N)$.

(e) As a much more challenging problem, you can try to prove that the maximum hitting time between any two vertices for the random walk on the hypercube is $O(N)$, and that the cover time is correspondingly $O(N \log N)$.