Markov Chains: Examples with Theory

Steven R. Dunbar

Email address: sdunbarne@gmail.com

Contents

List of Figures

List of Tables

Preface

Welcome to *Markov Chains: Examples with Theory.* Before launching into the content, take some time to learn my purpose in writing it, the audience I intended, and why a math book contains so many computer programs.

One day when I mentioned that I was working on this book, my wife asked me "Who are you writing this for?". Without thinking too hard, I quickly answered "Myself". My interests over the last 10 years have ranged over the mathematics underneath bioinformatics and political redistricting. In mathematically understanding both of these widely separated topics, large Markov chains are an essential tool. Although I had the usual basic knowledge about small Markov chains, I did not think I really understood large Markov chains. So I began studying large chains and eventually realized that my notes had a logical order as a book.

A quote from the well-known mathematician and author Paul Halmos captures the spirit of this book:

> A good stack of examples, as large as possible, is indispensable
> for a thorough understanding of any concept, and when I want to
> learn something new, I make it my first job to build one.

The stack of examples here are explicit Markov chains, mostly from familiar contexts, progressing from small state spaces to large state spaces. Many books on Markov chains present examples with small state spaces for the admirable reason that readers can grasp the ideas and compute the required quantities by hand. With modern computing power and with improved algorithms it is now possible to directly compute required quantities in chains with hundreds, thousands, or even millions of states. These examples are scattered throughout the literature and on the internet. Part of the goal is to present this "stack of examples" with unified notation and common vocabulary to make understanding the subject as a whole easier.

Each chapter has the same components. Each chapter starts with an introductory paragraph or two followed by a difficulty rating. In teaching and reading, I

have found it useful to give an idea of what is ahead. So I assign one of four ratings to each chapter:

- Everyone: contains no mathematics, suitable for general reading.
- Student: contains scenes of mild algebra or calculus that may require guidance.
- Mathematically Mature: may contain mathematics beyond calculus with proofs. This generally means mathematics up to the fourth-year level covered in an American university.
- Mathematicians Only: prolonged scenes of intense rigor. This generally means mathematics covered in graduate level courses.

Immediately after the chapter introduction is a Starter Question. This is for the reader to warm up and think critically about the topic. The main part of each chapter follows in several sections, written in a mildly formal, but not terse, Definition-Theorem-Proof-Remark style. This main part concludes with my answer to the Starter Question. Since the Starter Questions are open-ended, the answers are not definitive, and the reader may have valid answers which are shorter, longer or even different. The Chapter Ending Answers usually involve results from the main body of the Chapter, so the reader is not necessarily going to be able to answer. Remember, the Starter Questions are just intended to get the reader thinking critically about the chapter topic. Each chapter finishes with a summary of the definitions, vocabulary, and the notations. Illustrating a main idea from the chapter is an Algorithm and one or more programs, currently in the R programming language. The Algorithm gives the idea for the program in pseudocode. Finally there are a few exercises for the reader, some of which fill in details from the chapter.

Although the book leads to large Markov chains, two small examples appear wherever they are useful. The first is the almost trivial "weather in Oz" example which appears in many treatments of Markov chains. This small but useful example introduces concepts understandable almost at a glance. The second example appearing throughout is the Ehrenfest urn model. This canonical example is non-trivial but still can be understood with hand calculations.

Some features of the text are a little unusual. The first is that longer proofs are broken into short numbered steps. I find that this helps the reader follow along step by step. If the reader gets lost, it is easier to refer to a step by number rather than "the next to last sentence in the first paragraph." Enumerating the steps also makes the proofs more detailed with fewer gaps. The second is that except for this preface there are no first person pronouns. I try to present the subject just as it is without injecting personalities. I hope the reader finds this factual and helpful, not impersonal. The third feature is an emphasis on calculating variance and covariance of the random variables when possible, not just the expected value.

This text is more explicit about mathematical details, generally not skipping over steps. This makes the text longer than it would be in a printed edition. The omission of details may be an advantage with a paper book, saving cost and weight. High-quality electronic presentations in book format eliminate the need for space-saving omissions.

It is now fashionable to make short videos explaining all sorts of topics. I believe the short video format is not suitable for deep mathematical ideas and examples. An

extended long format text with a table of contents, an index, numbered theorems, illustrations, tables, and lengthy explanations allow the reader the chance to stop, think, re-read if necessary. I believe a text such as this is the best form for the topic of Markov chains with examples and theory.

Intended Audience and Background

This book is primarily for beginning graduate students in mathematics and mathematicians who want to learn about Markov chains in detail. Learners in physical sciences, computer science, economics, and engineering will also benefit from the book with its examples of Markov chains.

This book is for students after courses on calculus-based probability theory and linear algebra. To understand the explanations, theorems, proofs and to complete the exercises:

(1) The reader should understand probability theory including distributions, random variables, and elementary random processes.

(2) The reader should understand linear algebra and matrix theory, including eigenvalues and eigenvectors.

(3) The reader should understand real analysis, including ϵ - δ style proofs of convergence. Understanding asymptotics is necessary in Part 2, including $O(n)$ and $o(n)$ style notation. Measure-theoretic probability appears in only a few places.

(4) The reader should be familiar with common statistical concepts of parameter point evaluations and confidence intervals and hypothesis testing.

Notational Conventions

(1) The sequence of random variables for a Markov chain is X_n, the state space for the random variables is $\mathcal{X}$ and the values in the state space are x_i, x_j and so on.

(2) The variables i, j and ℓ are generic variables representing states if the state space context is clear or unnecessary.

(3) The variables n and m are generic variables for discrete values, usually times or indices.

(4) If $\mathcal{X}$ is finite, its cardinality is $k = \text{card}\,(\mathcal{X})$.

(5) The variable ν is exclusively used as a dummy variable of summation whenever the index of summation has no immediate interpretation. Sometimes μ is also used as a dummy variable of summation.

(6) P is the transition probability matrix for a Markov chain. Q is a related, derived, or auxiliary transition probability matrix.

(7) The stationary distribution for a Markov chain is π. Math boldface for this distribution or vector distinguishes it from the mathematical constant and also from the symbol of a permutation from the symmetric group.

Program Scripts

An important feature of this book is the simulation scripts that accompany most sections. The simulation scripts illustrate the concepts and theorems of the section with numerical and graphical evidence. The scripts are part of the "Rule of 3" teaching philosophy of presenting mathematical ideas symbolically, numerically and graphically.

The programs are springboards for further experimentation, not finished apps for everyday use. The scripts are minimal in size, in scope of implementation and with minimal output. The scripts are not complete, stand-alone, polished applications, rather they are proof-of-concept starting points. The reader should run the scripts to illustrate the ideas and provide numerical examples of the results in the section. The scripts provide a seed for new scripts to increase the size and scope of the simulations. Increasing the size can often demonstrate convergence or increase confidence in the results of the section. Increasing the size can also demonstrate that although convergence is mathematically guaranteed, sometimes the rate of convergence is slow. The reader is also encouraged to change the output, to provide more information, to add different graphical representations, and to investigate rates of convergence.

The scripts are not specifically designed to be efficient, either in program language implementation or in mathematical algorithm. Efficiency is not ignored, but it is not the primary consideration in the construction of the scripts. Similarity of the program algorithm to the mathematical idea takes precedence over efficiency. One noteworthy aspect of both the similarity and efficiency is that all the languages use vectorization along with other notational simplifications such as recycling. Vectorized scripts look more like the mathematical expressions found in the text, making the code easier to understand. Vectorized code often runs much faster.

The scripts minimize using additional language packages and add-ons, instead relying on the base language as much as possible. Some packages, particularly for Markov chain simulations and for linear algebra, are too useful to ignore and so the scripts use these mature packages judiciously. Avoiding simplifying packages sometimes has the consequence of making the scripts longer, or more complicated, or less efficient. On the other hand, for the reader interested in immediately applying the scripts, it eliminates the need for searching for and installing packages, some of which might be specialized, obscure, even outdated. Also, since plotting varies so much by language, graphics format, operating system, even available hardware or resources, most scripts avoid plotting or make only the simplest plots. The reader is encouraged to try additional language packages to make the scripts easier to read, or more efficient. The reader is also encouraged to make plots of the calculated quantities, using available plotting software or language packages. Since many of the scripts rely on independent samples, those scripts can be made to run faster by using parallelization of the sampling. The reader is encouraged to experiment with parallelization in the code.

The scripts are not intended to be a tutorial on how to do mathematical programming in any of the languages. A description of the algorithm used in the scripts

is in each section. The description is usually in full sentences rather than the more formal symbolic representation found in computer science pseudo-code. Given the description and some basic awareness of programming ideas, the scripts provide multiple examples for study. The scripts provide a starting point for investigating, testing and comparing language features from the documentation or from other sources. The scripts use good programming style whenever possible, but clarity, simplicity and similarity to the mathematics are primary considerations.

Disclaimer. The final preparation of this text happened at the same time that generative large language models (sometimes called AI) became widely available. Although these models can summarize large volumes of information, they do so with the possibility of errors, even "hallucinations". Not a single word of this text was written with the help of a generative large language model. Any mistakes are entirely my own.

Part 1

Basic Examples and Theory

Markov Chains

This chapter is a survey of basic concepts and facts about discrete time Markov processes and Markov chains. Detailed treatments are in [**Dur09, Fel73, GS97, KT75, KST74, KS60, LPW09, Ros06, Sti05**], among other standard references.

Mathematically Mature: may contain mathematics beyond calculus with proofs.

1.1. Chapter Starter Question

Consider the magical Land of Oz where the weather follows a pattern. If it is raining today, then there is a 50% chance of raining again tomorrow, and a 25% chance of either having a nice day or a snowy day tomorrow. Similarly if it is snowing today, there is a 50% chance of having snow tomorrow, and a 25% chance of either having a nice day, or a rainy day tomorrow. Also the land of Oz never has two nice days in a row, with equal chance of rain or snow the day after a nice day. What is the chance it is nice in two days given it is rainy today?

1.2. Introduction

Begin by understanding where Markov chains fit in the larger landscape of probability theory.

Definition 1.1. A **stochastic process** is a random function on a domain. The domain can be discrete or continuous. The domain is often interpreted as time. Stochastic processes are in contrast to deterministic processes. An outcome of the process at some time must be specified in terms of the probability of it happening.
◇

Example 1.2. Consider the following simple example:

$$f(t) = 2t$$

$$g(t) = \begin{cases} 2t & \text{with probability } 0.98 \\ 3t & \text{with probability } 0.02. \end{cases}$$

For $f(2)$ the outcome will always be 4, it is deterministic. However for $g(2)$ the most likely outcome is 4 but this will fail to be true for about 2% of the trials. $\quad \triangledown$

Definition 1.3. The outcome of a **discrete time discrete state space stochastic process** is one of several discrete states at a sequence of discrete times. The set of states is $\mathcal{X} = \{x_1, x_2, \ldots, x_i, \ldots\}$. Sometimes the discrete state space $\mathcal{X}$ is infinite, but often the state space is finite. The sequence of discrete times is $\{t_0, t_1, \ldots, t_n, \ldots\}$. Often the precise times are not needed, so the sequence of events is indicated with $1, 2, \ldots, n, \ldots\}$. At time n, guided by some information, the process acts in a probabilistic manner and outputs some x_i. Thus along a specific sample, as time progresses an observed sequence of states from the set $\mathcal{X}$ is a **sample path**. $\quad \diamond$

Definition 1.4. A **discrete time Markov process** X_n is a stochastic process having a fixed probability that it will be in state x_j at time $n+1$ when the process is in state x_i at time n, with no dependence on the prior history. Symbolically,

$$\mathbb{P}\left[X_{t_{n+1}} \mid X_{t_n} = x_n, X_{t_{n-1}} = x_{n-1}, \ldots X_{t_1} = x_1, X_{t_0} = x_0\right] =$$
$$\mathbb{P}\left[X_{t_{n+1}} \mid X_{t_n} = x_n\right]$$

Denote the probability of making the transition from state x_i to x_j at time n as:

$$P_{ij}(n) = \mathbb{P}\left[x_{n+1} = x_j \mid X_n = x_i\right].$$

Note that already the exact times have been dropped and replaced with their sequence position. $\quad \diamond$

Example 1.5. As a contrast, consider a discrete time stochastic process modeling the daily closing price of a company's stock. At time n, there is a wealth of available information influencing the current price: the past prices of the stock, the current market conditions, even the weather forecast for the next few days. Without simplifying assumptions, stock prices would not be a Markov process.

Instead, simplify by modeling the daily closing price of a stock as a discrete time *Markov process*, at time $n+1$ the *only* information that would affect the process would be the observed stock price at time n. $\quad \triangledown$

Definition 1.6. If the probability of a state transition in a Markov process does not depend on n, then it is a **time-homogeneous Markov chain**. This means $P_{ij}(n) = P_{ij}(m)$ for all n, m. That is, the transition probabilities from x_i to x_j are time invariant with fixed P_{ij}, called the transition probabilities from x_i to x_j. If the state space is finite, say $\text{card}(\mathcal{X}) = k$, then the P_{ij} are the entries of an $k \times k$ matrix P called the transition probability matrix. $\quad \diamond$

1.3. Example of a Markov Chain

The following is a classic example from [**KST74**], repeated in [**GS97**] and many other locations. The example shows the concepts in context as well as introducing some other ideas.

Example 1.7. Consider the magical Land of Oz where the weather follows a pattern. If it is raining today, then there is a 50% chance of raining again tomorrow, and a 25% chance of either having a nice day or a snowy day tomorrow. Similarly if it is snowing, there is a 50% chance of again having snow, and a 25% chance of either having a nice day, or a rainy day. The land of Oz never has two nice days in a row, and Oz equally has rain and snow the day after a nice day.

First label the states conveniently as R, N, S, instead of x_1, x_2, x_3. With the information above, define $P_{RR} = 1/2$, $P_{RN} = 1/4$ and so on. Writing out every possible transition gives 3^2 distinct possibilities. With k states, the process has k^2 transitions with the probability P_{ij} of going from x_i to x_j.

Use matrices to represent the transition probabilities for the Markov chain. In the example, the transition matrix is (the row and column labels of R, N and S are for convenience only and are not usually included in the display of the matrix)

$$P = \begin{array}{c} \\ R \\ N \\ S \end{array} \begin{pmatrix} R & N & S \\ 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/2 \end{pmatrix}.$$

Note that all the rows sum to 1, thus this is a **stochastic matrix**.

How to calculate the chance it is nice in two days given it is rainy today? Use the Law of Total Probability, considering every possible weather combination with their respective conditional probabilities. See Figure 1 for a diagram of the combination process. Mixing the $R = x_1$, $N = x_2$, $S = x_3$ notation, this example has

$$\mathbb{P}\left[X_{n+2} = N | X_n = R\right] = \sum_{\nu=1}^{3} P_{1\nu} P_{\nu 2}.$$

For two days, this is not too tedious, but what about the probability it will be nice a week from today given it is rainy today?

Now matrix notation is convenient. Notice that the probability above of going from R to N two days hence is the $(1, 2)$ entry in the matrix product of P with itself. More generally, the transition probability through n steps is the matrix P^n, and this is known as (the finite Markov chain version of) the *Chapman-Kolmogorov Theorem*. Thus consider the following matrices from the example:

$$P^2 = \begin{array}{c} \\ R \\ N \\ S \end{array} \begin{pmatrix} R & N & S \\ 7/16 & 3/16 & 6/16 \\ 6/16 & 4/16 & 6/16 \\ 6/16 & 3/16 & 7/16 \end{pmatrix}$$
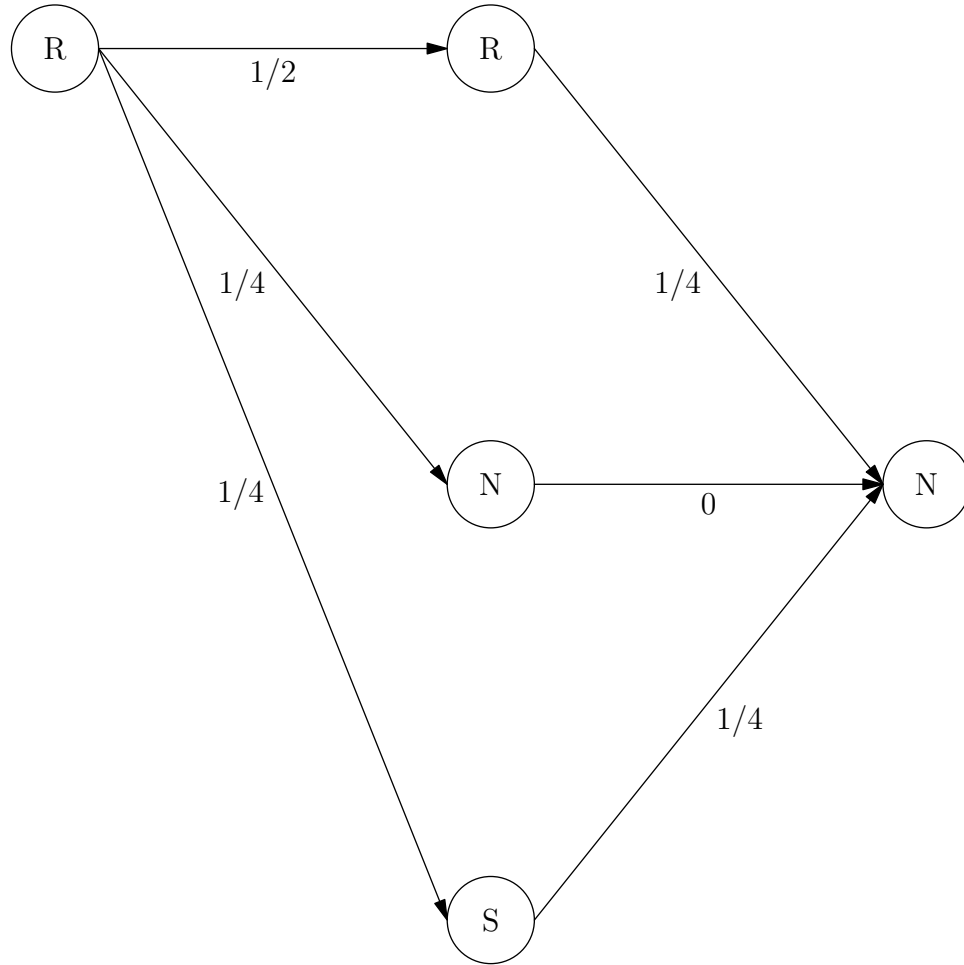
Figure 1. A schematic diagram of the calculation of the two-step transition probability.

and

$$
P^7 = \begin{matrix} R \\ N \\ S \end{matrix}
\begin{pmatrix}
R & N & S \\
0.400024 & 0.200012 & 0.399963 \\
0.400024 & 0.199951 & 0.400024 \\
0.399963 & 0.200012 & 0.400024
\end{pmatrix}.
$$

If it is raining today, then the chance it is nice in 2 and 7 days is $3/16$ and 0.200012 respectively. ∇

1.4. Formal Definitions

Let the state space be a finite set $\mathcal{X} = \{x_1, x_2, \ldots, x_k\}$.

Definition 1.8. A **Markov chain** is a discrete time discrete state space stochastic process defined by its transition probability matrix P with entries

$$P_{ij}(n) = \mathbb{P}\left[X_n = x_j \mid X_{n-1} = x_i\right].$$

The entries must satisfy $P_{ij} \geq 0$ and $\sum_{\nu=1}^{k} P_{i\nu} = 1$ for each $i = 1, \ldots, k$, so that each row is a probability distribution on $\mathcal{X}$. $\diamond$

This means that the Markov chain is a random walk among the states with dependence only on the current state. If x_i is the current state, choose x_j with probability P_{ij}; from x_j choose x_l with probability P_{jl}, and so on. The sequence of outcomes $X_0 = x_i, X_1 = x_j, X_2 = x_l \ldots$ is a **sample path** of the Markov chain.

In terms of conditional probabilities,

$$\mathbb{P}\left[X_1 = x_j \mid X_0 = x_i\right] = P_{ij}$$
$$\mathbb{P}\left[X_2 = x_l, X_1 = x_j \mid X_0 = x_i\right] = P_{ij}P_{jl}.$$

Then combining over all the intermediate possibilities

$$\mathbb{P}\left[X_2 = x_l \mid X_0 = x_i\right] = \sum_{j=1}^{k} P_{ij}P_{jl}$$

which is matrix multiplication. This is the *Chapman-Kolmogorov equation* for finite Markov chains. Extending this observation by induction

$$\mathbb{P}\left[X_n = j \mid X_0 = i\right] = (P^n)_{ij}$$

the i, j entry of the nth power of P.

A Markov chain has a **stationary distribution** if there is a vector π satisfying

$$\sum_{i=1}^{k} \pi_i P_{ij} = \pi_j$$

with $\pi_i > 0$, and $\sum_{i=1}^{k} \pi_i = 1$. Thus π is a left eigenvector of P with eigenvalue 1. The Perron Theorem guarantees that such a left eigenvector exists for transition probability matrices. Alternative terminology is that the Markov chain is **stable on the distribution** π and that π is the **stable distribution** for the Markov chain. The probabilistic interpretation of the left eigenvector equation is "pick x_i from $\mathcal{X}$ with probability π_i and take a step with probability P_{ij}, the probability of being at x_j is π_j". Thus π is stationary for the evolution of the Markov chain.

The Fundamental Theorem of Markov Chains guarantees that a stable distribution always exists under reasonable conditions.

Theorem 1.9 (Fundamental Theorem of Markov Chains)**.** *For an irreducible, positive recurrent and aperiodic Markov chain, $\lim_{n\to\infty}(P^n)_{ij}$ exists and is independent of i. Furthermore, letting*

$$\pi_j = \lim_{n\to\infty} (P^n)_{ij}$$

then π_j is the unique non-negative solution of

$$\sum_i \pi_i P_{ij} = \pi_j,$$

$$\sum_i \pi_i = 1.$$

The precise definitions of the conditions on a Markov chain for the following Fundamental Theorem to hold are in the next chapter "Classes of States and Stationary Distributions". The proof of the Fundamental Theorem is also in the next section "Classes of States and Stationary Distributions".

An irreducible, positive recurrent and aperiodic Markov chain with transition matrix P has the following property: As $n \to \infty$, the matrix P^n approaches a limiting matrix P^∞, where P^∞ consists of n copies of the same row vector. This matrix has different names in different texts, two common names are the "limit matrix", or the matrix of "limiting probabilities."

The probabilistic content of the Fundamental Theorem is that from any starting state x_i, for large n the nth step of a sample path of the Markov chain has a probability close to π_j of being at x_j. Later sections address the question of what "probability close to π_j" means and how large n must be for this probability relationship to hold.

Example 1.10. For the weather in Oz example, consider P^7 further:

$$P^7 = \begin{array}{c} \\ R \\ N \\ S \end{array} \begin{array}{c} R \\ \left(\begin{array}{ccc} 0.400024 & 0.200012 & 0.399963 \\ 0.400024 & 0.199951 & 0.400024 \\ 0.399963 & 0.200012 & 0.400024 \end{array} \right) \end{array}.$$

Note that the chance of being in any of the three states after seven days is almost independent of the initial state. This means the weather in seven days is almost completely independent of what it is today. Considering P^n for larger values of n, the row vectors approach $\pi = (2/5, 1/5, 2/5)$.   ▽

Example 1.11. Convergence to a stationary distribution works for the weather example, but it will not work for every matrix. Consider:

$$Q = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

This matrix describes a two-state system switching states at each time step. Thus $Q^2 = I$, $Q^3 = Q$, $Q^4 = I$ and so on. This particular matrix does not have the aperiodic property so it does not satisfy the requirements of the Fundamental Theorem.   ▽

1.5. Absorbing States

Up to this point the main example has been an irreducible, positive recurrent, and aperiodic Markov chain. Now consider the following transition matrix:

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Note that if the process starts in state x_1 or x_5, it can't leave these states. Similarly, if the process starts in any other state and "arrives" in states x_1 or x_5 at time N, then it stays in those states for all steps $n \geq N$. Call these **absorbing states**, that is, states which once entered are never left.

The transition matrix P represents the symmetric random walk on x_1, x_2, x_3, x_4, x_5 that stops at x_1 or x_5. Markov chains with absorbing states are *not* irreducible. Therefore the Fundamental Theorem does not apply for a limit matrix. Instead introduce the **transition probability matrix canonical form**. Given a Markov chain defined by transition matrix P with a absorbing states and t transient states where $a + t = k$, then after reordering the states the matrix has the canonical form:

$$P = \begin{pmatrix} I & 0 \\ A & T \end{pmatrix}$$

where I is the $a \times a$ identity matrix, and 0 is the $a \times t$ matrix of zeros. The random walk matrix from above has the canonical form:

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 1/2 & 0 & 1/2 & 0 \end{pmatrix}.$$

The canonical form of the transition probability matrix ordering the absorbing states first is common in theoretical treatments and in software packages implementing Markov chains. However in some contexts, especially games and sports, it is more convenient to order the start states as 1 and put the end absorbing states of winning or losing as the last of the states. The Markov chain then "flows" from the start state 1 of the game to some end state. Then the canonical form becomes

$$P = \begin{pmatrix} Q & R \\ 0 & I \end{pmatrix}.$$

The matrix $N = (I - T)^{-1}$ (or $N = (I - Q)^{-1}$) is called the **fundamental matrix**. The section on "Waiting Time to Absorption" proves that entry N_{ij} of the fundamental matrix is the expected number of times the chain will be at state x_j before absorption, given it started in state x_i. So in the random walk example:

$$N = (I - T)^{-1} = \begin{pmatrix} 3/2 & 1 & 1/2 \\ 1 & 2 & 1 \\ 1/2 & 1 & 3/2 \end{pmatrix}.$$

Letting $\mathbf{1}$ be the $1 \times t$ vector of all ones, $\mathbf{1}N$ is the vector with the expected number of steps for the transient state to be absorbed in one of the absorbing states, in this case $(3, 4, 3)$. Define $B = NA$, so B_{ij} is the probability that starting in the ith transient state, the process ends up in the jth absorbing state. In the example:

$$B = (I - T)^{-1}A = \begin{pmatrix} 3/4 & 1/4 \\ 1/2 & 1/2 \\ 1/4 & 3/4 \end{pmatrix}.$$

The canonical form gives information about P^n for large values of n. As $n \to \infty$, P^n approaches the matrix

$$P^\infty = \begin{pmatrix} I & 0 \\ B & 0 \end{pmatrix}.$$

1.6. Chapter Ending Answer

Use the Law of Total Probability to calculate this probability. This considers every possible weather combination with their respective conditional probabilities.

$$\mathbb{P}\left[X_2 = N \mid X_0 = R\right] = \frac{1}{2} \cdot \frac{1}{4} + \frac{1}{4} \cdot 0 + \frac{1}{4} \cdot \frac{1}{4} = \frac{3}{16}.$$

1.7. Chapter Summary

Key Concepts.

(1) A **stochastic process** isa random function on a domain. The domain can be discrete or continuous. The domain is often interpreted as time. Stochastic processes are in contrast to deterministic processes. A specific outcome of the process at some time must be specified in terms of the probability of it happening.

(2) A discrete time discrete state space stochastic process is a **Markov chain** if $\mathbb{P}\left[X_{n+1} \mid X_n, X_{n-1}, \ldots, X_1\right] = \mathbb{P}\left[X_{n+1} \mid X_n\right]$. That is, the conditional probability depends only on the immediately preceding state, not the history.

(3) Denote the probability of making the transition from state x_i to x_j at time n as:
$$P_{ij}(n) = \mathbb{P}\left[X_{n+1} = x_j \mid X_n = x_i\right].$$
If the probability of a state transition in a Markov process does not depend on n, then it is a **time-homogeneous Markov chain**. If the state space is finite, say $\text{card}\,(\mathcal{X}) = k$, then the p_{ij} are the entries of an $k \times k$ matrix P called the **transition probability matrix**.

(4) The transition probability for multiple time steps is the matrix P^n, and this is known as (the finite Markov chain version of) the *Chapman-Kolmogorov Theorem*.

(5) The Fundamental Theorem of Markov Chains says: Let $\mathcal{X}$ be a finite state space and let P_{ij} define a Markov chain on x_i. If there is an integer n_0 such that $(P^{n_0})_{ij} > 0$ for all $n > n_0$, then P has a unique stationary distribution π, and as $n \to \infty$, $(P^n)_{ij} \to \pi_j$ for each i, j.

(6) **Absorbing states** are states that once entered are never left.

(7) The limiting matrix can be found by putting the matrix in **canonical form**, then finding the **fundamental matrix**, and then multiplying blocks from the canonical matrix.

Vocabulary.

(1) A **stochastic process** is a random function on a domain.

(2) The outcome of a **discrete time discrete state space process** is one of several states at a sequence of times. Along a specific sample, as time progresses the observed sequence of states from the set $\mathcal{X}$ is a **sample path**.

(3) The outcome of a **discrete time discrete state space stochastic process** is one of several discrete states at a sequence of discrete times.

(4) A **discrete time Markov process** is a stochastic process having a fixed probability that it will be in state x_j at time $n+1$ when the process is in state x_i at time n, with no dependence on the prior history.

(5) If the probability of a state transition in a Markov process does not depend on n, it is a **time-homogeneous Markov chain**.

(6) If the values in each the rows of a matrix sum to 1, it is a **stochastic matrix**.

(7) A **Markov chain** is a discrete time discrete state space stochastic process defined by its transition probability matrix P with entries

$$P_{ij}(n) = \mathbb{P}\left[X_{n+1} = x_j \mid X_n = x_i\right].$$

The entries must satisfy $P_{ij} \geq 0$ and $\sum_{\nu=1}^{k} P_{i\nu} = 1$ for each $i = 1, \ldots, k$, so that each row is a probability distribution on X.

(8) The sequence of states $X_0 = x_i, X_1 = x_j, X_2 = x_l \ldots$ is a **sample path** of the Markov chain.

(9) A regular Markov chain has a **stationary distribution** $\pi_i > 0$, and $\sum_{i=1}^{k} \pi_i = 1$ with π satisfying

$$\sum_{i=1}^{k} \pi_i P_{ij} = \pi_j.$$

Thus π is a left eigenvector of P with eigenvalue 1. Alternative terminology is that the Markov chain is **stable on the distribution** π and that π is the **stable distribution** for the Markov chain.

(10) States which once entered are never left are **absorbing states**.

(11) Given a Markov chain defined by transition matrix P with a absorbing states and t non-absorbing states where $a+t = k$, then the matrix has the **transition probability matrix canonical form**

$$P = \begin{pmatrix} I & 0 \\ R & Q \end{pmatrix}$$

where I is the $u \times u$ identity matrix, and 0 is the $u \times v$ matrix of zeros. The matrix $N = (I - Q)^{-1}$ is called the **fundamental matrix**.

Notation.

(1) $\mathcal{X} = \{x_1, x_2, \ldots, x_i, \ldots, \}$ – state space of a discrete space stochastic process

(2) $\text{card}\,(\mathcal{X}) = k$ – cardinality of a finite state space

(3) P – $k \times k$ transition probability matrix

(4) ν – A dummy variable for summation

(5) i, j – arbitrary or generic state indices

(6) $(P^n)_{ij}$ – the i, j entry of the nth power of P.

(7) π – stationary distribution satisfying

$$\sum_{\nu=1}^{k} \pi_\nu P_{\nu j} = \pi_j$$

with $\pi_\nu > 0$, and $\sum_{\nu=1}^{k} \pi_\nu = 1$

(8)

$$P = \begin{pmatrix} I & 0 \\ A & T \end{pmatrix}$$

– canonical form for the transition matrix P with a absorbing states and t transient states where $a + t = k$, where I is the $a \times a$ identity matrix, and 0 is the $a \times t$ matrix of zeros.

1.8. Sources

This section is adapted from: Notes prepared by LT Grant and used with permission. Pieces of this section are adapted from: *Finite Markov Chains*, Kemeny and Snell, 1960, and from *Random Walks and Electric Networks*, Doyle and Snell, 1984, as well as *A First Course in Stochastic Processes* by S. Karlin and H. Taylor, Chapter 2, 1975.

1.9. Links

(1) *Introduction to Probability*, Grinstead and Snell. Chapter 11 - Markov Chains http://tinyurl.com/qw6sa Introduction to Probability, Chapter 11 Accessed Wed Jan 16 07:32:50 CST 2019

(2) Markov The markovchain package Accessed Wed Jan 16 07:34:15 CST 2019

(3) Getting Started with Markov Chains Accessed Wed Jan 16 07:32:22 CST 2019

1.10. Algorithms and Scripts

Data: State names and transition probability matrix
Result: Information about a simple Markov chain

1 *Initialization and sample paths*
2 Load Markov chain library
3 Set state names, set transition probability matrix, set start state
4 Set an example length and create a sample path of example length
5 Create a second sample path of example length

6 *Simulation of stationary distribution and comparison to theoretical*
7 Set a long path length, and a transient time
8 Create a long sample path
9 Slice the long sample path from the transient time to the end
10 In the slice count the appearance of each state
11 Store in an empirical array
12 Compute the theoretical stable array
13 **return** *Stable distribution and theoretical stable distribution*

Algorithm 1: Markov chain simulation.

Scripts.

R: R script for markovchain.

```r
library(markovchain)

stateNames <- c("R", "N", "S")
transMatrix <- matrix( c(1/2, 1/4, 1/4, 1/2, 0, 1/2, 1/4, 1/4, 1/2),
                       nrow=3, byrow=TRUE)
startState <- "N"

weatherOz <- new("markovchain", transitionMatrix=transMatrix,
                 states=stateNames, name="Weather_in_Oz")
print( weatherOz )

print( summary(weatherOz) )

pathLength <- 10
weatherPath <- rmarkovchain(n=pathLength,
                            object = weatherOz, t0 = startState)
cat("A sample starting from N: ", weatherPath, "\n")
weatherPath <- rmarkovchain(n=pathLength,
                            object = weatherOz, t0 = startState)
cat("Another sample starting from N: ", weatherPath, "\n")

largeN <- 1000; startStable <- 100
weatherPath <- rmarkovchain(n=largeN,
                            object = weatherOz, t0 = startState)
stablePattern <-  weatherPath[startStable:largeN]
empiricalStable <- c( mean( stablePattern == "R" ),
                      mean( stablePattern == "N" ),
                      mean( stablePattern == "S" ))
theorStable <- steadyStates(weatherOz)
cat("Empirical stable distribution: ",  empiricalStable, "\n")
cat("Theoretical stable distribution: ", theorStable, "\n")
```

.
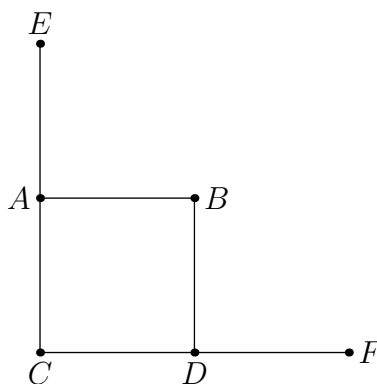
Figure 2. Grid for the random walk.

1.11. Problems to Work for Understanding

1: Students' progress through a (two-year) Associate Degree program is not always a two year process. A student taking first year courses has a 15% chance of dropping out, a 25% chance of repeating the first year, and a 60% chance of moving on to the second year. A student taking second year material has a 10% chance of dropping out, a 20% of repeating the second year, and a 70% chance of graduating.

(a) How many states are in this model?

(b) Which states are absorbing? Which are transient?

(c) What is the chance a student entering the program this year will graduate "on time"? What is the chance a student will graduate in 3 years?

(d) What is the chance that a student will graduate eventually? What is the chance the student will drop out?

(e) What is the expected time until a student graduates?

2: Consider the grid in Figure 2 upon which a random walker is stuck. If she reaches states E and F she will stop, otherwise she will keep walking. At any of the other states, she is equally likely to choose any of the possible neighboring states.

(a) What is the transition matrix?

(b) What are the absorbing states? The transient states?

(c) What is the canonical form of this matrix? The fundamental form?

(d) Starting at state A, on average, how many steps will the walker take before reaching states E or F?

(e) Starting at state A, what is the probability the walker will end at the state F? Starting at D what is the probability the walker ends at state E? Discuss these two answers.

3: The walker returns a year later to the same grid, only to find that there now exists a road between B and E.

(a) Without any calculations, how would you expect this to change the absorption probabilities of state B?

(b) Without any calculations, would you expect the same results as you found in part (e) above?

(c) Find the new transition matrix.

(d) Calculate the actual probabilities for part (a) and (b) and comment on your results.

4: Determine the transition probability matrix for the following Markov chain: N black balls and N white balls are placed in two urns so that each contains N balls. At each step one ball is selected at random from each urn and the two balls interchange urns. The state of the system is the number of white balls in the first urn. What are the absorbing states and transient states?

5: Determine the transition probability matrix for the following Markov chain: Consider two urns A and B with a total of N balls. An experiment is performed in which a ball is selected at random (all selections equally likely) at time n, $n = 1, 2, \ldots$ from among the totality of N balls. Then an urn is selected at random, urn A with probability p and urn B with probability $q = 1 - p$ and the ball previously drawn is placed in this urn. The state of the system is the number of balls in the urn A. What are the absorbing states and transient states, if any?

6: Determine the transition probability matrix for the following Markov chain: Consider two urns A and B with a total of N balls. An experiment is performed in which an urn is selected at random in proportion to its contents (i.e. if urn A has exactly i balls, then it is selected with probability i/N. The a ball is selected from A with probability p and from urn B with probability $q = 1 - p$ and placed in the previously chosen urn. The state of the system is the number of balls in the urn A. What are the absorbing states and transient states, if any?

7: Your baby is learning to walk. The baby begins by holding onto a couch. Whenever she is next to the couch, there is a 25% chance that she will take a step forward and a 75% percent chance that she will stay clutching the couch. If the baby is one or more steps away from the couch, there is a 25% chance that she will take a step forward, a 25% chance she will stay in place and a 50% chance she will take one step back toward the couch. In the long run, what percent of the time does the baby choose to clutch the couch?

8: Louie lives in a town that has a 50 percent chance of rain each morning and an independent 40 percent chance each evening. Louie walks to and from work each day, bringing an umbrella with him if it is raining and not bringing one if it isn't. On Sunday night, 2 of his 3 umbrellas were with him at home and 1 was at his office. What were the chances that he made it through the work week without getting wet?

9: You are going to play a game. Like many probability games, this involves an infinite supply of ping-pong balls labeled with integers. The balls are numbered 1 through N, so there are infinitely many labeled 1, infinitely many labeled 2, and so on up to N. There is also a group of N cups, labeled 1 through N, each of which can hold an unlimited number of ping-pong balls. The game is played in rounds. A round is composed of 2 phases: throwing and pruning.

During the throwing phase, the player takes balls randomly, 1 at a time, from the supply and tosses them at the cups. The throwing phase is over when every cup contains at least 1 ping-pong ball. Next comes the pruning phase. During this phase the player goes through all the balls in each cup and removes any ball whose number does not match the containing cup.

Every ball drawn has a uniformly random number from 1 to N, every ball lands in a uniformly random cup, and every throw lands in some cup. The game is over when, after a round is completed, there are no empty cups.

(a) How many balls would you expect to need to draw and throw to finish this game?

(b) For $N = 3$, how many rounds would you expect to need to play to finish this game?

Classes of States and Stationary Distributions

The states of a Markov chain partition into equivalence classes according to their type: communicating, periodic versus aperiodic, recurrent versus transient, or positive recurrent versus null recurrent. This chapter investigates the relationships among these classes. The chapter concludes with several proofs of the Fundamental Theorem of Markov Chains.

Mathematicians Only: prolonged scenes of intense rigor.

2.1. Chapter Starter Question

Randomly distribute three balls between two urns, labeled A and B. Each period, select an urn at random, and if it is not empty, remove a ball from that urn and put it in the other urn. How can you find the fraction of time urn A is empty? What is that fraction of time? Does this depend on how the balls are initially distributed between the two urns?

2.2. Classification of States

Definition 2.1. State j is **accessible** from state i if $(P^n)_{ij} > 0$ for some $n \geq 1$. That is, state j is accessible from state i if and only if it is possible that the process will enter state j in n steps, for some n. ◇

Definition 2.2. Two states **communicate** if state j is accessible from state i and state i is accessible from j. ◇

Remark 2.3. Note that it is possible that j is accessible from i but that i and j do not communicate. An example is if j is an absorbing state accessible from i.

Proposition 2.4. *Communicating is an equivalence relation, so that the states of a Markov chain partition into disjoint communicating classes.*

Proof. Reflexive:: If i communicates with itself, then i is accessible from i in n steps, and so i is accessible from itself again in n steps, so communication is reflexive.

Symmetric:: If i and j communicate, then state j is accessible from state i in n_{ij} steps and state i is accessible from state j in n_{ji} steps. Therefore, state i is accessible from state j in n_{ji} steps and state j is accessible from state i in n_{ij} steps. That is, communication between i and j is a symmetric relation.

Transitive:: If i communicates with j and j communicates with l, then $(P^{n_{ij}})_{ij} > 0$ for some $n_{ij} \geq 1$, and likewise $(P^{n_{jl}})_{jl} > 0$ for some $n_{jl} \geq 1$. Then $(P^{n_{ij}+n_{jl}})_{il} > (P^{n_{ij}})_{ij}(P^{n_{jl}})_{jl} > 0$ and so it is possible that the process can enter state l from i in $n = n_{ij} + n_{jl}$ steps, so l is accessible from i. Likewise, $(P^{n_{lj}})_{lj} > 0$ for some $n_{lj} \geq 1$, and $(P^{n_{ji}})_{ji} > 0$ for some $n_{ji} \geq 1$. Then $(P^{n_{lj}})_{lj}(P^{n_{ji}})_{ji} > 0$ and so it is possible that the process can enter state i in $n = n_{lj} + n_{ji}$ steps, so i is accessible from l. That is, i and l communicate and communication is transitive.

$\square$

Definition 2.5. The period of a state i, written as $d(i)$, is the greatest common divisor of all integers $n \geq 1$ for which $(P^n)_{ii} > 0$. If $(P^n)_{ii} = 0$ for all n, define $d(i) = 0$. $\Diamond$

Proposition 2.6. *(1) If i communicates with j, then $d(i) = d(j)$.*

(2) If state i has period $d(i)$, then there exists an integer N depending on i such that for all integers $n \geq N$, $(P^{nd(i)})_{ii} > 0$. That is, return to state i can occur at all sufficiently large multiples of the period $d(i)$.

(3) If $(P^m)_{ij} > 0$, then $(P^{m+nd(i)})_{ij} > 0$ for all sufficiently large n.

Remark 2.7. The proofs of the second and third parts rely on the next lemma from number theory, a generalization of a standard result about the greatest common divisor of two integers.

Lemma 2.8. *Let $n_1, n_2, \ldots, n_s$ be positive integers with greatest common divisor d. Then there exists a positive integer M such that if $m \geq M$ then there exist nonnegative integers $c_1, c_2, \ldots, c_s$ such that*

$$AD = \sum_{\nu=1}^{s} c_\nu n_\nu.$$

Proof. (1) Let

$$A = \{n : n = c_1 n_1 + \cdots + c_s n_s, c_s \text{nonnegative integers}\}$$

and

$$B = \{b_1 m_1 + \cdots + b_j m_j : m_i \in A, b_i \in \mathbb{Z}, j \geq 1, i = 1, \ldots, j\}.$$

(2) Let d' be the least *positive* integer in B, where $d' = b_1 m_1' + \cdots + b_j m_j'$. Since $m_i' \in A$ for all i, each can be written as $m_i' = c_{1i}' n_1 + \cdots + c_{si}' n_s$.

(3) The first claim is that d' is a common divisor of all elements of A. To prove the claim, let $a = c_1 n_1 + \cdots + c_s n_s \in A$. By the Division Algorithm let $a = d'q + r$ with $0 \le r < d'$. Then

$$
\begin{aligned}
r &= a - d'q \\
&= a - q(b_1 m_1' + \cdots + b_j m_j') \\
&= (c_1 n_1 + \cdots + c_s n_s) - q\left(\sum_{i=1}^{j} b_i \sum_{\nu=1}^{s} c_{\nu i}' n_\nu \right) \\
&= \left(c_1 - q\sum_{i=1}^{j} b_i c_{1i}' \right) n_1 + \cdots + \left(c_s - q\sum_{i=1}^{j} b_i c_{si}' \right) n_s.
\end{aligned}
$$

(4) This means that $r \in B$ (with $m_i = 1 \cdot n_i \in A$ and $b_i = c_i - q\sum_{\nu=1}^{j} b_\nu c_{i\nu}'$). Since $0 \le r < d'$, and d' is the least *positive* integer in B, then $r = 0$ and d' is a divisor of a, establishing the first claim.

(5) For simplicity drop the prime on d' so $d = d' = \gcd(n_1, \ldots, n_s)$. As in the previous step, rearrange the terms in the representation of $d = b_1' m_1 + \cdots + b_j' m_j = \alpha_1 n_1 + \cdots + \alpha_s n_s$ so that the terms with positive coefficients are written first. Thus $d' = N_1 - N_2$ with $N_1 \in A$ and $N_2 \in A$. Let M be the positive integer N_2^2/d (a positive integer since d' is a divisor of $N_2 \in A$). Every integer $m \ge M$ can be written as $m = M + K = N_2^2/d + K$, with $K \ge 0$. Using the Division Algorithm $K = q'(N_2/d) + r'$ with $0 \le r' < N_2/d$ and the integer $q' = \lfloor K/(N_2/d) \rfloor$. Now

$$
\begin{aligned}
md &= (N_2^2/d + q'N_2/d + r')d \\
&= N_2^2 + q'N_2 + r'd \\
&= N_2^2 + q'N_2 + r'(N_1 - N_2) \\
&= N_2(N_2 + q' - r') + r'N_1.
\end{aligned}
$$

Recall that $r' < N_2/d$, so $r' \le N_2$.

(6) Since $N_2 \in A$ and $N_1 \in A$, the previous step shows there exist nonnegative integers $c_1, c_2, \ldots, c_s$ such that

$$
md = \sum_{\nu=1}^{s} c_\nu n_\nu.
$$

$\square$

Proof of Proposition 2.6. (1) (a) If i communicates with j, then $(P^{n_{ij}})_{ij} > 0$ for some $n_{ij} \ge 1$ and $(P^{n_{ji}})_{ji} > 0$ for some $n_{ji} \ge 1$. Additionally since i and j communicate, $(P^{n_{ii}})_{ii} > 0$ for some n_{ii}.

(b) Then

$$
(P^{n_{ji}+n_{ii}+n_{ij}})_{jj} \ge (P^{n_{ji}})_{ji} \cdot (P^{n_{ii}})_{ii} \cdot (P^{n_{ij}})_{ij} > 0.
$$

Furthermore, $(P^{2n_{ii}})_{ii} \ge (P^{n_{ii}})_{ii} \cdot (P^{n_{ii}})_{ii} > 0$, so

$$
(P^{n_{ji}+2n_{ii}+n_{ij}})_{jj} \ge (P^{n_{ji}})_{ji} \cdot (P^{2n_{ii}})_{ii} \cdot (P^{n_{ij}})_{ij} > 0.
$$

(c) This means that $d(j)$ divides $n_{ji}+n_{ii}+n_{ij}$ and $d(j)$ divides $n_{ji}+2n_{ii}+n_{ij}$ so $d(j)$ divides the difference n_{ii}.

(d) Now $d(j)$ is a divisor of n_{ii} and $d(i) = \gcd(n_{ii})$ so $d(j) \leq d(i)$.

(e) By a symmetrical argument, $d(i) \leq d(j)$ so $d(i) = d(j)$.

(2) (a) For state i, let its equivalence class of communicating states be labeled $1, \ldots, k$. For each state j, $1 \leq j \leq k$, there is an m_{ij} such that $(P^{m_{ij}})_{ij} > 0$ and an m_{ji} such that $(P^{m_{ji}})_{ji} > 0$.

(b) Let $n_j = m_{ij} + m_{ji}$. Then

$$(P^{n_j})_{ii} = (P^{m_{ij}+m_{ji}})_{ii} \geq (P^{m_{ij}})_{ij}(P^{m_{ji}})_{ji} > 0.$$

That is, the probability of a return to i from i in n_j steps is at least as great as the probability of an excursion from i to j in m_{ij} steps with return to i from j in m_{ji} steps.

(c) Using the Markov property of independence of the past and the future given the present state, $(P^{n_j})_{ii} > 0$ implies that for any positive integer c_j, $(P^{c_j n_j})_{ii} > 0$, that is, the probability of c_j sequential returns is still positive.

(d) By the number theoretic Lemma 2.8, there exists N and positive integers $c_1, \ldots, c_k$ such that if $n \geq N$

$$(P^{nd(i)})_{ii} = (P^{c_1 n_1 + \cdots + c_k n_k})_{ii} \geq (P^{c_1 n_1})_{ii} \cdots \cdot (P^{c_k n_k})_{ii} > 0.$$

(3) Using the hypothesis

$$(P^{m+nd(i)})_{ji} \geq (P^{nd(i)})_{ii}(P^m)_{ji} > 0.$$

That is, the probability of an excursion to j from i in $m + nd(i)$ steps is at least as great as the probability of an excursion from i to i in $nd(i)$ steps followed by moving to j from i in m steps. By the previous part of the lemma, there is an N such that for $n \geq N$, $(P^{nd(i)})_{ii} > 0$ and so $(P^{m+nd(i)})_{ji} > 0$ for sufficiently large n.

$\square$

Definition 2.9. The Markov chain is **irreducible** if it has only one communicating class. $\diamond$

Definition 2.10. A Markov chain in which each state has period 1 is **aperiodic**. $\diamond$

2.3. Recurrent and Transient States

Definition 2.11. Let the random variable τ_{ij} be the **first return time** to state j given the chain starts in i:

$$\tau_{ij} = \min \{n \geq 1 : X_n = j \mid X_0 = i\}$$
$$\tau_{ij} = \infty \text{ if } X_n \neq j, n \geq 1.$$

$\diamond$

Remark 2.12. The first return time to state j given the chain starts in i might be called the *first visit time*. But the most frequent application is when $j = i$ in which case the chain returns to i, accounting for the name

Definition 2.13. Let the distribution of τ_{ij} be f_{ij}^n, the probability that starting from i, the first return to state j occurs at step n. That is, for each $n \geq 1$,

$$f_{ij}^n = \mathbb{P}\left[X_n = j, X_\nu \neq i, \nu = 1, 2, \ldots, n-1 \mid X_0 = i\right].$$

For convenience, $f_{ij}^0 = 0$. Note that $f_{ij}^1 = P_{ij}$. ◇

Definition 2.14. Absorbing states are a special case of states for which $f_{ii}^1 = P_{ii} = 1$. ◇

Lemma 2.15. *Recursively $f_{ij}^n = \sum_{\nu=0}^n f_{ij}^\nu P_{jj}^{\nu-k}$.*

Proof. See the exercises. □

Remark 2.16. The recursive equation is an example of a *renewal equation.*

Definition 2.17. The **expected return time** is $\mathbb{E}\left[\tau_{ii}\right] = m_i = \sum_{\nu=0}^\infty \nu f_{ii}^\nu$. ◇

Definition 2.18. For any state i, let $f_i = \mathbb{P}\left[\tau_{ii} < \infty\right] = \sum_{\nu=0}^\infty f_{ii}^\nu$ denote the probability that starting in i the process will *ever* reenter state i. State i is **recurrent** if $f_i = 1$ and **transient** if $f_i < 1$. ◇

Remark 2.19. Note that the qualities of recurrence and transience are complementary, a state is one or the other, there are no other possibilities. If the state i is transient, $f_i < 1$, then i will be visited a finite random number of times. Starting from i the number of visits N_i (counting the starting visit) to transient state i has a geometric distribution

$$\mathbb{P}\left[N_i = n\right] = f_i^{n-1}(1 - f_i), n \geq 1.$$

This observation depends on the Strong Markov property, once the chain visits state i, the future is independent of the past and the chain starts over again with the same distribution. In fact, the expected number of visits is $\mathbb{E}\left[N_i\right] = 1/(1 - f_i)$.

Proposition 2.20. *If state i is recurrent, then the process will reenter i infinitely often.*

Proof. (1) Suppose that the process starts in state i and i is recurrent, so that with probability 1 the process will eventually reenter state i.

(2) However, by the definition of a Markov chain, it follows that the process will be starting over again when it reenters state i, and therefore state i will eventually be visited again.

(3) By induction, starting in state i will reenter state i any number of times, that is, infinitely often.

□

Proposition 2.21. *(1) State i is recurrent if*

$$\sum_{\nu=1}^\infty (P^\nu)_{ii} = \infty.$$

(2) State i is **transient** if

$$\sum_{\nu=1}^{\infty} (P^{\nu})_{ii} < \infty.$$

Proof. (1) The two statements are complementary and equivalent so it suffices to prove the statement for transience.

(2) Suppose that state i is transient, so by definition $f_i < 1$. Let N_i count the total number of returns to i.

(3) Write N_i in terms of indicator variables as

$$N_i = \sum_{\nu=1}^{\infty} \mathbf{1}_{[X_\nu = i]}.$$

(4) The distribution of N_i is geometric with parameter f_i so

$$\mathbb{E}\left[N_i \mid X_0 = i\right] = \frac{f_i}{1 - f_i}.$$

Then

$$\infty > \mathbb{E}\left[N_i \mid X_0 = i\right] = \mathbb{E}\left[\sum_{n=1}^{\infty} \mathbf{1}_{[X_n = i | X_0 = i]}\right] = \sum_{n=1}^{\infty} (P^n)_{ii}.$$

$\square$

Corollary 2.22. *(1) For any communication class C of a Markov chain, all states in C are either recurrent or all states in C are transient. That is*
- *if i and j communicate, and i is recurrent, then so is j*
- *if i and j communicate, and i is transient, then so is j.*

(2) For an irreducible Markov chain, either all states are recurrent or all states are transient.

Proof. Left as an exercise. $\square$

2.4. Positive Recurrence

Definition 2.23. A state i is **positive recurrent** if it is recurrent and starting in i, the expected time until the process returns to state i is finite. Otherwise the state is **null recurrent**. $\lozenge$

Definition 2.24. Let the random variable τ_{ij} be the **first passage time** to state j given the chain starts in i:

$$\tau_{ij} = \min\{n \geq 1 : X_n = j \mid X_0 = i\}$$
$$\tau_{ij} = \infty \text{ if } X_n \neq j, n \geq 1.$$

$\lozenge$

Proposition 2.25. *Suppose states i, j are both recurrent. If i and j communicate and if j is positive recurrent, $\mathbb{E}\left[\tau_{jj}\right] < \infty$, then i is positive recurrent and furthermore $\mathbb{E}\left[\tau_{ij}\right] < \infty$. In particular, all states in a recurrent communicating class are either all positive recurrent or all null recurrent.*

Proof. (1) Assume $\mathbb{E}\left[\tau_{jj}\right] < \infty$ and i and j communicate.

(2) Choose the smallest $n \geq 1$ such that $(P^n)_{ji} > 0$.

(3) With $x_0 = j$, let $A = \{x_l \neq j, 1 \leq l < n, x_n = i\}$ be a path of states visited from j to i. Note that $\mathbb{P}\left[A\right] > 0$.

(4) Then $\mathbb{E}\left[\tau_{jj}\right] \geq \mathbb{E}\left[\tau_{jj} \mid A\right] \cdot \mathbb{P}\left[A\right] = (n + \mathbb{E}\left[\tau_{ij}\right]) \cdot \mathbb{P}\left[A\right]$. Hence $\mathbb{E}\left[\tau_{ij}\right] < \infty$.

(5) With $x_0 = j$, let $\{Y_m : m \geq 1\}$ be the interarrival times between visits to state j. The distribution of Y_m is the same as the distribution of τ_{jj}. The nth revisit of the chain to state j is at time $t_n = Y_1 + \cdots + Y_n$ with $\mathbb{E}\left[Y_l\right] = \mathbb{E}\left[\tau_{jj}\right] < \infty$.

(6) Let p be the probability the chain visits state i before returning to state j, given that the chain started in j. Then $p \geq \mathbb{P}\left[A\right] > 0$ where A is defined in step 3.

(7) Every time the chain revisits state j, there is, independent of the past, probability p that the chain will visit state i before revisiting state j again. Letting N denote the number of revisits the chain makes to state j until first visiting state i, N has geometric distribution with "success" probability p and so $\mathbb{E}\left[N\right] < \infty$.

(8) N is a stopping time with respect to the $\{Y_m\}$ and

$$\tau_{ji} \leq \sum_{m=1}^{N} Y_m$$

and so by Wald's equation $\mathbb{E}\left[\tau_{ji}\right] \leq \mathbb{E}\left[N\right] \cdot \mathbb{E}\left[Y\right] < \infty$.

(9) Finally, $\mathbb{E}\left[\tau_{ii}\right] \leq \mathbb{E}\left[\tau_{ij}\right] + \mathbb{E}\left[\tau_{ji}\right] < \infty$.

$\square$

Proposition 2.26. *In an irreducible finite-state Markov chain all recurrent states are positive recurrent and have a stationary distribution satisfying $\pi = P\pi$.*

Proof. (1) If the state space is finite, then not *all* states can be transient. Otherwise, after a finite number of steps, the chain would leave every state never to return, clearly an impossibility. That is, all the communicating states are recurrent.

(2) Now show that state i can't be null recurrent, that is, it must be positive recurrent.

(3) For any fixed $m \geq 1$, the rows of P^m must sum to 1. That is, for any $i \in \mathcal{X}$

$$\sum_{j \in \mathcal{X}} (P^m)_{ij} = 1.$$

(4) If the state is null recurrent, then for any $j \in \mathcal{X}$.

$$\lim_{n \to \infty} \frac{1}{n} \sum_{\nu=1}^{n} (P^\nu)_{ij} = 0.$$

(5) Summing over j

$$\sum_{j \in \mathcal{X}} \lim_{n \to \infty} \frac{1}{n} \sum_{\nu=1}^{n} (P^\nu)_{ij} = 0.$$

(6) Since the state space is finite, interchange the outer sum and the limit.

$$\lim_{n \to \infty} \sum_{j \in \mathcal{X}} \frac{1}{n} \sum_{\nu=1}^{n} (P^\nu)_{ij} = 0.$$

(7) Interchange the order of summation,

$$\lim_{n \to \infty} \frac{1}{n} \sum_{\nu=1}^{n} \sum_{j \in \mathcal{X}} (P^\nu)_{ij} = 0.$$

(8) But

$$\lim_{n \to \infty} \frac{1}{n} \sum_{\nu=1}^{n} \sum_{j \in \mathcal{X}} (P^\nu)_{ij} = \lim_{n \to \infty} \frac{1}{n} \lim_{n \to \infty} \frac{1}{n} \sum_{\nu=1}^{n} \sum_{j \in \mathcal{X}} (P^\nu)_{ij} \sum_{\nu=1}^{n} 1 = \lim_{n \to \infty} \frac{1}{n} \cdot n = 1,$$

a contradiction.

(9) Thus, the state must be positive recurrent.

$\square$

Definition 2.27. Positive recurrent and aperiodic states are **ergodic**. $\Diamond$

Remark 2.28. This text does not use the term *ergodic*. Instead the fully descriptive terms *positive recurrent* and *aperiodic* emphasize the state properties.

Example 2.29. In infinite state space Markov chains recurrence, transience, positive recurrence and null recurrence are distinct. Symmetric nearest neighbor random walk on $\mathbb{Z}$ and $\mathbb{Z}^2$ is null recurrent. However symmetric nearest neighbor random walk on $\mathbb{Z}^N$ is transient. See [**Les05**, Chapter 13] $\triangledown$

2.5. Fundamental Theorems for Markov Chains

Remark 2.30. Make the notation convention that $(P^0)_{ii} = 1$. Recall the definition of the **expected return time** as $m_i = \mathbb{E}[\tau_{ii}] = \sum_{\nu=0}^{\infty} \nu f_{ii}^\nu$.

Theorem 2.31 (Limit Theorem for Transition Matrix Powers)**.**

$$\lim_{n \to \infty} (P^n)_{ij} = \lim_{n \to \infty} (P^n)_{ii}.$$

Proof. (1) Start with the recursion relation

$$(P^n)_{ji} = \sum_{\nu=0}^{n} f_{ji}^\nu (P^{n-\nu})_{ii}$$

for $i \neq j$ and $n \geq 0$.

(2) Change the order of summation

$$(P^n)_{ji} = \sum_{\nu=0}^{n} f_{ji}^{n-\nu}(P^\nu)_{ii}$$

for $i \neq j$ and $n \geq 0$.

(3) More generally, consider the convolution type relation

$$y_n = \sum_{\nu=0}^{n} a_{n-\nu}x_\nu$$

where $a_m \geq 0$, $\sum_{\nu=0}^{\infty} a_\nu = 1$, and $\lim_{\nu \to \infty} x_\nu = c$. The goal is to prove that $\lim_{\nu \to \infty} y_\nu = c$.

(4) Taking the difference with c and break into sums from 0 to n and $n+1$ to ∞

$$y_n - c = \sum_{\nu=0}^{n} a_{n-\nu}x_\nu - c \sum_{\nu=0}^{\infty} a_n = \sum_{\nu=0}^{n} a_{n-\nu}(x_k - c) - c \sum_{\nu=n+1}^{\infty} a_\nu.$$

(5) Let $\epsilon > 0$ be given and determine $N(\epsilon)$ so that $|x_\nu - c| < \epsilon/3$ for $\nu \geq N(\epsilon)$. For $\nu \geq N(\epsilon)$

$$y_n - c = \sum_{\nu=0}^{N(\epsilon)} a_{n-\nu}(x_k - c) + \sum_{\nu=N(\epsilon)+1}^{n} a_{n-\nu}(x_k - c) - c \sum_{\nu=n+1}^{\infty} a_\nu.$$

Then

$$|y_n - c| \leq M \sum_{\nu=0}^{N(\epsilon)} a_{n-\nu} + \frac{\epsilon}{3} \sum_{\nu=N(\epsilon)+1}^{n} a_{n-\nu} + |c| \sum_{\nu=n+1}^{\infty} a_\nu.$$

(6) Choose $N(\epsilon)$ so that $|c| \sum_{\nu=n+1}^{\infty} a_\nu \leq \epsilon/3$ and

$$\sum_{\nu=0}^{N(\epsilon)} a_{n-\nu} = \sum_{\nu=n-N(\epsilon)}^{n} a_\nu \leq \frac{\epsilon}{3M}$$

for $n \geq N(\epsilon)$.

(7) Summarizing, for $n \geq N(\epsilon)$, $|y - c| \leq \epsilon/3 + \epsilon/3 + \epsilon/3 = \epsilon$.

(8) Applying this to $y_n = (P^n)_{ji}$, $a_n = f_{ji}^n$, $x_n = (P^n)_{ii}$ gives the final result.

$\square$

Remark 2.32. This proof is a purely analytic result with the probabilistic application due to the assumption that $\sum_{\nu=0}^{\infty} a_\nu = 1$.

Lemma 2.33. *(1) Let a_n, u_n, b_n be sequences for $n \geq 0$.*

(2) $a_n \geq 0$, $a_1 > 0$, and $\sum_{\nu=0}^{\infty} a_\nu = 1$.

(3) The greatest common divisor of the integers n for which $a_n > 0$ is 1.

(4) $b_n \geq 0$ and $b_n \to 0$ as $n \to \infty$.

(5) The renewal equation

$$u_n - \sum_{\nu=0}^{n} a_{n-\nu}u_\nu = b_n$$

is satisfied by a bounded sequence u_n.

Then

$$\lim_{n \to \infty} u_n = \frac{\sum_{\nu=0}^{\infty} b_\nu}{\sum_{\nu=0}^{\infty} \nu a_\nu}.$$

Proof. (1) Changing the variable of summation, the renewal equation is equivalent to

$$u_n - \sum_{\nu=0}^{n} a_\nu u_{n-\nu} = b_n.$$

(2) For $n = 0$, the renewal equation is $u_0 - a_0 u_0 = b_0$. Using hypotheses 2 and 4, $u_0 \geq 0$. Inductively,

$$u_n - a_0 u_n - \sum_{\nu=1}^{n} a_\nu u_{n-\nu} = b_n$$

so $u_n \geq 0$.

(3) By hypothesis 4, u_n is a bounded sequence, so $\lambda = \limsup_{n \to \infty} u_n < \infty$. Let $n_1 < n_2 < \ldots$ be a subsequence along which $\lim_{j \to \infty} u_{n_j} = \lambda$. The claim is that $\lim_{j \to \infty} u_{n_j - 1} = \lambda$ by using $a_1 > 0$ from hypothesis 2.

(4) Proof of the claim by contradiction.
   (a) Suppose $\lim_{j \to \infty} a_{n_j - 1} \neq \lambda$. Then there is λ' such that $u_{n_j} < \lambda'$ for infinitely many j. Use $\epsilon = a_1(\lambda - \lambda')/4$, $M = \sup_{\nu \geq 0} u_\nu$, and N so large that $\sum_{\nu=N}^{\infty} a_\nu > 1 - \epsilon/M$ for $n > N$. Let j be large enough that $n_j > N$ and $u_{n_j} > \lambda - \epsilon$, $u_{n_j - 1} < \lambda' < \lambda$, $0 \leq b_{n_j} < \epsilon$ and $u_n < \lambda + \epsilon$ for $n_j - N$.
   (b) Then

$$u_{n_j} \leq \sum_{\nu=0}^{n_j} a_\nu u_{n_j - \nu} + \epsilon$$

$$< \sum_{\nu=0}^{N} a_\nu u_{n_j - \nu} + M \sum_{\nu=N+1}^{n_j} a_k + \epsilon$$

$$< \sum_{\nu=0}^{N} a_\nu u_{n_j - \nu} + 2\epsilon$$

$$< (a_0 + a_2 + \cdots + a_N)(\lambda + \epsilon) + a_1 \lambda' + 2\epsilon$$

$$< (1 - a_1)(\lambda + \epsilon) + a_1 \lambda' + 2\epsilon$$

$$< \lambda + 3\epsilon - a_1(\lambda - \lambda')$$

$$= \lambda - \epsilon.$$

   (c) The inequality from the previous step contradicts $u_{n_j} > \lambda - \epsilon$ and the claim is established, $\lim_{j \to \infty} u_{n_j - 1} = \lambda$.

(5) Repeating the argument for $d \geq 0$, $\lim_{j \to \infty} u_{n_j - d} = \lambda$.

(6) Let $r_n = \sum_{\nu=n+1}^{\infty} a_\nu$. Then by summation by parts, $\sum_{\nu=0}^{\infty} k a_k = \sum_{\nu=0}^{\infty} r_n$. No assumption is made about the convergence of $\sum_{\nu=0}^{\infty} k a_k = \sum_{\nu=0}^{\infty} r_n$. Note that $a_n = r_{n-1} - r_n$ for $n \geq 1$. Also note that $r_0 = 1 - a_0$.

(7) Starting with the renewal equation in the form

$$u_n - a_0 u_n - a_1 u_{n-1} - \cdots a_n u_0 = b_n$$

and substituting $-a_\nu = r_\nu - r_{\nu-1}$ gives

$$u_n - a_0 u_n + (r_1 - r_0)u_{n-1} + \cdots + (r_n - r_{n-1})u_0 = b_n$$

Rearranging and using $r_0 = 1 - a_0$

$$r_0 u_n + r_1 u_{n-1} + \cdots r_n u_0 = r_0 u_{n-1} + r_1 u_{n-2} + \cdots + r_{n-1} u_0 + b_n.$$

(8) Set $A_n = r_0 u_n + \cdots + r_n u_0$, the previous equation becomes $A_n = A_{n-1} + b_n$ with $A_0 = r_0 u_0 = (1 - a_0)u_0 = b_0$. Then $A_n = \sum_{\nu=0}^{n} b_\nu$.

(9) Since $r_n \geq 0$ and $u_n \geq 0$, for any $N > 0$ and $j > 0$

$$r_0 u_{n_j} + r_1 u_{n_j-1} + \cdots r_N u_{N_j-N} \leq A_{n_j} = \sum_{\nu=0}^{n_j} b_\nu.$$

(10) Letting $j \to \infty$ gives

$$(r_0 + \cdots + r_N)\lambda \leq \sum_{\nu=0}^{\infty} b_\nu$$

or equivalently $\lambda \leq \sum_{\nu=0}^{\infty} b_\nu / \sum_{\nu=0}^{N} r_\nu$.

(11) Since $N > 0$ is arbitrary, $\lambda \leq \sum_{\nu=0}^{\infty} b_\nu / \sum_{\nu=0}^{\infty} r_\nu$.

(12) Since $u_n \geq 0$ for all n, this proves the theorem in the case $\sum_{\nu=0}^{\infty} k a_k = \sum_{\nu=0}^{\infty} r_n = \infty$ since then $\lambda = \lim_{n\to\infty} u_n = 0$.

(13) If $\sum_{\nu=0}^{\infty} r_n < \infty$, let $\mu = \liminf_{n\to\infty} u_n$. As in the previous step, then $\lim_{j\to\infty} u_{n_j-d} = \mu$ for each integer d.

(14) Set $g_N = \sum_{\nu=N+1}^{\infty} r_n$ so $\lim_{N\to\infty} g_N = 0$ and

$$\sum_{\nu=0}^{n_j} b_\nu \leq r_0 u_{n_j} + r_1 u_{n_j-1} + \cdots r_N u_{N_j-N} + g_N \cdot M.$$

(15) Letting $j \to \infty$

$$\sum_{\nu=0}^{n_j} b_\nu \leq (r_0 + r_1 + \cdots r_N)\mu + g_N \cdot M$$

and taking the limit as $N \to \infty$

$$\sum_{\nu=0}^{\infty} b_\nu \leq \mu \sum_{\nu=0}^{\infty} r_\nu$$

or $\mu \geq \sum_{\nu=0}^{\infty} b_\nu / \sum_{\nu=0}^{\infty} r_\nu$.

(16) Steps 10 and 15 imply $\mu \geq \lambda$ which means $\mu = \lambda = \lim_{n\to\infty} u_n$ with value

$$\lim_{n\to\infty} u_n = \sum_{\nu=0}^{\infty} b_\nu \Big/ \sum_{\nu=0}^{\infty} r_\nu .$$

$\square$

Theorem 2.34 (Power Limit as Reciprocal Mean Return Time)**.** *For an irreducible, recurrent, and aperiodic Markov chain,*

$$\lim_{n \to \infty} (P^n)_{ii} = \frac{1}{m_i},$$

and

Proof. (1) The renewal equation for the transition probabilities is

$$P^n_{ii} - \sum_{\nu=0}^{n} f^{n-\nu}_{ii}(P^\nu)_{ii} = \begin{cases} 1 & n = 0, \\ 0 & n > 0. \end{cases}$$

(2) Using the previous Lemma, identify $u_n = (P^n)_{ii}$, $a_n = f^n_{ii}$, $b_0 = 1$, $b_n = 0$ for $n > 0$.

(3) Then

$$\lim_{n \to \infty} u_n = \sum_{\nu=0}^{\infty} b_\nu \bigg/ \sum_{\nu=0}^{\infty} r_\nu$$

$$= 1 \bigg/ \sum_{\nu=0}^{\infty} \nu f^n_{ii}$$

$$= 1/m_i.$$

$\square$

Lemma 2.35. *If $a_n \to a$ as $n \to \infty$, then*

$$\lim_{n \to \infty} \frac{1}{n} \sum_{\nu=1}^{n} a_\nu = a,$$

so that if a_n converges to a, the sequence of averages converges to a.

Proof. Left as an exercise. $\square$

Definition 2.36. When the limit exists, let

$$\pi_j = \lim_{n \to \infty} \frac{1}{n} \sum_{\nu=1}^{n} \mathbf{1}_{[X_\nu = j | X_0 = i]}$$

denote the **long run proportion of time the chain spends in state** j or just the *long run proportion* for short. Note that the right side appears to depend on the starting state i, but the Limit Theorem shows that the long run proportion is independent of i. $\Diamond$

Corollary 2.37 (Long Run Proportion as Reciprocal Mean Return Time)**.**

$$\pi_i = \lim_{n \to \infty} \frac{1}{n} \sum_{\nu=1}^{n} (P^\nu)_{ii} = \frac{1}{\sum_{\nu=0}^{\infty} \nu f^\nu_{ii}} = \frac{1}{m_i},$$

Remark 2.38. Taking expected values,

$$\mathbb{E}\left[\mathbf{1}_{[X_m=j|X_0=i]}\right] = \mathbb{P}\left[X_m = j \mid X_0 = i\right] = (P^m)_{ij}$$

so the long run proportion is alternatively defined as

$$\pi_j = \lim_{n\to\infty} \frac{1}{n} \sum_{\nu=1}^{n} (P^\nu)_{ij}$$

using the Bounded Convergence Theorem to interchange the limit and the expectation.

Remark 2.39. Recalling that $(P^\nu)_{ij}$ is the i, j component of the matrix P^ν and using the independence on i, the long run proportions in matrix form are

$$\lim_{n\to\infty} \frac{1}{n} \sum_{\nu=1}^{n} P^\nu = \begin{pmatrix} \pi_1 & \pi_2 & \dots & \pi_k \\ \pi_1 & \pi_2 & \dots & \pi_k \\ \vdots & \vdots & \ddots & \vdots \\ \pi_1 & \pi_2 & \dots & \pi_k \end{pmatrix} = \begin{pmatrix} \pi \\ \pi \\ \vdots \\ \pi \end{pmatrix}.$$

Remark 2.40. Observing the Markov chain at a time far in the future, π_j is approximately the probability the chain is in state j. To see this: Let N be a random observation time, with uniform distribution over the integers $\{1, 2, \dots, n\}$, independent of the chain. Then conditioning on N

$$\mathbb{P}\left[X_N = j\right] = \sum_{\nu=1}^{n} \mathbb{P}\left[X_\nu = j \mid X_0 = i\right] \mathbb{P}\left[N = \nu\right] = \frac{1}{n} \sum_{\nu=1}^{n} (P^\nu)_{ij} \approx \pi_j.$$

Theorem 2.41 (Fundamental Theorem of Markov Chains)**.** *For an irreducible, positive recurrent and aperiodic Markov chain $\lim_{n\to\infty}(P^n)_{ij}$ exists and is independent of i. Furthermore, letting*

$$\pi_j = \lim_{n\to\infty} (P^n)_{ij}$$

then π_j is the unique nonnegative solution of

(2.1) $$\sum_i \pi_i P_{ij} = \pi_j,$$

(2.2) $$\sum_i \pi_i = 1.$$

Proof. (1) Recall k is the number of states in the Markov chain, possibly $k = \infty$.

(2) For every n and $M \le k$, $1 = \sum_{j=0}^{k}(P^n)_{ij} \ge \sum_{j=0}^{M}(P^n)_{ij}$.

(3) Letting $n \to \infty$, the limit exists, and using the Limit Theorem for Transition Matrix Powers, $1 \ge \sum_{j=0}^{M} \pi_j$ for every M. Thus, $1 \ge \sum_{j=0}^{k} \pi_j$.

(4) Now $(P^{n+1})_{ij} \ge \sum_{\nu=0}^{M}(P^n)_{i\nu}P_{\nu j}$. Again let $n \to \infty$ to obtain

$$\pi_j \ge \sum_{\nu=0}^{M} \pi_\nu P_{\nu j}.$$

(5) Since the left side is independent of M, let $M \to k$ giving

$$\pi_j \geq \sum_{\nu=0}^{k} \pi_\nu P_{\nu j}.$$

(6) Multiply π_j by P_{ji} to observe

$$\pi_j P_{ji} = \lim_{n \to \infty} (P^n)_{ij} P_{ji} = \lim_{n \to \infty} (P^{n+1})_{ii} = \pi_i.$$

(7) Multiply by P_{ji}, sum on j,

$$\pi_j P_{ji} = \pi_i \geq \sum_{\nu=0}^{\infty} \pi_\nu (P^2)_{\nu i}$$

and then by induction and renaming the state index from i back to j

$$\pi_j \geq \sum_{\nu=0}^{\infty} \pi_\nu (P^n)_{\nu j}.$$

(8) In the previous inequality, suppose strict inequality holds for some j, then add over j

$$\sum_{j=0}^{k} \pi_j > \sum_{j=0}^{k} \sum_{\nu=0}^{\infty} \pi_\nu (P^n)_{\nu j} = \sum_{\nu=0}^{\infty} \pi_\nu \sum_{j=0}^{k} (P^n)_{\nu j} = \sum_{j=0}^{k} \pi_j$$

which is a contradiction. Thus

$$\pi_j = \sum_{j=0}^{k} \pi_j (P^n)_{jj}.$$

(9) Let $n \to \infty$, and use that $\sum \pi_\nu$ converges and $(P^n)_{\nu j}$ is uniformly bounded to conclude

$$\pi_j = \sum_{\nu=0}^{\infty} \pi_\nu \lim_{n \to \infty} (P^n)_{vj} = \pi_j \sum_{\nu=0}^{\infty} \pi_\nu$$

for every j. Thus $\sum_{\nu=0}^{\infty} \pi_\nu = 1$ since $\pi_j > 0$ by positive recurrence.

(10) Now suppose $x = \{x_n\}$ satisfies equations 2.1 and 2.2. Then as before

$$x_j \geq \sum_{\nu=0}^{k} x_\nu (P^n)_{\nu j}.$$

Let $n \to \infty$ as before to see that

$$x_j = \sum_{\nu=0}^{k} x_\nu (P^n)_{\nu j} = \pi_k \sum_{\nu=0}^{k} x_j = \pi_k.$$

$\square$

Definition 2.42. Any vector (π_j) satisfying equations 2.1 and 2.2 is a **stationary probability distribution** of the Markov chain. A Markov chain started according to a stationary distribution (π_j) will have this distribution for all future times. $\Diamond$

Remark 2.43. A limiting distribution is always a stationary distribution, but the converse is not true. A Markov chain may have a stationary distribution but no limiting distribution. For example, the periodic Markov chain whose transition probability matrix is

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

has no limiting distribution but $\pi = (1/2, 1/2)$ is a stationary distribution. Notice that this Markov chain is not aperiodic so it fails to satisfy the requirements of the Fundamental Theorem.

Remark 2.44. The Fundamental Theorem says that a probability transition matrix for an irreducible, positive recurrent, and aperiodic Markov chain has a *left* eigenvector with corresponding eigenvalue 1. This is a special case of the more general Perron-Frobenius Theorem.

Remark 2.45. The Fundamental Theorem says that under appropriate conditions, the powers of the probability transition matrix converge to the stationary distribution but gives no information about the rate of convergence. The following theorem gives a first indication about the rate of convergence.

Theorem 2.46 (Convergence Theorem by Total Variation). *Suppose that P is the transition probability matrix for an irreducible and aperiodic Markov chain with stationary distribution π. Then there exist constants $\alpha \in (0, 1)$ and $C > 0$ such that*

$$\max_{x \in \mathcal{X}} \|(P^n)_{i\cdot} - \pi\|_{TV} \leq C\alpha^n.$$

Proof. (1) Since P is irreducible and aperiodic, there exist r such that P^r has strictly positive entries.

(2) Let Π be the matrix with $\text{card}(\mathcal{X}) = k$ rows, each of which is the row vector π.

(3) For sufficiently small $\delta > 0$, $(P^r)_{ij} \geq \delta \pi_j$ for $i, j \in \mathcal{X}$.

(4) Let $\theta = 1 - \delta$. Define stochastic matrix Q by

$$P^r = (1 - \theta)\Pi + \theta Q.$$

(5) Note that $M\Pi = \Pi$ for any stochastic matrix and $\Pi M = \Pi$ for any matrix with $\pi M = \pi$.

(6) Claim:

$$P^{rn} = (1 - \theta^n)\Pi + \theta^n Q^n.$$

Proof by Induction: For $n = 1$, this is the definition of Q. Assume the claim is true for n.

$$P^{r(n+1)} = P^{rn}P^r = [(1 - \theta^n)\Pi + \theta^n Q^n]P^r$$
$$= (1 - \theta^n)\Pi P^r + (1 - \theta)\theta^n Q^n \Pi + \theta^{n+1}Q^n Q$$

Use $\Pi P^r = \Pi$ and $Q^n \Pi = \Pi$.

$$P^{r(n+1)} = (1 - \theta^{n+1})\Pi + \theta^{n+1}Q^{n+1}.$$

Hence the relation holds for all n.

(7) Multiply by P^j, and rearrange to obtain

$$P^{rn+j} - \Pi = \theta^n(Q^n P^j - \Pi).$$

(8) Sum the absolute value of row ν on both sides and divide by 2. On the right, the absolute row-ν sum from $(Q^n P^j - \Pi)$ is at most the largest possible total variation distance between distributions (which is at most 1). Hence

$$\|(P^{rn+j})_{i\cdot} - \pi\|_{TV} \leq \theta^n.$$

(9) To finish the proof, let $\alpha = \theta^{1/r}$ and $C = 1/\theta$.

$\square$

2.6. Examples of Stationary Distributions

2.7. Amusing Example of a Stationary Distribution

This finite state Markov chain has interest partly in the humorous context and partly that the context obscures that the Markov chain has more states than is at first obvious. It is also similar to a more serious application, a finite state machine, an abstract machine that can be in exactly one of a finite number of states at any given time. The finite state machine can change from one state to another in response to some inputs. Alternatively, this is an example of a random walk on a graph, illustrated in Figure 1.

> There is a bathroom in your office building that has only one toilet. There is a small sign stuck to the outside of the door that you can slide from "Vacant" to "Occupied" so that no one else will try the door handle (theoretically) when you are inside. Unfortunately, people often forget to slide the sign to "Occupied" when entering, and they often forget to slide it to "Vacant" when exiting.
>
> Assume that 1/3 of bathroom users don't notice the sign upon entering or exiting. Therefore, whatever the sign reads before their visit, it still reads the same thing during and after their visit. Another 1/3 of the users notice the sign upon entering and make sure that it says "Occupied" as they enter. However, half the time they forget to slide it to "Vacant" when they exit. The remaining 1/3 of the users are very conscientious: They make sure the sign reads "Occupied" when they enter, and then they slide it to "Vacant" when they exit. Finally, assume that the bathroom is occupied exactly half of the time, all day, every day.
>
> Two questions about this workplace situation:
> (1) If you go to the bathroom and see that the sign on the door reads "Occupied," what is the probability that the bathroom is actually occupied?
> (2) If the sign reads "Vacant," what is the probability that the bathroom actually is vacant?
> (3) Extra credit: What happens as the percentage of conscientious bathroom users changes?

The first step defines the states and transition probabilities. Because the bathroom can be either *occupied* or *vacant*, and the sign in front can either read "Vacant" or "Occupied", there are at least four states, one for each possible pair of occupation state and sign. However, consider the state "bathroom is occupied and the sign says it's occupied". The states must distinguish between the cases where the person occupying the bathroom is conscientious (they will definitely slide the sign to "Vacant" when they leave) or not (they might leave the sign as "Occupied" after they leave).

Imagine a sequence of short times, say every minute. Using the assumption that the bathroom is occupied exactly half of the time, all day, every day, the transition from state to state each minute is a Markov chain with transition probability

$$\begin{array}{cc} & \begin{array}{cc} O & V \end{array} \\ \begin{array}{c} O \\ V \end{array} & \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix} . \end{array}$$

It is easy to see that this chain is irreducible, positive recurrent and aperiodic. The stationary distribution is $(1/2, 1/2)$, consistent with the assumption that the bathroom is occupied exactly half of the time, all day, every day.

Augment the states to the Markov chain corresponding to the different ways in which the bathroom can be occupied. There are three types of users: oblivious, forgetful, and conscientious, each of proportion $1/3$ and 4 combinations of occupation status and outside sign. There are 12 possible states but some will not occur because of the user habits.

Referring to Figure 1 the possible transitions at each minute are:

(1) A Conscientious person is in the bathroom, and it is Occupied and the sign says Occupied. This state is OCO. Either the person stays and the state remains Occupied with sign Occupied with probability $1/2$, or the conscientious person leaves and the state is VV with probability $1/2$.

(2) A Forgetful person is in the bathroom, and it is Occupied and the sign says Occupied. This state is OFO. Either the person stays and the state remains Occupied with sign Occupied with probability $1/2$, or the forgetful person leaves and the state is VO with probability $1/2 \cdot 1/2 = 1/4$ or the forgetful person leaves, changing the sign and the state is VV with probability $1/2 \cdot 1/2 = 1/4$.

(3) An Oblivious person is in the bathroom, and it is Occupied and the sign says Occupied. Name this state as OOO. Either the person stays and the state remains Occupied with sign Occupied with probability $1/2$, or the oblivious person leaves and the state is VO with probability $1/2$.

(4) A Oblivious person is in the bathroom, and it is Occupied but the sign says Vacant. Name this state as OOV. Either the person stays and the state remains Occupied with sign Vacant with probability $1/2$, or the oblivious person leaves and the state is VV with probability $1/2$.

(5) The bathroom is Vacant, but the sign says Occupied. With probability $1/2$ some person will approach, nevertheless try the door and find the true state of vacancy and enter, dealing with the sign according to their type. Then the

transition is to OCO with probability 1/6, OFO with probability 1/6 and OOO with probability 1/6. With probability 1/2 the state remains VO.

(6) The bathroom is Vacant, and the sign says Vacant. With probability 1/2 some person will approach, nevertheless try the door and easily find the true state of vacancy and enter, dealing with the sign according to their type. Then the transition is to OCO with probability 1/6, OFO with probability 1/6 and OOV with probability 1/6. With probability 1/2 the state remains VV.

A state transition diagram is in Figure 1.
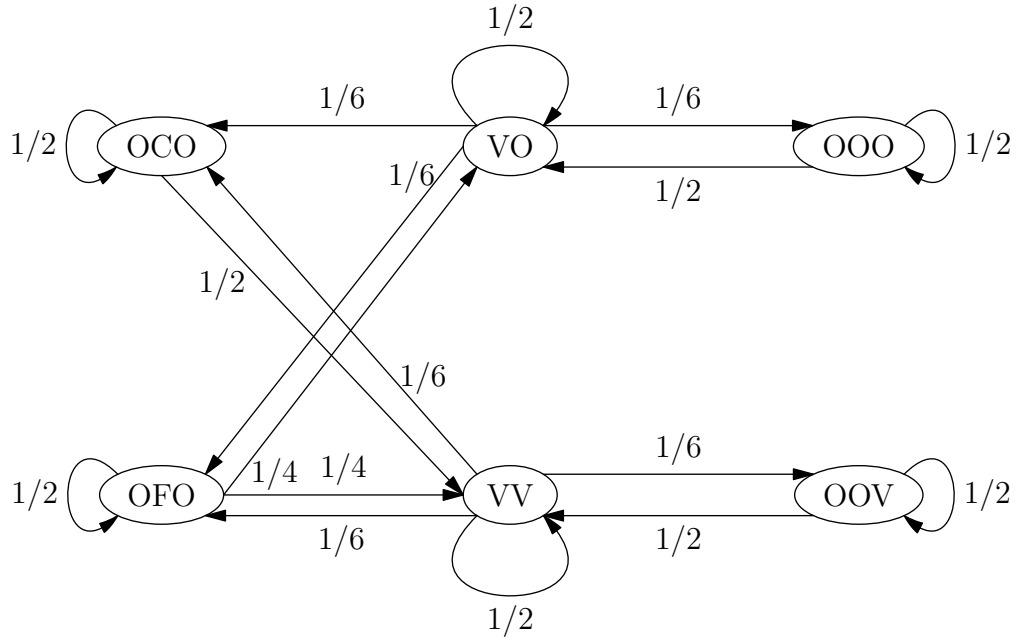
Figure 1. The state transition diagram for bathroom occupancy.

Then the transition probability matrix is

$$
P = \begin{array}{c} \\ OCO \\ OFO \\ OOO \\ OOV \\ VO \\ VV \end{array}
\begin{array}{c} \begin{array}{cccccc} OCO & OFO & OOO & OOV & VO & VV \end{array} \\
\left(\begin{array}{cccccc}
1/2 & 0 & 0 & 0 & 0 & 1/2 \\
0 & 1/2 & 0 & 0 & 1/4 & 1/4 \\
0 & 0 & 1/2 & 0 & 1/2 & 0 \\
0 & 0 & 0 & 1/2 & 1/2 & 0 \\
1/6 & 1/6 & 1/6 & 0 & 1/2 & 0 \\
1/6 & 1/6 & 0 & 1/6 & 0 & 1/2
\end{array} \right) \end{array}.
$$

From Figure 1 it is easy to see that the chain is irreducible and recurrent. Because of the self-loops the states are aperiodic. The stationary distribution is then the solution of

$$
\pi P = \pi, \qquad \sum_i \pi_i = 1
$$

with solution

$$\pi_{OCO} = \frac{1}{6}, \qquad\qquad \pi_{OFO} = \frac{1}{6}, \qquad\qquad \pi_{OOO} = \frac{1}{12},$$

$$\pi_{OOV} = \frac{1}{12}, \qquad\qquad \pi_{VO} = \frac{1}{4}, \qquad\qquad \pi_{VV} = \frac{1}{4}.$$

Note that the bathroom is still Vacant half the time, although the time splits equally over the 2 sign states.

$$\mathbb{P}\left[\text{Vacant} \mid \text{says "Vacant"}\right] = \frac{\pi_{VV}}{\pi_{VV} + \pi_{OOV}} = \frac{1/4}{1/4 + 1/12} = \frac{3}{4}$$

$$\mathbb{P}\left[\text{Occupied} \mid \text{says "Occupied"}\right] = \frac{\pi_{OCO} + \pi_{OFO} + \pi_{OOO}}{\pi_{OCO} + \pi_{OFO} + \pi_{OOO} + \pi_{VO}} = \frac{5}{8}.$$

2.8. Example on an Infinite State Space

Consider a random walk restricted to the nonnegative integers. Here $P_{0,1} = 1$ and otherwise, $P_{i,i-1} = p$ with $p > 1/2$ and $P_{i,i+1} = 1 - p < 1/2$. The restriction $p > 1/2$ insures that a stationary distribution exists. The equations for stationarity are

$$\pi_0 = p\pi_1, \quad \pi_1 = \pi_0 + p\pi_2, \quad \pi_i = (1-p)\pi_{i-1} + p\pi_{i+1} \quad i \geq 2.$$

together with the normalization condition $\sum_{\nu=0}^{\infty} \pi_\nu = 1$. The solution is the stationary distribution $\pi_0 = \frac{2p-1}{2p}$ and $\pi_\nu = \frac{2p-1}{2p}\left(\frac{(1-p)^{\nu-1}}{p^\nu}\right)$ for $\nu \geq 1$. (See the exercises.)

This Markov chain can be interpreted as random walk with negative drift and reflecting boundary on the nonnegative integers. It can also be interpreted as the embedded discrete-time chain for an M/M/1 queue in which $p = \frac{s}{r+s}$ where r is the arrival rate of customers and s is the service rate.

2.9. Chapter Ending Answer

Because the urns are labeled arbitrarily, there is no reason to believe that the fraction of time urn A is empty depends how the balls are initially distributed between the two urns. The randomizing of urn selection should also quickly remove any initial information, so again there is no reason to believe that the fraction of time urn A is empty depends how the balls are initially distributed. Without the additional theory provided in this chapter, one way to determine the fraction of time urn A is empty would be simulation. The theory in this chapter provides an easy way to determine that fraction.

2.10. Chapter Summary

Key Concepts.

(1) The states of a Markov chain partition into equivalence classes according to their type: communicating, periodic versus aperiodic, recurrent versus transient, or positive recurrent versus null recurrent.

(2) Every irreducible, aperiodic Markov chain has a unique nonnegative solution of

$$\sum_i \pi_i P_{ij} = \pi_j$$

$$\sum_i \pi_i = 1$$

called a stationary distribution.

(3) For an irreducible and aperiodic Markov chain,

(a)

$$\lim_{n \to \infty} (P^n)_{ii} = \frac{1}{m_i} = \pi_{ii}$$

and

(b)

$$\lim_{n \to \infty} (P^n)_{ij} = \lim_{n \to \infty} (P^n)_{ii} = \pi_{ij}.$$

Vocabulary.

(1) State j is **accessible** from state i if $(P^n)_{ij} > 0$ for some $n \geq 1$.

(2) The states of a Markov chain partition into equivalence classes according to their type: **communicating**, **periodic** versus **aperiodic**, and **recurrent** versus **transient**, **positive recurrent** versus **null recurrent**.

(3) The Markov chain is **irreducible** if it has only one communicating class.

(4) Let the random variable τ_{ii} be the **return time** to state i given the chain starts in i:

$$\tau_{ii} = \min \{n \geq 1 : X_n = i \mid X_0 = i\},$$
$$\tau_{ii} = \infty \text{ if } X_n \neq i, n \geq 1.$$

The distribution of τ_{ii} is f_{ii}^n. The **expected return time** m_i is $\sum_{\nu=1}^{\infty} \nu f_{ii}^{\nu}$.

(5) Let the random variable τ_{ij} be the **first passage time** to state j given the chain starts in i:

$$\tau_{ij} = \min \{n \geq 1 : X_n = j \mid X_0 = i\}$$
$$\tau_{ij} = \infty \text{ if } X_n \neq j, n \geq 1.$$

(6) If a state i is recurrent, then it is **positive recurrent** if, starting in i, the expected time until the process returns to state i is finite. Otherwise the state is **null recurrent**.

(7) Positive recurrent and aperiodic states are **ergodic**.

(8) When the limit exists, let

$$\pi_j = \lim_{n \to \infty} \frac{1}{n} \sum_{\nu=1}^{n} \mathbf{1}_{[X_\nu = j \mid X_0 = i]}$$

denote the **long run proportion of time the chain spends in state j** or just the *long run proportion* for short.

(9) For a positive recurrent and aperiodic Markov chain the vector (π_j) satisfies the set of the equations

$$\sum_i \pi_i P_{ij} = \pi_j$$

$$\sum_i \pi_i = 1$$

and is a **stationary probability distribution** of the Markov chain.

Notation.

(1) i, j, l – arbitrary or generic state indices

(2) $(P^n)_{ij}$ – the i, j entry of the nth power of P.

(3) $d(i)$ – the period of a state i

(4) τ_{ii} – the **return time** to state i given the chain starts in i

(5) τ_{ij} – the time to state j given the chain starts in i

(6) f_{ii}^n – the probability that starting from i, the first return to state i occurs at step n

(7) $f_i = \mathbb{P}\left[\tau_{ii} < \infty\right]$ – the probability that starting in i the process will *ever* reenter state i

(8)

$$\pi_j = \lim_{n \to \infty} \frac{1}{n} \sum_{\nu=1}^{n} \mathbf{1}_{[X_\nu = j | X_0 = i]}$$

– the **long run proportion of time the chain spends in state** j

(9) m_i – the **expected return time** $\sum_{\nu=1}^{\infty} \nu f_{ii}^\nu$.

2.11. Sources

This section is adapted from: Ross, *Introduction to Probability Models*, Taylor and Karlin, *An Introduction to Stochastic Modeling*, Karlin and Taylor, *A Second Course in Stochastic Processes*. The bathroom example is adapted from the solution in Lessard based on the problem originally posed in The Riddler. and Lessard, Lessard. The example of random walk on the non-negative integers is adapted from class notes by Karl Sigman, Sigman.

2.12. Links

(1) http://www.columbia.edu/ ks20/stochastic-I/stochastic-I-MCII.pdf

2.13. Algorithms and Scripts

Algorithm.

> **Data:** State names and probability transition matrix
> **Result:** Information about a simple Markov chain
>
> **1** *Initialization and sample paths*
> **2** Load Markov chain library
> **3** Set state names, set transition probability matrix, set start state
> **4** Set an example length and create a sample path of example length
> **5** *Simulation of stationary distribution and comparison to theoretical*
> **6** Set a long path length, and a transient time
> **7** Create a long sample path
> **8** Slice the long sample path from the transient time to the end
> **9** In the slice count the appearance of each state
> **10** Store in an empirical array
> **11** Compute the theoretical stable array
> **12** **return** *Stable distribution and theoretical stable distribution*
>   **Algorithm 2:** Markov chain simulation of bathroom example.

Scripts.

```
library("markovchain")

rows = 6

stateNames <- c("OCO","OFO","OOO", "OOV", "VO", "VV")
P <- matrix(c(1/2,   0,   0,   0,    0, 1/2,
                0, 1/2,   0,   0, 1/4, 1/4,
                0,   0, 1/2, 0, 1/2,   0,
                0,   0, 0,   1/2, 1/2, 0,
              1/6, 1/6, 1/6, 0, 1/2, 0,
              1/6, 1/6,   0, 1/6, 0, 1/2),
              nrow=rows, byrow=TRUE)
rownames(P) <- stateNames
colnames(P) <- stateNames

startState  <- "VV"

bathroom  <- new("markovchain", transitionMatrix=P,
                 states=stateNames, name="Bathroom_Occupancy")
print( summary(bathroom) )

pathLength  <- 20
bathroomHistory  <- rmarkovchain(n=pathLength, object=bathroom,
                                 t0=startState)
cat("A sample history, starting from VV: ", bathroomHistory, "\n")

largeN  <- 4000
startStable <- 2000
history  <- rmarkovchain(n=largeN, object=bathroom, t0=startState)
stablePattern  <- history[startStable:largeN]
empiricalStable <- c(mean( stablePattern == "OCO" ),
                     mean( stablePattern == "OFO" ),
                     mean( stablePattern == "OOO" ),
                     mean( stablePattern == "OOV" ),
                     mean( stablePattern == "VO"  ),
                     mean( stablePattern == "VV"  )
                    )
theorStable <- steadyStates(bathroom)
cat("Empirical stable distribution: ",  empiricalStable, "\n")
cat("Theoretical stable distribution: ", theorStable, "\n")
```

.

2.14. Problems to Work for Understanding

1: Provide examples of the classifications of states:

  (a) A trivial two-state Markov chain in which neither state is accessible from the other.

  (b) A trivial two-state Markov chain in which both states communicate.

  (c) A Markov chain in which all states have period 2.

  (d) A three-state Markov chain with two states absorbing and one transient state. What states are communicating in this example?

  (e) An irreducible, positive recurrent, and aperiodic three-state Markov chain.

2: Randomly distribute three balls between two urns, labeled A and B. Each period, select an urn at random, and if it is not empty, remove a ball from that urn and put it in the other urn. If the urn is empty, go to the next period. Make a Markov chain model of this situation and classify all states. In the long run what fraction of time is urn A empty? Does this depend on how the balls are initially distributed between the two urns?

3: A professor has n umbrellas, of which initially $k \in (0, n)$ are at home and $n - k$ are at work. Every day, the professor walks to work in the morning and returns home in the evening. In each trip, the professor takes an umbrella only if it is raining. Assume that in every trip between home and work or back, the chance of rain is $p = 1/2$, independently of other trips.

  (1) Model this process as a Markov chain.

  (2) Determine the stationary distribution.

  (3) Asymptotically, in what fraction of trips does the professor get wet?

Remark 2.47. This is a standard problem appearing in many texts on Markov chains in various degrees of generality.

4: Assume the fraction of oblivious, forgetful, and conscientious users are p, q, and r respectively, with $0 \le p, q, r \le 1$ and $p + q + r = 1$. Solve the bathroom problem under this general assumption.

5: Assume the fraction of oblivious, forgetful, and conscientious users are p, q, and r respectively, with $0 \le p, q, r \le 1$ and $p + q + r = 1$. Further assume the bathroom is occupied $2/3$ of the time, and vacant $1/3$ of the time. Solve the bathroom problem under this general assumption.

6: Assume the fraction of oblivious, forgetful, and conscientious users are p, q, and r respectively, with $0 \le p, q, r \le 1$ and $p + q + r = 1$. Further assume the bathroom is occupied $1 - v$ of the time, and vacant v of the time, where $0 < v < 1$. Solve the bathroom problem under this general assumption. What happens as $v \to 0$ or $v \to 1$?

7:    Assume the fraction of oblivious, forgetful, and conscientious users are p, q, and r respectively, with $0 \leq p, q, r \leq 1$ and $p + q + r = 1$. Also assume that the oblivious people spend twice as long in the bathroom as the conscientious or forgetful people. Assume the bathroom is vacant v of the time, where $0 < v < 1$. Find the transition probability matrix. What is the condition on v to make the probabilities consistent?

8:   Prove Corollary 2.22:

(1) For any communication class C of a Markov chain, all states in C are either recurrent or all states in C are transient. That is
   - if i and j communicate, and i is recurrent, then so is j
   - if i and j communicate, and i is transient, then so is j.

(2) For an irreducible Markov chain, either all states are recurrent or all states are transient.

9:   Prove $f_{ij}^n = \sum_{\nu=0}^{n} f_{ij}^{\nu} P_{jj}^{\nu-k}$.

10:   If $a_n \to a$ as $n \to \infty$, then

$$\lim_{n \to \infty} \frac{1}{n} \sum_{\nu=1}^{n} a_\nu = a,$$

so that if a_n converges to a, the sequence of averages converges to a.

11:   For the random walk restricted to the nonnegative integers with $P_{0,1} = 1$ and otherwise, $P_{i,i-1} = p$ with $p > 1/2$ and $P_{i,i+1} = 1 - p < 1/2$, show that the stationary distribution is $\pi_0 = \frac{2p-1}{2p}$ and $\pi_\nu = \frac{2p-1}{2p} \left(\frac{(1-p)^{\nu-1}}{p^\nu} \right)$ for $\nu \geq 1$.

Waiting Time to Absorption

Let $\{X_n\}$ be a finite-state absorbing Markov chain with a absorbing states and t transient states, where $a + t = k$. Starting at one of the transient states i where $a+1 \le i \le a+t$ such a process will remain in the transient states for some duration. Ultimately, the process gets trapped in one of the absorbing states $i = 1, \dots a$. Before the Markov chain transitions to one of the absorbing states, the number of times it visits a transient state is a random variable. Let Y_{ij} denote the number of visits the system makes to transient state j before reaching an absorbing state, given the system started in transient state i. Thus, Y_{ij} is a discrete random variable that can take on any nonnegative integer value. The random variables Y_{ij} are the fundamental random variables of interest here. These fundamental random variables are the building blocks for constructing and investigating other random variables. The mean, variance and covariances of the Y_{ij} are the first statistics to investigate. Of special interest is the mean time until absorption.

Mathematically Mature: may contain mathematics beyond calculus with proofs.

3.1. Chapter Starter Question

For a Markov chain with an absorbing state, describe the random variable for the time until the chain gets absorbed.

3.2. Theory

Let $\{X_n\}$ be a finite-state absorbing Markov chain with a absorbing states and t transient states, where $a + t = k$. Let the $(a + t) \times (a + t)$ transition probability matrix be P. Order the states so the absorbing states come first and non-absorbing, i.e. transient, states come last. The states $a+1, a+2, \dots, a+t$ are transient in that $(P^n)_{ij} \to 0$ as $n \to \infty$ for $a + 1 \le i, j \le a + t$, while states $1, \dots, a$ are absorbing, $P_{ii} = 1$ for $1 \le i \le a$. Then the transition probability matrix has the block-matrix

form

$$P = \begin{pmatrix} I_a & 0 \\ A & T \end{pmatrix}.$$

Here I_a is an $a \times a$ identity matrix, A is the $t \times a$ matrix of single-step transition probabilities from the t transient states to the a absorbing states, T is a $t \times t$ submatrix of single-step transition probabilities among the transient states, and 0 is a $a \times t$ matrix of 0s representing the single-step transition probabilities from absorbing states to transient states.

Starting at one of the transient states i where $a + 1 \leq i \leq a + t$ such a process will remain in the transient states for some duration. Ultimately, the process gets trapped in one of the absorbing states $i = 1, \ldots a$. Before the Markov chain transitions to one of the absorbing states, the number of times it visits a transient state is a random variable. Let Y_{ij} denote the number of visits the system makes to transient state j before reaching an absorbing state, given the system started in transient state i. Thus, Y_{ij} is a discrete random variable that can take on any nonnegative integer value. The random variables Y_{ij} are the fundamental random variables of interest here. These fundamental random variables are the building blocks for constructing and investigating other random variables. The mean, variance and covariances of the Y_{ij} are the first statistics to investigate. Of special interest is the mean time until absorption. Define the **absorption time**

$$w_i = \min \{n \geq 0 : X_n \leq a \mid X_0 = i\}.$$

Notice $w_i = \sum_{j=a+1}^{a+t} Y_{ij}$, the total number of visits the process makes among the transient states. The expected value of this random time, $\mathbb{E}[w_i]$ for $i = a+1, \ldots, a+t$, is a first measure of the random variable w_i. Also of interest is the probability distribution of the states into which absorption takes place. Using the fundamental random variables, it is possible to compute this probability too.

3.3. Indicator Bernoulli Random Variables

Let the indicator random variables be

$$U_{ij}^{(m)} = \begin{cases} 1 & \text{if the Markov chain is in transient state} j \\ & \text{after } m \text{ steps given that it starts in transient state } i \\ 0 & \text{if the Markov chain is } not \text{ in transient state } j \\ & \text{after } m \text{ steps given that it starts in transient state } i \end{cases}$$

for $m = 0, 1, 2, \ldots$. The case $m = 0$ simply indicates where the system starts, with the usual notation, $U_{ij}^{(0)}$ is the Kronecker delta function, δ_{ij}. The indicator random variables connect to the fundamental random variables Y_{ij} through the sum

$$Y_{ij} = \sum_{m=0}^{\infty} U_{ij}^{(m)}.$$

3.4. Expected Number of Visits Between States

The expected number of visits to transient state j given the Markov chain starts in transient state i in terms of the indicator random variable is

$$\mathbb{E}\left[Y_{ij}\right] = \mathbb{E}\left[\sum_{m=0}^{\infty} U_{ij}^{(m)}\right] = \sum_{m=0}^{\infty} \mathbb{E}\left[U_{ij}^{(m)}\right].$$

Use mathematical induction to show

$$P^m = \begin{pmatrix} I_a & 0 \\ A & T \end{pmatrix}^{m-1} \begin{pmatrix} I_a & 0 \\ A & T \end{pmatrix} = \begin{pmatrix} I_a & 0 \\ (I_t + T + T^2 + \cdots + T^{m-1})A & T^m \end{pmatrix}.$$

(See the exercises.) The elements $(P^m)_{ij}$ of P^m are the m-step transition probabilities between all states. Since $\mathbb{E}\left[U_{ij}^{(m)}\right] = (P^m)_{ij}$, so

$$\mathbb{E}\left[Y_{ij}\right] = \sum_{m=0}^{\infty} \mathbb{E}\left[U_{ij}^{(m)}\right] = \sum_{m=0}^{\infty} (P^m)_{ij}.$$

When i and j are transient states, consider the entries in the transient corner matrix T^m, so

$$\mathbb{E}\left[Y_{ij}\right] = \sum_{m=0}^{\infty} (T^m)_{ij} = \left(\sum_{m=0}^{\infty} T^m\right)_{ij}.$$

The basic theory of finite absorbing Markov chains ensures that the induced 2-norm (or operator norm) of T is less than 1 under typical conditions, that is $\|T\| < 1$. (See the exercises.) Therefore $\sum_{m=0}^{\infty} T^m$ converges. Furthermore, it converges to the **fundamental matrix** $N = (I - T)^{-1}$. (See the exercises.) Thus $\mathbb{E}\left[Y_{ij}\right] = N_{ij}$.

3.5. Waiting Time to Absorption

Theorem 3.1. *The entries N_{ij} of the fundamental matrix are the expected number of times the chain started from state i will be in state j before ultimate absorption and the vector of expected waiting times to absorption is $\mathbb{E}\left[\mathbf{w}\right] = N = (I - T)^{-1}\mathbf{1}$.*

Proof. The sum over all states j of the expected number of times that the chain started from state i will be in state j before ultimate absorption is the expected waiting time to absorption. $\square$

3.6. Covariances of Numbers of Visits Between States

Now use the indicator Bernoulli random variables to derive $\text{Cov}\left[Y_{ij}, Y_{il}\right]$ where i, j, l are transient states. Recall

$$\text{Cov}\left[Y_{ij}, Y_{il}\right] = \mathbb{E}\left[Y_{ij} \cdot Y_{il}\right] - \mathbb{E}\left[Y_{ij}\right] \cdot \mathbb{E}\left[Y_{il}\right]$$

and since $\mathbb{E}\left[Y_{ij}\right]$ and $\mathbb{E}\left[Y_{il}\right]$ are known, all that is necessary is $\mathbb{E}\left[Y_{ij} \cdot Y_{il}\right]$. Start with

$$Y_{ij}Y_{il} = \left(\sum_{\nu=0}^{\infty} U_{ij}^{(\nu)}\right)\left(\sum_{\mu=0}^{\infty} U_{il}^{(\mu)}\right) = \sum_{\nu=0}^{\infty}\sum_{\mu=0}^{\infty} U_{ij}^{(\nu)}U_{il}^{(\mu)}$$

so

$$\mathbb{E}\left[Y_{ij}Y_{il}\right] = \mathbb{E}\left[\sum_{\nu=0}^{\infty}\sum_{\mu=0}^{\infty}U_{ij}^{(\nu)}U_{il}^{(\mu)}\right] = \sum_{\nu=0}^{\infty}\sum_{\mu=0}^{\infty}\mathbb{E}\left[U_{ij}^{(\nu)}U_{il}^{(\mu)}\right].$$

Rearrange the double sum into 3 double sums, summing first in μ over the lattice points above the line $\mu = \nu$, a second term summing over the lattice points along the line $\mu = \nu$, and last summing first in ν over lattice points below the line $\mu = \nu$,

$$\sum_{\nu=0}^{\infty}\sum_{\mu=\nu+1}^{\infty}\mathbb{E}\left[U_{ij}^{(\nu)}U_{il}^{(\mu)}\right] + \sum_{\nu=0}^{\infty}\mathbb{E}\left[U_{ij}^{(\nu)}U_{il}^{(\nu)}\right] + \sum_{\mu=0}^{\infty}\sum_{\nu=\mu+1}^{\infty}\mathbb{E}\left[U_{il}^{(\mu)}U_{ij}^{(\nu)}\right].$$

The first and third terms are symmetric, so only evaluate the first term.

The expression $\mathbb{E}\left[U_{ij}^{(\nu)}U_{il}^{(\mu)}\right]$ is the probability that the system is in transient state j after exactly ν steps from the start in state i *and* the system is in transient state l after exactly μ steps from the start in state i. In the first double sum $\nu < \mu$. Using the Markov chain property, this is $\mathbb{E}\left[U_{ij}^{(\nu)}U_{il}^{(\mu)}\right] = p_{ij}^{(\nu)}p_{jl}^{(\mu-\nu)}$. Therefore, the first term is

$$\sum_{\nu=0}^{\infty}\sum_{\mu=\nu+1}^{\infty}\mathbb{E}\left[U_{ij}^{(\nu)}U_{il}^{(\mu)}\right] = \sum_{\nu=0}^{\infty}\sum_{\mu=\nu+1}^{\infty}p_{ij}^{(\nu)}p_{jl}^{(\mu-\nu)} = \sum_{\nu=0}^{\infty}\sum_{\mu-\nu=1}^{\infty}p_{ij}^{(\nu)}p_{jl}^{(\mu-\nu)}$$

$$= \sum_{\nu=0}^{\infty}\sum_{z=1}^{\infty}p_{ij}^{(\nu)}p_{jl}^{(z)} = \left(\sum_{\nu=0}^{\infty}p_{ij}^{(\nu)}\right)\left(\sum_{z=1}^{\infty}p_{jl}^{(z)}\right) = \left(\sum_{\nu=0}^{\infty}p_{ij}^{(\nu)}\right)\left(\sum_{z=0}^{\infty}p_{jl}^{(z)} - p_{jl}^{(0)}\right)$$

$$= \left(\sum_{\nu=0}^{\infty}(T^{\nu})_{ij}\right)\left(\sum_{z=0}^{\infty}(T^z)_{jl} - \delta_{jl}\right) = \left(\sum_{\nu=0}^{\infty}T^{\nu}\right)_{ij}\left(\sum_{z=0}^{\infty}T^z - \delta_{jl}\right)_{jl}$$

$$= \left((I_t - T)^{-1}\right)_{ij}\left((I_t - T)^{-1} - \delta_{jl}\right)_{jl}$$

$$= N_{ij}(N_{jl} - \delta_{jl}).$$

The third double sum is the first double sum with j and l interchanged, so

$$\sum_{\mu=0}^{\infty}\sum_{\nu=\mu+1}^{\infty}\mathbb{E}\left[U_{il}^{(\mu)}U_{ij}^{(\nu)}\right] = N_{il}(N_{lj} - \delta_{lj}).$$

Finally, the second term is $\sum_{\nu=0}^{\infty}\mathbb{E}\left[U_{ij}^{(\nu)}U_{il}^{(\nu)}\right]$ where each summand is the probability the Markov chain is in transient state j after exactly ν steps after starting in transient state i and simultaneously in state l after exactly ν steps starting from transient state i. This is only possible if $j = l$ hence $\mathbb{E}\left[U_{ij}^{(\nu)}U_{il}^{(\nu)}\right] = p_{ij}^{(\nu)}\delta_{jl}$. Thus for the second term

$$\sum_{\nu=0}^{\infty}\mathbb{E}\left[U_{ij}^{(\nu)}U_{il}^{(\nu)}\right] = \sum_{\nu=0}^{\infty}p_{ij}^{(\nu)}\delta_{jl} = \left(\sum_{\nu=0}^{\infty}(T^{\nu})_{ij}\right)\delta_{jl} = \left(\sum_{\nu=0}^{\infty}T^{\nu}\right)_{ij}\delta_{jl} = N_{ij}\delta_{jl}.$$

Putting all terms together

$$\mathbb{E}\left[Y_{ij}Y_{il}\right] = N_{ij}(N_{jl} - \delta_{jl}) + N_{ij}\delta_{jl} + N_{il}(N_{lj} - \delta_{lj}) = N_{ij}N_{jl} + N_{il}N_{lj} - N_{il}\delta_{lj}.$$

Then

$$\operatorname{Cov}\left[Y_{ij}, Y_{il}\right] = \mathbb{E}\left[Y_{ij}Y_{il}\right] - \mathbb{E}\left[Y_{ij}\right]\mathbb{E}\left[Y_{il}\right]$$
$$= N_{ij}N_{jl} + N_{il}N_{lj} - N_{ij}N_{il} - N_{il}\delta_{lj}.$$

The result is symmetric under the interchange of j and l as it should be.

In particular, letting $j = l$ gives the variance of the number of visits to state j starting from i:

$$\operatorname{Var}\left[Y_{ij}\right] = 2N_{ij}N_{jj} - N_{ij}^2 - N_{ij}.$$

Define $\operatorname{diag}(N)$ to be the diagonal matrix setting all off-diagonal elements of N to 0, and define N_{sq} to be the matrix resulting from squaring each entry, then

$$\operatorname{Var}\left[Y_{(i)}\right] = N(2\operatorname{diag}(N) - I) - N_{\mathrm{sq}}.$$

Note that this is the variance of the number of visits to transient state j when starting from transient state i, *not* the variance of the waiting time to an absorbing state. The variance of the waiting time until absorption into any state from state i is the sum of all t^2 entries in $\operatorname{Var}\left[Y_{(i)}\right]$.

The covariance, for a fixed i, is the $t \times t$ matrix of entries $\operatorname{Cov}\left[Y_{ij}, Y_{il}\right]$. Letting $N_{(i,\cdot)}$ denote the $1 \times t$ row-vector from row i and setting $(\operatorname{diag} N_{(i,\cdot)})$ to be the diagonal matrix with this vector along the diagonal this becomes

$$\left[\operatorname{Cov}\left[Y_{ij}, Y_{il}\right]\right] = (\operatorname{diag} N_{(i,\cdot)})N + N^T(\operatorname{diag} N_{(i,\cdot)}) - N_{(i,\cdot)}^T N_{(i,\cdot)} - \operatorname{diag} N_{(i,\cdot)}.$$

Note the outer product $N_{(i,\cdot)}^T N_{(i,\cdot)}$. Note also the notation $\left[\operatorname{Cov}\left[Y_{ij}, Y_{il}\right]\right]$ for the matrix with i, j entry $\operatorname{Cov}\left[Y_{ij}, Y_{il}\right]$.

3.7. First-Step Analysis

First-step analysis says the absorption time from state i is the first step to another transient state plus a weighted average, according to the transition probabilities over the transient states, of the absorption times from the other transient states. In symbols, first-step analysis says

$$w_i = 1 + \sum_{j=a+1}^{a+t} P_{ij}w_j.$$

As before, for a Markov chain with a absorbing states and t transient states, re-order the states so the absorbing states come first and non-absorbing, i.e. transient, states come last. Then the transition matrix has the canonical form:

$$P = \begin{pmatrix} I_a & 0 \\ A & T \end{pmatrix}.$$

Here I_a is an $a \times a$ identity matrix, A is the $t \times a$ matrix of single-step transition probabilities from the t transient states to the a absorbing states, T is a $t \times t$ submatrix of single-step transition probabilities among the transient states, and 0 is a $a \times t$ matrix of 0s representing the single-step transition probabilities from absorbing states to transient states.

Expressing the first-step analysis compactly in vector-matrix form as

$$\mathbf{w} = \mathbf{1} + T\mathbf{w}$$

or
$$(I - T)\mathbf{w} = \mathbf{1}.$$

Then $\mathbf{w} = (I - T)^{-1}\mathbf{1}$. The matrix $N = (I - T)^{-1}$ is the **fundamental matrix** for the absorbing Markov chain. The entries N_{ij} of this matrix have a probabilistic interpretation. The entries N_{ij} are the expected number of times the chain started from state i will be in state j before ultimate absorption.

The next theorem shows the $t \times a$ matrix of **absorption probabilities** $B = NA = (I - T)^{-1}A$ has as entries the probability of starting at state i and ending up at a given absorbing state j.

Theorem 3.2. *Let b_{ij} be the probability of the Markov chain starting in transient state i and ending in absorbing state j, then*

$$B = (b_{ij}) = NA.$$

Proof. The proof is by first-step analysis. Starting in state i, the process may be captured in j in one or more steps. The probability of capture in a single step is p_{ij}. If this does not happen, the process may move to another absorbing state, in which case it is impossible to reach j, or to a transient state k. In the latter case, the probability of being captured in j is b_{kj}. Hence

$$b_{ij} = p_{ij} + \sum_{l=a+1}^{a+t} p_{il}b_{lj}$$

or in matrix form $B = A + TB$. Thus, $B = (I - T)^{-1}A = NA$. $\square$

Kemeny and Snell [**KS60**, page 51] derive the variance of the waiting time using first-step analysis.

Theorem 3.3.
$$\mathrm{Var}\,[\mathbf{w}] = (2N - I)N\mathbf{1} - (N\mathbf{1})_{sq}$$

where $(N\mathbf{1})_{sq}$ is the element-wise squared vector.

Proof. (1) Start with $\mathbb{E}\,[\mathbf{w}] = N\mathbf{1}$. Let $\mathbf{w}_{sq}$ be the element-wise squared vector. Then use first-step analysis to evaluate $\mathbb{E}\,[\mathbf{w}_{sq}]$.

(2) From starting state i the chain can go to any state l with probability p_{il}. If the new state is absorbing, then it can never reach another state, and the contribution is 1 step. If the new state is transient count the weighted average of the squares of the waiting times plus 1.

(3) In symbols:

$$\mathbb{E}\,[\mathbf{w}_{sq}] = \sum_{l=1}^{a} p_{il}\mathbf{1} + \sum_{l=a+1}^{a+t} p_{il}(\mathbb{E}\,[(\mathbf{w} + \mathbf{1})_{sq}])_l$$
$$= \sum_{l=1}^{a} p_{il}\mathbf{1} + \sum_{l=a+1}^{a+t} p_{il}\left((\mathbb{E}\,[\mathbf{w}]_{sq})_l + 2(\mathbb{E}\,[\mathbf{w}])_l + \mathbf{1}\right)$$
$$= T\mathbb{E}\,[\mathbf{w}_{sq}] + 2T\mathbb{E}\,[\mathbf{w}] + \mathbf{1}.$$

Figure 1. Image of a possible random walk in phase line after an odd number of steps.

(4) Rearrange using $N = (I_t - T)^{-1}$ and $NT = (I_t - T)^{-1}T = N - I_t$ (See the exercises.)
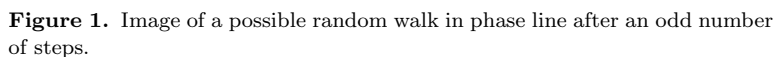
$$\mathbb{E}\left[\mathbf{w}_{sq}\right] = (I_t - T)^{-1}(2T\mathbb{E}\left[\mathbf{w}\right] + \mathbf{1})$$
$$= 2NT\mathbb{E}\left[\mathbf{w}\right] + N\mathbf{1}$$
$$= 2(N - I_t)\mathbb{E}\left[\mathbf{w}\right] + \mathbb{E}\left[\mathbf{w}\right]$$
$$= (2N - I_t)\mathbb{E}\left[\mathbf{w}\right].$$

(5) Thus

$$\mathrm{Var}\left[\mathbf{w}\right] = (2N - I)N\mathbf{1} - (N\mathbf{1})_{\mathrm{sq}}.$$

$\square$

Remark 3.4. Carchidi and Higgins [**CH17**] give an alternate proof and slightly different derivation of this formula for the variance of the waiting time until absorption using equation (3.1).

3.8. Examples

Example 3.5. Consider a random walk of a particle that moves along a straight line in unit steps. Each step is 1 unit to the right with probability p and to the left with probability $q = 1 - p$. It moves until it reaches one of two extreme points which are absorbing boundaries. Assume that if the process reaches the boundary points, it remains there from that time on. Figure 1 has 9 states numbered from -4 to 4. The absorbing boundary states are -4 and 4.

The full transition probability matrix is

$$P = \begin{array}{c} \\ -4 \\ -3 \\ -2 \\ -1 \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccccccc} -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 \end{array} \\ \left(\begin{array}{ccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ q & 0 & p & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & q & 0 & p & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & q & 0 & p & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q & 0 & p & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & q & 0 & p & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & q & 0 & p & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & q & 0 & p \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right) \end{array}.$$

Reorder the states as $-4, 4, -3, -2, -1, 0, 1, 2, 3$ to bring the transition probability matrix to the standard form

$$P_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ q & 0 & 0 & p & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & q & 0 & p & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q & 0 & p & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & q & 0 & p & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & q & 0 & p & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & q & 0 & p \\ 0 & p & 0 & 0 & 0 & 0 & 0 & q & 0 \end{pmatrix}$$

so

$$T = \begin{pmatrix} 0 & p & 0 & 0 & 0 & 0 & 0 \\ q & 0 & p & 0 & 0 & 0 & 0 \\ 0 & q & 0 & p & 0 & 0 & 0 \\ 0 & 0 & q & 0 & p & 0 & 0 \\ 0 & 0 & 0 & q & 0 & p & 0 \\ 0 & 0 & 0 & 0 & q & 0 & p \\ 0 & 0 & 0 & 0 & 0 & q & 0 \end{pmatrix}.$$

A computer algebra system can compute the fundamental matrix $N(p) = (I - T)^{-1}$ but the general form in p and q is long and unhelpful. Two representative numerical examples suffice to show the possibilities.

$$N(1/2) = \begin{pmatrix} 7/4 & 3/2 & 5/4 & 1 & 3/4 & 1/2 & 1/4 \\ 3/2 & 3 & 5/2 & 2 & 3/2 & 1 & 1/2 \\ 5/4 & 5/2 & 15/4 & 3 & 9/4 & 3/2 & 3/4 \\ 1 & 2 & 3 & 4 & 3 & 2 & 1 \\ 3/4 & 3/2 & 9/4 & 3 & 15/4 & 5/2 & 5/4 \\ 1/2 & 1 & 3/2 & 2 & 5/2 & 3 & 3/2 \\ 1/4 & 1/2 & 3/4 & 1 & 5/4 & 3/2 & 7/4 \end{pmatrix}.$$

Then the waiting times to absorption are $7, 12, 15, 16, 15, 12, 7$. For the variances, consider only the central state, originally labeled as 0, after reordering it is the sixth state in the middle of the transient states.

$$[\text{Cov}\,[Y_{0j}, Y_{0l}]] = \begin{pmatrix} 3/2 & 5/2 & 2 & 1 & 0 & -1/2 & -1/2 \\ 5/2 & 6 & 13/2 & 4 & 3/2 & 0 & -1/2 \\ 2 & 13/2 & 21/2 & 9 & 9/2 & 3/2 & 0 \\ 1 & 4 & 9 & 12 & 9 & 4 & 1 \\ 0 & 3/2 & 9/2 & 9 & 21/2 & 13/2 & 2 \\ -1/2 & 0 & 3/2 & 4 & 13/2 & 6 & 5/2 \\ -1/2 & -1/2 & 0 & 1 & 2 & 5/2 & 3/2 \end{pmatrix}.$$

The variance of the number of visits to 0 until absorption is $\mathbf{1}^T V \mathbf{1} = 160$, the standard deviation is $4\sqrt{10}$. The negative covariance $-1/2$ of the number of visits from 0 to -3 until absorption with the number of visits from 0 to 3 until absorption indicates the random number of visits from 0 to -3 until absorption has a somewhat opposite distribution to that of the random number of visits from 0 to 3 until absorption. That is, when the number of visits from 0 to -3 until absorption is

large, the number of visits from 0 to 3 until absorption will be small. This makes sense, since if the number of visits from 0 to -3 is large, the likelihood of absorption into -4 increases, and the number of visits from 0 to 3 decreases.

The probabilities of absorption are

$$N(1/2)A = \begin{pmatrix} 7/8 & 1/8 \\ 3/4 & 1/4 \\ 5/8 & 3/8 \\ 1/2 & 1/2 \\ 3/8 & 5/8 \\ 1/4 & 3/4 \\ 1/8 & 7/8 \end{pmatrix}$$

and are symmetric as expected.

If $p = 2/3$ so the probability of moving to the right is twice the probability of moving to the left, then

$$N(2/3) = \begin{pmatrix} 127/85 & 126/85 & 124/85 & 24/17 & 112/85 & 96/85 & 64/85 \\ 63/85 & 189/85 & 186/85 & 36/17 & 168/85 & 144/85 & 96/85 \\ 31/85 & 93/85 & 217/85 & 42/17 & 196/85 & 168/85 & 112/85 \\ 3/17 & 9/17 & 21/17 & 45/17 & 42/17 & 36/17 & 24/17 \\ 7/85 & 21/85 & 49/85 & 21/17 & 217/85 & 186/85 & 124/85 \\ 3/85 & 9/85 & 21/85 & 9/17 & 93/85 & 189/85 & 126/85 \\ 1/85 & 3/85 & 7/85 & 3/17 & 31/85 & 63/85 & 127/85 \end{pmatrix}$$

and the waiting times to absorption are approximately 9.05, 12.07, 12.08, 10.59, 8.34, 5.72, 2.91. For the variances, consider only the central state, originally labeled as 0 and after reordering is the sixth state in the middle of the transient states.

$$[\text{Cov}\,[Y_{0j}, Y_{0k}]] = \begin{pmatrix} \frac{462}{1445} & \frac{162}{289} & \frac{708}{1445} & \frac{72}{289} & 0 & -\frac{144}{1445} & -\frac{144}{1445} \\ \frac{162}{289} & \frac{2232}{2682} & \frac{2682}{1445} & \frac{324}{289} & \frac{504}{1445} & 0 & -\frac{144}{1445} \\ \frac{708}{1445} & \frac{2682}{1445} & \frac{5124}{1445} & \frac{882}{289} & \frac{1764}{1445} & \frac{504}{1445} & 0 \\ \frac{72}{289} & \frac{324}{289} & \frac{882}{289} & \frac{1260}{289} & \frac{882}{289} & \frac{324}{289} & \frac{72}{289} \\ 0 & \frac{504}{1445} & \frac{1764}{1445} & \frac{882}{289} & \frac{5838}{1445} & \frac{720}{289} & \frac{912}{1445} \\ -\frac{144}{1445} & 0 & \frac{504}{1445} & \frac{324}{289} & \frac{720}{289} & \frac{4068}{1445} & \frac{1728}{1445} \\ -\frac{144}{1445} & -\frac{144}{1445} & 0 & \frac{72}{289} & \frac{912}{1445} & \frac{1728}{1445} & \frac{1176}{1445} \end{pmatrix}.$$

The variance is $\mathbf{1}^T V \mathbf{1} = 15264/289 \approx 52.817$, the standard deviation is 7.268, less than the symmetric case where $p = 1/2 = q$, as expected.

The probabilities of absorption are

$$N(2/3)A = \begin{pmatrix} \frac{127}{255} & \frac{128}{255} \\ \frac{21}{85} & \frac{64}{85} \\ \frac{31}{255} & \frac{224}{255} \\ \frac{1}{17} & \frac{16}{17} \\ \frac{7}{255} & \frac{248}{255} \\ \frac{1}{85} & \frac{84}{85} \\ \frac{1}{255} & \frac{254}{255} \end{pmatrix}.$$

The absorption probabilities are strongly biased to the right, as expected. $\qquad \triangledown$

Example 3.6. An urn contains two unpainted balls. At a sequence of times, choose a ball at random, then paint it either red or black, and put it back. For an unpainted ball, choose a color at random. For a painted ball, change its color. Form a Markov chain by taking as a state the triple (x, y, z) where x is the number of unpainted balls, y the number of red balls, and z the number of black balls. The transition matrix is then

$$
\begin{array}{c}
\\
(0,1,1)\\
(0,2,0)\\
(0,0,2)\\
(2,0,0)\\
(1,1,0)\\
(1,0,1)
\end{array}
\begin{array}{c}
(0,1,1)\ \ (0,2,0)\ \ (0,0,2)\ \ (2,0,0)\ \ (1,1,0)\ \ (1,0,1)\\
\left(\begin{array}{cccccc}
0 & 1/2 & 1/2 & 0 & 0 & 0\\
1 & 0 & 0 & 0 & 0 & 0\\
1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1/2 & 1/2\\
1/4 & 1/4 & 0 & 0 & 0 & 1/2\\
1/4 & 0 & 1/4 & 0 & 1/2 & 0
\end{array}\right)
\end{array}.
$$

In this case, the system has no absorbing state, that is, a state that once entered remains the same thereafter. However, the first three states together are an irreducible set, once the process enters that set, it continues in that set. So lump those states together as a single absorbing state, and the transient states are $(2,0,0)$, $(1,1,0)$, and $(1,0,1)$. Then

$$
T = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 \\ 0 & 1/2 & 0 \end{pmatrix}
$$

and the fundamental matrix is

$$
N = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 4/3 & 2/3 \\ 0 & 2/3 & 4/3 \end{pmatrix}.
$$

With this,

$$
\mathbb{E}\left[\mathbf{w}\right] = N\mathbf{1} = (3, 2, 2)^T
$$

and

$$
\operatorname{Var}\left[\mathbf{w}\right] = (2N - I)\mathbb{E}\left[\mathbf{w}\right] - (\mathbb{E}\left[\mathbf{w}\right])_{\text{sq}} = (2, 2, 2)^T.
$$

Finally,

$$
\operatorname{Var}\left[\mathbf{X}\right] = N(2\operatorname{diag}(N) - I) - N_{\text{sq}} = \begin{pmatrix} 0 & 2/3 & 2/3 \\ 0 & 4/9 & 2/3 \\ 0 & 2/3 & 4/9 \end{pmatrix}.
$$

Since the process must immediately leave state $(2,0,0)$ and cannot go back, the variance is 0 for the number of times in this state. $\triangledown$

Example 3.7. The following is a larger example of the painting the balls puzzle.

You play a game with four balls in a box: One ball is red, one is blue, one is green and one is yellow. You draw a ball out of the box at random and note its color. Without replacing the first ball, you draw a second ball and then paint it to match the color of the first. Replace both balls, and repeat the process. The game ends when all four balls have become the same color. What is the expected number of turns to finish the game?

Take the states as the number of balls of each different color without regard for the colors themselves. For example, partition 4 balls in 5 different ways:

$$1 + 1 + 1 + 1, \quad 2 + 1 + 1, \quad 2 + 2, \quad 3 + 1, \quad 4.$$

The partition $2 + 1 + 1$, for example, consists of cases where two of the balls are the same color and the other two balls are two other colors. For example, the cases "red & red & green & blue" and "blue & blue & yellow & red' correspond to the partition $2 + 1 + 1$. Using these five partitions as states in a Markov Chain, compute the transition probabilities to go from one state to the next. For example, the probability of transition from $2 + 1 + 1$ to $3 + 1$ is $1/3$ because in order for this transition to occur, we must first choose one of the two identically colored balls with probability $2/4$, then we must choose one of the other two balls out of the remaining three with probability $2/3$. The joint probability is $2/4 \cdot 2/3 = 1/3$. As another example, the probability of transition from $2 + 2$ to $3 + 1$ is the probability of first picking a ball of either color, leaving 1 ball of that color and 2 balls of the other color, then from those 3 balls picking the second color with probability $2/3$. Calculate all remaining transition probabilities in the same way.

$$
P = \begin{array}{c}
\\
1+1+1+1 \\
2+1+1 \\
3+1 \\
2+2 \\
4
\end{array}
\begin{array}{c}
1+1+1+1 \quad 2+1+1 \quad 3+1 \quad 2+2 \quad 4 \\
\left(\begin{array}{ccccc}
0 & 1 & 0 & 0 & 0 \\
0 & 1/2 & 1/3 & 1/6 & 0 \\
0 & 0 & 1/2 & 1/4 & 1/4 \\
0 & 0 & 2/3 & 1/3 & 0 \\
0 & 0 & 0 & 0 & 1
\end{array} \right)
\end{array}.
$$

The absorbing state is 4. Rearranging into the canonical block-matrix form

$$
P = \begin{array}{c}
\\
4 \\
1+1+1+1 \\
2+1+1 \\
3+1 \\
2+2
\end{array}
\begin{array}{c}
4 \quad 1+1+1+1 \quad 2+1+1 \quad 3+1 \quad 2+2 \\
\left(\begin{array}{ccccc}
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1/2 & 1/3 & 1/6 \\
1/4 & 0 & 0 & 1/2 & 1/4 \\
0 & 0 & 0 & 2/3 & 1/3
\end{array} \right)
\end{array}.
$$

Then $N = (I - T)^{-1}$ is

$$
\begin{pmatrix}
1 & 2 & 4 & 2 \\
0 & 2 & 4 & 2 \\
0 & 0 & 4 & 3/2 \\
0 & 0 & 4 & 3
\end{pmatrix}
$$

and the waiting time from the initial state is $1 + 2 + 4 + 2 = 9$. From the state $1 + 1 + 1 + 1$ (labeled as state 2 in the canonical format), the covariance matrix is

$$
\begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 \\
0 & 0 & 12 & 6 \\
0 & 0 & 6 & 6
\end{pmatrix}.
$$

Then the variance of the number of visits is 32 and the standard deviation of the number of visits is $4\sqrt{2} \approx 5.657$.

$\triangledown$

Example 3.8. It's your 30th birthday, and your friends bought you a cake with 30 candles on it. You make a wish and try to blow them out. Every time you blow, you blow out a random number of candles between one and the number that remain, including one and that other number. How many times on average do you blow before you extinguish all the candles?

Let the states be the number of candles remaining lit. Order the $N = 31$ states as $0, 1, 2, 3, \ldots, 29, 30$. Then the state 0 is absorbing and all other states are transient. Interpret "blow out a random number of candles between one and the number that remain" as uniform distribution on the number of candles remaining. Instead of solving this full problem at once, try a smaller problem first, with the number of candles $N = 5$. Then the transition probability matrix in canonical form is

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 & 0 \end{pmatrix}.$$

Then the first-step equations for the waiting time, that is the number of attempts needed to blow out the candles are

$$w_1 = 1$$

$$w_2 = 1 + \frac{1}{2}w_1$$

$$w_3 = 1 + \frac{1}{3}w_1 + \frac{1}{3}w_2$$

$$w_4 = 1 + \frac{1}{4}w_1 + \frac{1}{4}w_2 + \frac{1}{4}w_3$$

$$w_5 = 1 + \frac{1}{5}w_1 + \frac{1}{5}w_2 + \frac{1}{5}w_3 + \frac{1}{5}w_4$$

Solving this recursively, $w_1 = 1$, $w_2 = 1 + \frac{1}{2}$, $w_3 = 1 + \frac{1}{3} + \frac{1}{3} \cdot (1 + \frac{1}{2})$, so $w_3 = 1 + \frac{1}{2} + \frac{1}{3}$. Continuing to solve recursively

$$w_4 = 1 + 1/2 + 1/3 + 1/4$$
$$w_5 = 1 + 1/2 + 1/3 + 1/4 + 1/5.$$

The inductive pattern is clear. The waiting time with N candles is the Nth **harmonic number** H_N

$$w_N = H_N = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots + \frac{1}{N}.$$

For the original problem with 30 candles the transient states are $1, 2, 3, \ldots, 30$ and the absorbing state is 0. Then the canonical form transition probability matrix

is

$$P = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 1/2 & 1/2 & 0 & \cdots & 0 & 0 \\ 1/3 & 1/3 & 1/3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1/30 & 1/30 & 1/30 & \cdots & 1/30 & 0 \end{pmatrix}.$$

The quantity of interest is the expected waiting time until absorption into the state 0. Expressing the first-step analysis compactly in vector-matrix form as

$$(I - T)\mathbf{w} = \mathbf{1},$$

substituting in the values from the transition matrix and solving with a computer, this is $w_{30} = H_{30} \approx 3.995$. By calculating the covariance matrix and taking the sum of all entries, the variance of the waiting time from the transient state of all candles lit is approximately 2.383. $\triangledown$

Example 3.9. The following example is from the December 22, 2017 Riddler at Fivethirtyeight.com. It concerns a game of chance called Left, Right, Center. In this game, players sit in a circle and starts with some number of \$1 bills. Players take turns, in order around the circle, rolling three dice. For each die, if it comes up 1 or 2, the player gives a dollar to the person to the left. If it comes up 3 or 4, the player gives a dollar to the person on the right. And if it comes up 5 or 6, the player puts a dollar in the center. Assume the following: First, if a player has no dollars, then her turn is skipped. Second, if a player has one or two dollars, then the player rolls only one or two dice, respectively. The game ends as soon as only a single person has any money left. How long is the game expected to last for six players each starting with three \$1 bills? For X players each starting with Y \$1 bills?

Consider the allocation of the total of \$18 among the 6 players and the center, that is 7 places, as the states of a Markov chain. Using the "stars and bars" argument, choosing $7 - 1 = 6$ bars from $18 + 6$ objects the number of states is $\binom{18+6}{6} = 134{,}596$. This also counts all the money in the center as one state which would not strictly be a state of the game. This makes 134,595 actual game states. This is the same as the number of states dividing any number of dollars from 1 to 18 among 6 players, ignoring the money in the center which would be $\sum_{j=6}^{23} \binom{j}{5} = 134{,}595$. This is essentially the conclusion of the hockey-stick identity for binomial coefficients.

The game ends when exactly 1 of the 6 players has any amount of money from \$1 to \$18. That is, there are $6 \cdot (1 + \cdots + 18) = 6 \cdot 18 \cdot 19/2 = 1026$ terminal states that can be considered as the absorbing states of the game.

The shortest possible game would be for the first 5 players to throw all 5's and 6's, leaving the sixth player with the original stake of money. This shortest game would be 5 turns with probability $(1/27)^5$.

Dollars	**6**	**7**	**8**	**9**
3	25.7096	32.3160	38.9660	45.3878
4	31.7890	39.7598	47.3006	55.3804
5	38.3244	47.4416	56.3536	65.4438
6	44.8394	54.9732	65.5688	75.3548

Table 1. Mean number of turns in 5000 simulations of the game with X players (columns) each with Y dollars (rows).

Dollars	**6**	**7**	**8**	**9**
3	6.546804	7.009249	7.564145	7.898221
4	7.751729	8.176348	8.594918	9.099816
5	8.989401	9.367077	9.878952	10.387840
6	10.16180	10.468601	11.080443	11.654668

Table 2. Variance of number of turns in 5000 simulations of the game with X players each with Y dollars.

Next consider the total number of dollars in the circle at each turn. The expected loss of dollars from the circle to the center at each turn is

$$0 \cdot \left(\frac{2}{3}\right)^3 + 1 \cdot \left(3 \cdot \left(\frac{2}{3}\right)^2 \cdot \left(\frac{1}{3}\right)\right) + 2 \cdot \left(3 \cdot \left(\frac{2}{3}\right) \cdot \left(\frac{1}{3}\right)^2\right) + 3 \cdot \left(\frac{1}{3}\right)^3 = 1.$$

That means it takes should take about 17 turns to end the game. More precisely, consider the lumped system where each state is the number of dollars in the circle, from \$18 to \$1. Let D_j be the duration of the game from j dollars with $j = 1, \ldots, 18$. Then a first-step analysis gives the system of equations $D_j = (8/27)D_j + (12/27)D_{j-1} + (6/27)D_{j-2} + (1/27)D_{j-3} + 1$, with $D_j = 0$ for $j = 1, 0, -1$. The solution of this system with Octave gives $D_{18} = 17.3333$ (see the scripts for the solution method) which is consistent with the previous crude estimate.

The number of states and absorbing states are too large to be effectively handled by matrix methods. Although the conceptual set-up of the game is clearly the waiting time until absorption in a Markov chain, simulation seems to be the best way to answer the question. Results of simulating games with 6 to 9 players each starting with 3 to 6 dollars are in Table 1 and Table 2. The Riddler says the game with X players each starting with Y dollars will last $2(X - 2)Y$ turns but does not give a proof. The simulations give means that are roughly consistent with this value.

$\triangledown$

3.9. Chapter Ending Answer

The number of times the Markov chain visits a transient state is a random variable. Let Y_{ij} denote the number of visits the system makes to transient state j before reaching an absorbing state, given the system started in transient state i. Thus, Y_{ij} is a discrete random variable that can take on any nonnegative integer value.

3.10. Chapter Summary

3.11. Chapter Summary

Key Concepts.

(1) Let $\{X_n\}$ be a finite-state absorbing Markov chain with a absorbing states and t transient states. Let the $(a + t) \times (a + t)$ transition probability matrix be P. Order the states so the absorbing states come first and non-absorbing, i.e. transient, states come last. Then the transition probability matrix has the block-matrix form

$$P = \begin{pmatrix} I_a & 0 \\ A & T \end{pmatrix}.$$

Here I_a is an $a \times a$ identity matrix, A is the $t \times a$ matrix of single-step transition probabilities from the t transient states to the a absorbing states, T is a $t \times t$ submatrix of single-step transition probabilities among the transient states, and 0 is a $a \times t$ matrix of 0s representing the single-step transition probabilities from absorbing states to transient states.

(2) The matrix $N = (I - T)^{-1}$ is the **fundamental matrix** for the absorbing Markov chain. The entries N_{ij} of this matrix have a probabilistic interpretation. The entries N_{ij} are the expected number of times the chain started from state i will be in state j before ultimate absorption.

(3) First-step analysis gives a compact expression in vector-matrix form for the waiting time $\mathbf{w}$ to absorption:

$$(I - T)\mathbf{w} = \mathbf{1}$$

so $\mathbf{w} = (I - T)^{-1}\mathbf{1}$.

Vocabulary.

(1) Let $\{X_n\}$ be a finite-state absorbing Markov chain with a absorbing states and t transient states. Let the $(a + t) \times (a + t)$ transition probability matrix be P. Order the states so the absorbing states come first and non-absorbing, i.e. transient, states come last. Then $(P^n)_{ij} \to 0$ as $n \to \infty$ for i and j in the transient states, while for i in the absorbing states, $P_{ii} = 1$. Define the **absorption time** as the random variable

$$T = \min\{n \geq 0 : X_n \leq a\}.$$

(2) The **absorption probability matrix** B is the probability of starting at state i and ending at absorbing state j.

(3) For a Markov chain with a absorbing states and t transient states, if necessary, reorder the states so the absorbing states come first and non-absorbing, i.e. transient, states come last. Then the transition matrix has the canonical form:

$$P = \begin{pmatrix} I_a & 0 \\ A & T \end{pmatrix}.$$

The matrix $N = (I - T)^{-1}$ is the **fundamental matrix** for the absorbing Markov chain.

(4) The Nth **harmonic number** H_N is

$$H_N = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots + \frac{1}{N}.$$

Notation.

(1) $\{X_n\}$ – a finite-state absorbing Markov chain with a absorbing states and t transient states.

(2) P – the $(a + t) \times (a + t)$ transition probability matrix.

(3) I_a is an $a \times a$ identity matrix, A is the $t \times a$ matrix of single-step transition probabilities from the t transient states to the a absorbing states, T is a $t \times t$ submatrix of single-step transition probabilities among the transient states, and 0 is a $a \times t$ matrix of 0s representing the single-step transition probabilities from absorbing states to transient states.

(4) i, j, l – arbitrary states of the Markov chain

(5) Y_{ij} – the number of visits the system makes to transient state j before reaching an absorbing state, given the system started in transient state i.

(6) w_i – mean time until absorption for transient state i

(7) $\mathbf{w}$ – vector of waiting times until absorption for transient states

(8) $U_{ij}^{(m)}$ – indicator random variable if the Markov chain is in transient state j after m steps given that it starts in transient state i

(9) δ_{ij} – Kronecker delta function

(10) $N = (I - T)^{-1}$ – fundamental matrix

(11) $\mathbf{1}$ – vector with all entries 1

(12) D_j – Duration of the Left, Right, Center game when j dollars are in the center.

3.12. Sources

The section on covariances is adapted from Carchidi and Higgins [**CH17**]. The section on first-step analysis is adapted from *Finite Markov Chains* by Kemeny and Snell [**KS60**]. Other ideas are from *An Introduction to Stochastic Modeling* by Taylor and Karlin and *Random Walks and Electrical Networks* by Doyle and Snell [**DS84**]. The birthday candle example is adapted from the January 13, 2017 "Riddler" at Fivethirtyeight.com/ The two colored balls example is adapted from *Finite Markov Chains* by Kemeny and Snell [**KS60**]. The four colored balls example is adapted from http://www.laurentlessard.com/bookproofs/colorful-balls-puzzle/ The Left-Center-Right game example is from the December 22, 2017 Riddler at Fivethirtyeight.com.

3.13. Outside Readings and Links:

(1) http://www.laurentlessard.com/bookproofs/colorful-balls-puzzle/

3.14. Algorithms and Scripts

Algorithm.

1 function playerIndex(j)
2 | Modular division to number the players in a circle
3 | **return** mod $((j-1), p) + 1$

4 function game(b, p)

5 Initialize the player circle to hold equal number of bills
6 while *more than one player has more than one bill* **do**
7 | Current player takes a `turn`
8 | Move to next player in circle
9 | Increment turns by 1
10 end
11 return turns

12 function turn(p)

13 Get 3 random integers from 1 to 6 into a vector
14 for *three places in the vector* **do**
15 | **if** *current player has bills* **then**
16 | | **if** *roll (vector entry) is* 1 *or* 2 **then**
17 | | | Player to left gets bill, current player loses a bill
18 | | **end**
19 | | **if** *roll (vector entry) is* 3 *or* 4 **then**
20 | | | Player to right gets bill, current player loses a bill
21 | | **end**
22 | | **if** *roll (vector entry) is* 5 *or* 6 **then**
23 | | | Center gets bill (don't keep track), current player loses a bill
24 | | **end**
25 | **end**
26 end
27 return State of players (the bills each player holds)

Algorithm 3: Supporting functions for Left-Center-Right

Data: Number of players, number of starting bills, number of simulations

Result: Matrices of mean and variances of game length

1 function main()

2 | Initialize Mean and Standard Deviation matrices to hold results

3 | Initialize number of simulations (5000)

4 | **for** $p \leftarrow 6$ **to** 9 **do**

5 | | **for** $b \leftarrow 3$ **to** 6 **do**

6 | | | Initialize empty vector to hold waiting time for each simulation

7 | | | **for** $i \leftarrow 1$ **to** *number of simulations* **do**

8 | | | | Play game

9 | | | | Record results of waiting times for each simulation

10 | | | **end**

11 | | **end**

12 | **end**

13 | **return** *Mean, Standard Deviation of Waiting Times*

Algorithm 4: Waiting Time Simulation for Left-Center-Right

Scripts. R script for Waiting Time in Left-Center-Right.

```r
player_index <- function(j) {
    ((j-1) %% player_count) + 1
}

game <- function(bills, player_count){

    turn <- function(player) {
        rolls <- sample(6, 3, replace=TRUE)
        for (roll in rolls) {
            if (players[player] > 0) {         #player has money
                if (roll <= 2) {               #pass dollar to Left
                    players[player_index(player - 1)] =
                        players[player_index(player - 1)] + 1
                    players[player] = players[player] - 1
                } else if (roll <= 4) {        #pass dollar to right
                    players[player_index(player + 1)] =
                        players[player_index(player + 1)] + 1
                    players[player] = players[player] - 1
                } else {                       #dollar goes to center
                    players[player] = players[player] - 1
                }
            }
        }
        return(players)
    }

    players <- rep(bills, player_count)
    turns <- 1
    current_player <- 1

    while ( length( players[ players > 0] ) > 1 ) {
        players <- turn(current_player)
        current_player <- player_index(current_player + 1)
        turns <- turns + 1
    }

    return(turns)
}

M <- matrix(0, 4, 4)
V <- matrix(0, 4, 4)
simulations <- 5000
```

```
43
44 for (player_count in c(6:9)) {
45     for (bills in c(3:6)) {
46
47         sim_results <- c()
48         for (i in 1:simulations) {
49             sim_results <- c( sim_results, game(bills, player_count) )
50         }
51
52         M[player_count-5, bills-2] <- mean(sim_results)
53         V[player_count-5, bills-2] <- sd(sim_results)
54     }
55 }
```

.

3.15. Problems to Work for Understanding

1: A law firm employs three types of lawyers: junior lawyers, senior lawyers, and partners. During a given year, there is probability 0.15 of promoting a junior lawyer to senior lawyer and probability 0.5 that the junior lawyer will leave the firm. There is probability 0.20 of promoting the senior lawyer to partner and probability 0.10 that the senior lawyer will leave the firm. Finally, there is probability 0.05 that a partner will leave the firm, see Table 3. The firm never demotes a lawyer or a partner.

Table 3. The transition probabilities

	leave firm	junior lawyer	senior lawyer	partner
leave firm	1	0	0	0
junior lawyer	0.05	0.80	0.15	0
senior lawyer	0.10	0	0.70	0.20
partner	0.05	0	0	0.95

(1) What is the average number of years that a newly hired junior lawyer stays with the firm?

(2) What is the variance of the number of years a newly hired junior lawyer stays with the firm?

2: Consider the random walk of a particle along a straight line from -4 to 4 with probability $2/3$ of moving to the right and probability $1/3$ of moving to the left. The absorbing boundary states are -4 and 4. Compute the vector of variances of waiting times until absorption using Theorem 3.3. Compare the result with the variance computed in the text for the walk started at 0, using the sum of the entries in the covariance matrix.

3: Consider three urns A, B, and C containing a, b and c balls respectively. Pick an urn at random, each being equally likely, and move a ball from it to one of the other two urns, each being equally likely. The game ends when one of the three urns becomes empty. Derive the mean duration of the game.

4: Use mathematical induction to show

$$P^m = \begin{pmatrix} I_a & 0 \\ A & T \end{pmatrix}^{m-1} \begin{pmatrix} I_a & 0 \\ A & T \end{pmatrix} = \begin{pmatrix} I_a & 0 \\ (I_t + T + T^2 + \cdots + T^{m-1})A & T^m \end{pmatrix}.$$

5:

(a) Use computer software to show the transient state transition probability matrix T in the random walk example has induced 2-norm (or operator norm) less than one for $p = 1/2$ and $p = 2/3$.

(b) Use computer software to show the transient state transition probability matrix T in the birthday candle example with $N = 5$, has induced 2-norm (or operator norm) less than 1.

(c) Use computer software to show the transient state transition probability matrix T in the two unpainted balls example has induced 2-norm (or operator norm) less than 1.

(d) (Mathematicians only) Show that if T is irreducible and row substochastic, with at least one row having sum less than 1, then the induced 2-norm (or operator norm) of T is less than 1, $\|T\| < 1$.

(e) Use computer software to show the transient state transition probability matrix T in the four painted balls example has induced 2-norm (or operator norm) *greater* than 1. Explain why the previous theorem is not violated. Nevertheless, show the eigenvalues of the matrix are all less than 1, so T is contractive and $(I - T)^{-1}$ still exists.

6: Show $NT = (I_t - T)^{-1}T = N - I_t$

7: Under the assumption the induced 2-norm (or operator norm) of T is less than 1, $\|T\| < 1$ prove that

$$\sum_{\nu=0}^{\infty}(T^\nu) = (I - T)^{-1}.$$

8: For the birthday candle problem, prove by induction that the waiting time with N candles is the Nth harmonic number H_N.

Standard Examples of Markov Chains

This chapter shows a wide variety of standard examples to illustrate simple Markov chains: sums of independent random variables, random walks on finite and infinite state spaces, and urn models. Each Markov chain is explained with their elementary properties and classifications.

Mathematically Mature: may contain mathematics beyond calculus with proofs.

4.1. Chapter Starter Question

Give an example from elementary probability of a simple Markov chain.

This chapter provides standard examples of Markov chains from simple probability, random walks, and urn models.

4.2. State Homogeneous Markov Chains

Simple State Homogeneous Markov Chain. Let ξ denote a discrete-valued random variable with possible values as the nonnegative integers $\mathbb{P}\left[\xi = i\right] = p_i$, where $p_i \geq 0$ and $\sum_{i=1}^{\infty} p_i = 1$. Let $\xi_1, \xi_2, \xi_3, \ldots$ be independent observations of ξ. Then consider the process X_i, $n = 1, 2, 3, \ldots$ defined by $X_n = \xi_n$. The transition probability matrix of this state homogeneous Markov chain is schematically

$$
P = \begin{pmatrix}
p_1 & p_2 & p_3 & \cdots \\
p_1 & p_2 & p_3 & \cdots \\
p_1 & p_2 & p_3 & \cdots \\
\vdots & \vdots & \vdots & \ddots
\end{pmatrix}.
$$

If $p_i = 0$ for $i > k$, then the state space of the Markov chain is finite, with corresponding $k \times k$ finite transition probability matrix.

In this example, make the sensible assumption that $0 < p_i < 1$ for the set of states, either finite or infinite, so that only states with a probability of appearing are considered. The stationary distribution is $p_i > 0$. Note that $P^n = P$. With the assumption, this Markov chain is

- regular,
- aperiodic,
- each state is accessible and all states communicate,
- recurrent,
- has no absorbing states.

This Markov chain is equivalent to independent samples from the distribution $\mathbb{P}[\xi = i] = p_i$.

Sums of Independent Random Variables as Markov Chains. Another standard Markov chain comes from successive partial sums η_n of the random variables of the independent observations of the ξ_i

$$\eta_n = \xi_1 + \xi_2 + \xi_3 + \cdots + \xi_n$$

with $\eta_0 = 0$ by definition. The entries of the probability transition matrix are defined by

$$
\begin{aligned}
&\mathbb{P}[X_{n+1} = j \mid X_n = i] \\
&= \mathbb{P}[\xi_1 + \xi_2 + \xi_3 + \cdots + \xi_n + \xi_{n+1} = j \mid \xi_1 + \xi_2 + \xi_3 + \cdots + \xi_n = i] \\
&= \mathbb{P}[\xi_{n+1} = j - i]
\end{aligned}
$$

assuming the independence of the ξ_i. Schematically the probability transition matrix is

$$
P = \begin{pmatrix}
p_1 & p_2 & p_3 & p_4 & \cdots \\
0 & p_1 & p_2 & p_3 & \cdots \\
0 & 0 & p_1 & p_2 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{pmatrix}.
$$

If the random variables ξ assumes both positive and negative integers, then the possible values of the partial sums η_n will be contained in the set of all integers. Then let the state space be $\mathbb{Z}$. Letting $\mathbb{P}[\xi = i] = p_i$, $i \in \mathbb{Z}$, with $p_i \geq 0$, $\sum_{i=-\infty}^{\infty} p_i = 1$. Then schematically the probability transition matrix is

$$
P = \begin{pmatrix}
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\cdots & p_{-1} & p_0 & p_1 & p_2 & p_3 & p_4 & \cdots \\
\cdots & p_{-2} & p_{-1} & p_0 & p_1 & p_2 & p_3 & \cdots \\
\cdots & p_{-3} & p_{-2} & p_{-1} & p_0 & p_1 & p_2 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{pmatrix}.
$$

This Markov chain is quite general and so special cases will be considered as examples of the various characteristics of Markov chains.

4.3. Random Walks as Markov Chains

Random Walk with Absorbing Boundaries. In **random walk with boundaries** the state space is the finite set of integers $\{0, 1, 2, 3 \ldots, k\}$. A particle at site i can stay at i with probability r_i or move to adjacent site $i+1$ with probability p_i and to adjacent site $i-1$ with probability q_i with $p_i + r_i + q_i = 1$. The transitions at the boundaries 0 and k get special consideration. The transition probability matrix in general is

$$P = \begin{pmatrix} r_0 & p_0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ q_1 & r_1 & p_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & q_2 & r_2 & p_2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & q_3 & r_3 & p_3 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & q_k & r_k \end{pmatrix}.$$

If $r_0 = r_k = 1$ the particle reaches sites 0 or k, then it stays at that site with probability 1. This is a **random walk with absorbing boundaries**. In a common special case, the transition probabilities do not depend on i and $r_i = 0$. The transition probability matrix is

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ q & 0 & p & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & q & 0 & p & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & q & 0 & p & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}.$$

This is also the Markov chain for the gambler's ruin.

All states are accessible from states i for $0 < i < k$, but because of the absorbing boundaries, not all states communicate. The transient states are $1, 2, \ldots, k-1$ and the absorbing states are 0 and k. Therefore, it does not make sense to consider any stable distributions.

Random Walk with Reflecting Boundaries. In a **random walk with reflecting boundaries** the state space is the finite set of integers $\{0, 1, 2, 3 \ldots, k\}$. A particle at site i with $1 < i < k-1$ moves to adjacent site $i+1$ with probability p and to adjacent site $i-1$ with probability q with $p+q = 1$. If the particle reaches sites 0 or k, then at the next step it is at 1 or $n-1$ respectively with probability 1. The transition probability matrix is

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ q & 0 & p & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & q & 0 & p & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \end{pmatrix}.$$

For this transition probability matrix P there is a probability distribution π such that $\pi P = \pi$. However it is not strictly true that the Markov chain has a stationary distribution because the chain has period 2. However, this can be fixed by

embedding this random walk in a *lazy random walk* sequence $X_0, X_1, \ldots, X_{n-1}, X_n, \ldots$ where given X_{n-1}, first choose to remain at X_{n-1} with probability $\frac{1}{2}$. Alternatively, with probability $\frac{1}{2}$ choose to move to X_n selected with probability q and p from the left and right nodes adjacent to x_{n-1}. The original chain is periodic with period 2 so is not regular, but the associated lazy random walk is aperiodic and regular. All states are accessible and all states communicate. This Markov chain has no transient states and all states are recurrent. The Markov chain has no absorbing states.

Random Walk on a Circle. In **random walk on a circle** the state space is the finite set of integers $\{0, 1, 2, 3 \ldots, k\}$. A particle at site i with $1 < i < k - 1$ moves to adjacent site $i + 1$ with probability p and to adjacent site $i - 1$ with probability q with $p + q = 1$. If the particle reaches sites 0, then at the next step it is at k or 1 with probability q or p respectively. If the particle reaches sites k, then at the next step it is at $k - 1$ or 0 with probability q or p respectively. The transition probability matrix is

$$P = \begin{pmatrix} 0 & p & 0 & 0 & 0 & \cdots & 0 & 0 & q \\ q & 0 & p & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & q & 0 & p & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ p & 0 & 0 & 0 & 0 & \cdots & 0 & q & 0 \end{pmatrix}.$$

This is also a random walk with state space $\mathbb{Z}_{k+1}$.

This Markov chain has a stationary distribution

$$(1/(k+1), 1/(k+1), \ldots, 1/(k+1))$$

(subject to the embedding into a lazy random walk as in the previous example.) The original chain is periodic with period 2 so it is not regular. All states are accessible and all states communicate. This Markov chain has no transient states and all states are recurrent. This Markov chain has no absorbing states.

Unbiased Walk on Integers. In **unbiased random walk on the integers**, also called symmetric random walk, the state space is the infinite set of integers $\mathbb{Z}$. A particle at site i with i moves to adjacent site $i + 1$ with probability $1/2$ and to adjacent site $i - 1$ with probability $1/2$. The transition probability is

$$p_{ij} = \begin{cases} 1/2 & j = i - 1 \\ 1/2 & j = i + 1 \\ 0 & j \neq i - 1, i + 1 \end{cases}.$$

Because this Markov chain has an infinite state space, it does not make sense to consider a stable distribution. The chain is periodic with period 2 so it is not regular. All states are accessible and all states communicate. It is a deep theorem that the Markov chain has no transient states and all states are recurrent, see Theorem 7 and the remarks following Theorem 7 in http://www.math.unl.edu/ sdunbar1/ProbabilityTheory/Lessons/BernoulliTrials/Recurrence/recurrence.html. Therefore, this is an example of an infinite state space Markov chain that is also recurrent. This Markov chain has no absorbing states.

Biased Walk on Integers. In **biased random walk on the integers** the state space is the infinite set of integers $\mathbb{Z}$. A particle at site i with i moves to adjacent site $i+1$ with probability p and to adjacent site $i-1$ with probability q with $p \neq q$ and $p+q = 1$. The transition probability is

$$p_{ij} = \begin{cases} p & j = i-1 \\ q & j = i+1 \\ 0 & j \neq i-1, i+1 \end{cases}.$$

Because this Markov chain has an infinite state space, it does not make sense to consider a stable distribution. The chain is periodic with period 2 so it is not regular. All states are accessible and all states communicate. It is a substantial theorem that all states are transient states and no states are recurrent, see Theorem 7 and the remarks following Theorem 7 in http://www.math.unl.edu/ sdunbar1/ProbabilityTheory/Lessons/BernoulliTrials/Recurrence/recurrence.html. Therefore, this is an example of an infinite state space Markov chain that is not recurrent. This Markov chain has no absorbing states.

Random Walks on Graphs. Each of these examples of a Markov chain is a specific case of a more general **random walk on a graph**. A **graph** is a finite or infinite set of nodes, also called vertices, $\{v_1, v_2, \ldots, v_k\}$, and an associated collection of edges $\{v_{ij}, 1 \leq i, j \leq n\}$. There need not be an edge between two vertices. If there is a connection between vertices v_i and v_j, then v_j is in the neighborhood $N(v_i)$ of v_i. A particle at vertex v_i will move to a vertex v_j in the neighborhood $N(v_i)$ with probability p_{ij} and may stay at v_i with probability p_{ii}, where $\sum_{j \in [\{i\} \cup N(v_i)]} p_{ij} = 1$.

Example 4.1. Consider the 3×3 square lattice graph in Figure 1. The graph has 9 vertices with 10 edges between nearest lattice neighbors. If a vertex has n edges then the probability of moving to a neighboring edge or staying at the vertex is $\frac{1}{n+1}$ uniformly. The random walk on this graph is often colorfully characterized as a frog hopping among lily pads or a bug leaping among plants. The transition probability matrix for this graph random walk is

$$\begin{pmatrix} 1/3 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 0 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 1/3 & 0 & 0 & 1/3 & 0 & 0 & 0 \\ 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 1/4 & 0 & 0 \\ 0 & 1/5 & 0 & 1/5 & 1/5 & 1/5 & 0 & 1/5 & 0 \\ 0 & 0 & 1/4 & 0 & 1/4 & 1/4 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & 1/3 & 0 & 0 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 0 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \end{pmatrix}.$$

This Markov chain has a stationary distribution

$$\pi = (\frac{1}{11}, \frac{4}{33}, \frac{1}{11}, \frac{4}{33}, \frac{5}{33}, \frac{4}{33}, \frac{1}{11}, \frac{4}{33}, \frac{1}{11}).$$
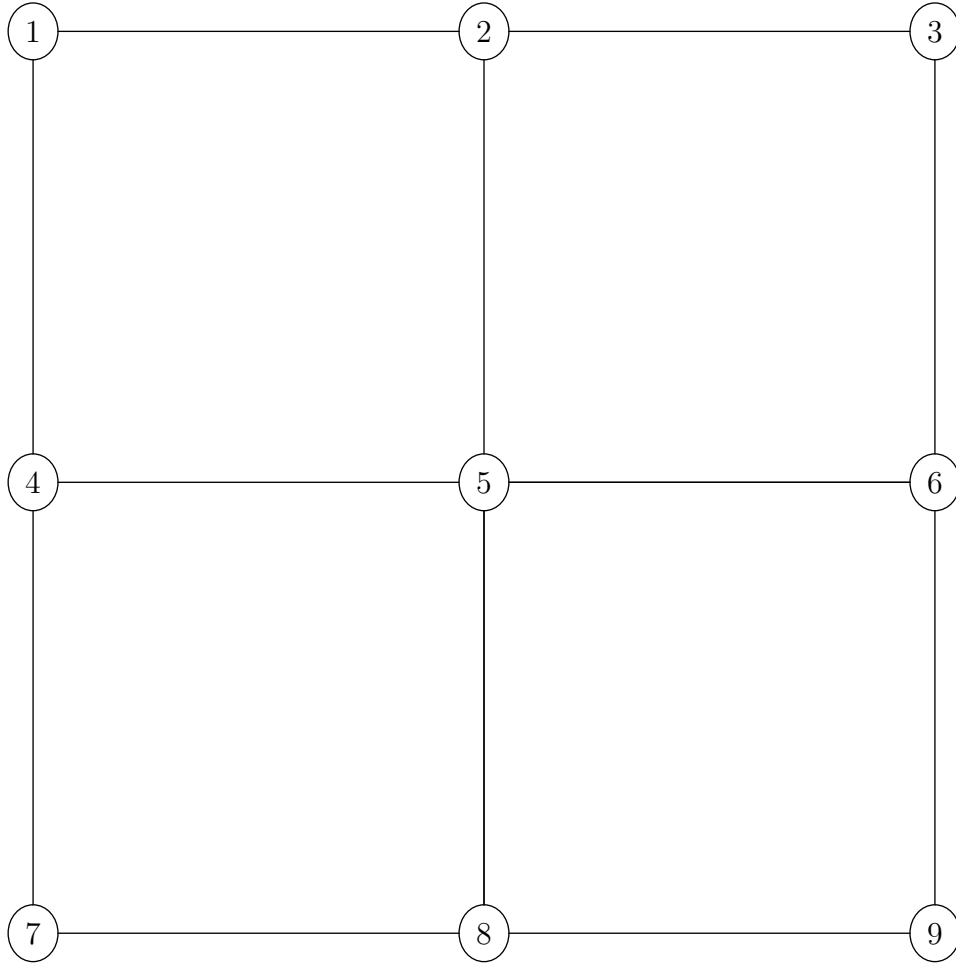
Figure 1. A 3×3 square lattice graph with uniform transition probabilities.

The chain is aperiodic. All states are accessible and all states communicate. This Markov chain has no transient states and all states are recurrent. This Markov chain has no absorbing states.

The stationary distribution π is unique and the chain converges to π as $n \to \infty$. Later sections will use this Markov chain as an example about quantitative rates of convergence, that is, how large n must be to make the chain sufficiently close to π. ∇

4.4. Urn Models

Ehrenfest Urn Model. The physicist P. Ehrenfest proposed the following model for statistical mechanics and kinetic theory. The motivation is diffusion through a membrane. Two urns labeled A and B contain a total of k balls. In the **Ehrenfest urn model** a ball is selected at random with all selections equally likely, and

moved from the urn it is in to the other urn. The state at each time is the number of balls in the urn A, from 0 to k. Then the transition probability matrix is

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ \frac{1}{k} & 0 & 1-\frac{1}{k} & 0 & \cdots & 0 & 0 \\ 0 & \frac{2}{k} & 0 & 1-\frac{2}{k} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1/k \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}.$$

The balls fluctuate between the two containers with a drift from the one with the larger number of balls to the one with the smaller numbers.

A stationary distribution for this Markov chain has entry $\pi_i = \binom{k}{i}/2^k$ subject to the embedding into a lazy Markov chain as in the previous examples. (See the exercises.) This is the binomial distribution on k, so that for large k, this can be approximated with the normal distribution. This conclusion is plausible given the physical origin of the Markov chain as a model for diffusion. All states are accessible and all states communicate. The chain is periodic with period 2 so it is not regular unless it is embedded in a lazy Markov chain. All states are recurrent with no transient states. This Markov chain has no absorbing states.

An Alternate Ehrenfest Urn Model. Two urns labeled A and B, contain a total of k balls. A ball is selected at random with all selections equally likely. Then an urn is selected, urn A with probability p and urn B with probability $q = 1 - p$ and the ball is moved to that urn. The state at each time is the number of balls in the urn A, from 0 to N. Then the transition probability matrix is

$$P = \begin{pmatrix} q & p & 0 & 0 & \cdots & 0 & 0 \\ \frac{1}{k}q & \frac{1}{k}p + \left(1-\frac{1}{k}\right)q & \left(1-\frac{1}{k}\right)p & 0 & \cdots & 0 & 0 \\ 0 & \frac{2}{k}q & \frac{2}{k}p + \left(1-\frac{2}{k}\right)q & \left(1-\frac{2}{k}\right)p & \cdots & 0 & 0 \\ 0 & 0 & \frac{3}{k}q & \frac{3}{k}p + \left(1-\frac{3}{k}\right)q & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & q & p \end{pmatrix}.$$

A stationary distribution for this Markov chain exists, but depends on p, i, and k in a more complicated relationship than in the previous example. All states are accessible and all states communicate. The chain is periodic with period 1 so it is regular. All states are recurrent and there are no transient states. This Markov chain has no absorbing states.

Another Ehrenfest Urn Model. Two urns labeled A and B contain a total of k balls. At time t there i balls in urn A. At time $t+1$ an urn is selected, urn A with probability i/k and urn B with probability $(k-i)/k$. Then independently a ball is selected from urn A with probability p or from urn B with probability $q = 1 - p$ and placed in the previously selected urn. The state at each time is the number of

balls in the urn A, from 0 to k. Then the transition probability matrix is

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \frac{k-1}{k}p & \frac{k-1}{k}q + \left(\frac{1}{k}\right)p & \left(\frac{1}{k}\right)q & 0 & \cdots & 0 & 0 \\ 0 & \frac{k-2}{k}p & \left(1 - \frac{2}{k}\right)q + \frac{2}{k}p & \left(\frac{2}{k}\right)p & \cdots & 0 & 0 \\ 0 & 0 & \frac{k-3}{k}p & \left(\frac{k-3}{k}\right)q + \frac{3}{k}p & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

States 1 through $k-1$ communicate while states 0 and k are absorbing. States 1 to $k-1$ are transient states. The chain is aperiodic.

Yet Another Generalized Ehrenfest Urn Model. Two urns labeled A and B, contain a total of k balls. At time t there k balls in urn A. At time $t+1$ an urn is selected, urn A with probability i/k and urn B with probability $(k-i)/k$. Then independently a ball is selected from urn A with probability i/k or from urn B with probability $(k-i)/k$ and placed in the previously selected urn. The state at each time is the number of balls in the urn A, from 0 to k. Then the transition probability matrix is

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \frac{1\cdot(k-1)}{k^2} & \frac{1^2}{k^2} + \frac{(k-1)^2}{k^2} & \frac{(k-1)\cdot 2}{k} & 0 & \cdots & 0 & 0 \\ 0 & \frac{2\cdot(k-2)}{k} & \frac{2^2}{k^2} + \frac{(k-2)^2}{k^2} & \frac{(k-2)\cdot 2}{k^2} & \cdots & 0 & 0 \\ 0 & 0 & \frac{3\cdot(k-3)}{k^2} & \frac{3^2}{k^2} + \frac{(k-3)^2}{k^2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

States 1 through $k-1$ communicate while states 0 and k are absorbing. States 1 to $k-1$ are transient states. The chain is periodic with period 1.

4.5. Chapter Ending Answer

A standard example from elementary probability theory that is also a Markov chain is the sequence of partial sums of a sequence of independent identically distributed random variables. This has the natural interpretation as a random walk, also a simple example of a Markov chain.

4.6. Chapter Summary

Key Concepts.

(1) A wide variety of standard examples illustrate simple Markov chains: sums of independent random variables, random walks on finite and infinite state spaces, and urn models.

(2) Random walks on finite state spaces can have absorbing or reflecting boundaries, or even circular boundaries. Different boundary conditions lead to different Markov chains.

(3) The physicist P. Ehrenfest proposed an urn model for statistical mechanics and kinetic theory. The motivation is diffusion through a membrane. Two urns labeled A and B, contain a total of N balls and the state is the number of balls in one urn. Different urn and ball selections lead to different Markov chains.

Vocabulary.

(1) In **random walk with boundaries** the state space is the finite set of integers $\{0, 1, 2, 3 \ldots, n\}$. A particle at site i can stay at i with probability r_i or move to adjacent site $i+1$ with probability p_i and to adjacent site $i-1$ with probability q_i with $p_i + r_i + q_i = 1$. The transitions at the boundaries 0 and n get special consideration.

(2) In a **random walk with absorbing boundaries**, $r_0 = r_n = 1$ so the particle reaches sites 0 or n, then it stays at that site with probability 1.

(3) In a **random walk with reflecting boundaries** if the particle reaches sites 0 or n, then at the next step it is at 1 or $n-1$ respectively with probability 1.

(4) In a **random walk on a circle** if the particle reaches site 0, then at the next step it is at n or 1 with probability q or p respectively. If the particle reaches site n, then at the next step it is at $n-1$ or 1 with probability q or p respectively.

(5) In **unbiased random walk on the integers**, also called symmetric random walk, the state space is the infinite set of integers $\mathbb{Z}$. A particle at site i moves to adjacent site $i+1$ with probability $1/2$ and to adjacent site $i-1$ with probability $1/2$.

(6) In **biased random walk on the integers** the state space is the infinite set of integers $\mathbb{Z}$. A particle at site i moves to adjacent site $i+1$ with probability p and to adjacent site $i-1$ with probability q with $p \neq q$ and $p + q = 1$.

(7) A **graph** is a finite or infinite set of nodes, also called vertices, $\{v_1, v_2, \ldots, v_n\}$, and an associated collection of edges $\{v_{ij}, 1 \leq i, j \leq n\}$. There need not be an edge between any two vertices, but if there is a connection between vertices v_i and v_j, then v_j is in the neighborhood $N(v_i)$ of v_i.

(8) In the more general **random walk on a graph** a particle at vertex v_i will move to a vertex v_j in the neighborhood $N(v_i)$ with probability p_{ij} and may stay at v_i with probability p_{ii}, where $\sum_{k \in [\{i\} \cup N(v_i)]} p_{ik} = 1$.

(9) In the **Ehrenfest urn model** a ball is selected at random from one of two urns with all selections equally likely, and moved from the urn it is in to the other urn. The state from 0 to N at each time is the number of balls in a fixed urn.

Notation.

(1) i, j, l – arbitrary states of the Markov chain

(2) ξ – a discrete-valued random variable with possible values as the nonnegative integers $\mathbb{P}[\xi = i] = p_i$

(3) $\xi_1, \xi_2, \xi_3, \ldots$ – independent observations of ξ

(4) X_n – Markov chain with transition probability matrix P

(5) successive partial sums η_n of the random variables of the independent observations of the ξ_i

(6) p_i, q_i, r_i – A particle at site i can stay at i with probability r_i or move to adjacent site $i+1$ with probability p_i and to adjacent site $i-1$ with probability q_i with $p_i + r_i + q_i = 1$.

(7) $p, q = 1 - p$ – simple Bernoulli probabilities

(8) n, t – time steps or periods of the Markov chain

(9) π – stationary distribution for a Markov chain

(10) k – related to the number of states of the Markov chain ($k + 1$ states if there is a 0 state)

4.7. Sources

This section is adapted from: *A First Course in Stochastic Processes*, by S. Karlin and H. Taylor, Chapter 2, pages 45–80, [**KT75**]; with more information and ideas from *Problems in Probability Theory, Mathematical Statistics and Theory of Random Functions* by A. A. Svesnikov, Chapter VIII, pages 231–274, [**Sve68**]

4.8. Outside Readings and Links:

4.9. Algorithms and Scripts

Data: State names and probability transition matrix
Result: Information about a simple Markov chain

1 *Initialization and sample paths*
2 Load Markov chain library
3 Set state names, set transition probability matrix, set start state
4 Set an example length and create a sample path of example length
5 Create a second sample path of example length

Algorithm.   **6** *Simulation of stationary distribution and comparison to theoretical*
7 Set a long path length, and a transient time
8 Create a long sample path
9 Slice the long sample path from the transient time to the end
10 In the slice count the appearance of each state
11 Store in an empirical array
12 Compute the theoretical stable array
13 **return** *Stable distribution and theoretical stable distribution*

Algorithm 5: Markov chain simulation.

Scripts.

R: R script for Random Walk with Absorbing Boundaries.

```
1 library(markovchain)
2
3 N <- 7
```

```
 4  stateNames <- as.character( 0:N )
 5  transMatrix <- matrix(0, N+1,N+1)
 6  transMatrix[1,1] <- 1
 7  transMatrix[N+1, N+1] <- 1
 8  for (row in 2:N) {  #row 2 is state 1, row 7 is state 6
 9      transMatrix[row, row-1] <- 1/2
10      transMatrix[row, row+1] <- 1/2
11  }
12  startState <- "3"
13
14  absorbingbdy <- new("markovchain", transitionMatrix=transMatrix,
15                      states=stateNames, name="AbsorbingBdy")
16  print(absorbingbdy)
17
18  print( summary(absorbingbdy) )
19
20  pathLength <- 10
21  path <- rmarkovchain(n=pathLength,
22                               object = absorbingbdy, t0 = startState)
23  cat("A sample starting from ", startState, path, "\n")
24  path <- rmarkovchain(n=pathLength,
25                               object = absorbingbdy, t0 = startState)
26  cat("Another sample starting from ", startState, path, "\n")
27
28  if ( length( transientStates(absorbingbdy) ) == 0 ) {
29      largeN <- 1000; startStable <- 100
30      path <- rmarkovchain(n=largeN,
31                               object = absorbingbdy, t0 = startState)
32      stablePattern <-   path[startStable:largeN]
33      empiricalStable <- rep(0, N+1)
34      for (i in seq_along(stateNames)) {
35          empiricalStable[i] <-  mean( stablePattern == (i-1) )
36      }
37      theorStable <- steadyStates(absorbingbdy)
38      cat("Empirical stable distribution: ",  empiricalStable, "\n")
39      cat("Theoretical stable distribution: ", theorStable, "\n")
40  }
```

.

R script for Random Walk with Reflecting Boundaries.

```
 1  library(markovchain)
 2
 3  N <- 7
 4  stateNames <- as.character( 0:N )
 5  transMatrix <- matrix(0, N+1,N+1)
 6  transMatrix[1,2] <- 1
 7  transMatrix[N+1, N] <- 1
 8  for (row in 2:N) {  #row 2 is state 1, row 7 is state 6
 9      transMatrix[row, row-1] <- 1/2
10      transMatrix[row, row+1] <- 1/2
11  }
12  startState <- "3"
13
14  reflectingbdy <- new("markovchain", transitionMatrix=transMatrix,
15                      states=stateNames, name="ReflectingBdy")
16  print(reflectingbdy)
17
18  print( summary(reflectingbdy) )
19
20  pathLength <- 10
21  path <- rmarkovchain(n=pathLength,
22                               object = reflectingbdy, t0 = startState)
23  cat("A sample starting from ", startState, path, "\n")
24  path <- rmarkovchain(n=pathLength,
25                               object = reflectingbdy, t0 = startState)
26  cat("Another sample starting from ", startState, path, "\n")
27
28  if ( length( transientStates( reflectingbdy) ) == 0 ) {
29      largeN <- 1000; startStable <- 100
```

```r
30      path <- rmarkovchain(n=largeN,
31                           object = reflectingbdy, t0 = startState)
32      stablePattern <-  path[startStable:largeN]
33      empiricalStable <- rep(0, N+1)
34      for (i in seq_along(stateNames)) {
35          empiricalStable[i] <-  mean( stablePattern == (i-1) )
36      }
37      theorStable <- steadyStates(reflectingbdy)
38      cat("Empirical stable distribution: ",  empiricalStable, "\n")
39      cat("Theoretical stable distribution: ", theorStable, "\n")
40 }
```

.

R script for Random Walk on a Circle.

```r
1  library(markovchain)
2
3  N <- 7
4  stateNames <- as.character( 0:N )
5  transMatrix <- matrix(0, N+1,N+1)
6  transMatrix[1,2] <- 1/2; transMatrix[1,N+1] <- 1/2
7  transMatrix[N+1, N] <- 1/2; transMatrix[N+1,1] <- 1/2
8  for (row in 2:N) {  #row 2 is state 1, row 7 is state 6
9      transMatrix[row, row-1] <- 1/2
10     transMatrix[row, row+1] <- 1/2
11 }
12 startState <- "3"
13
14 walkcircle <- new("markovchain", transitionMatrix=transMatrix,
15                   states=stateNames, name="Walkcircle")
16 print(walkcircle)
17
18 print( summary(walkcircle) )
19
20 pathLength <- 10
21 path <- rmarkovchain(n=pathLength,
22                      object = walkcircle, t0 = startState)
23 cat("A sample starting from ", startState, path, "\n")
24 path <- rmarkovchain(n=pathLength,
25                      object = walkcircle, t0 = startState)
26 cat("Another sample starting from ", startState, path, "\n")
27
28 if ( length( transientStates(walkcircle) ) == 0 ) {
29     largeN <- 1000; startStable <- 100
30     path <- rmarkovchain(n=largeN,
31                          object = walkcircle, t0 = startState)
32     stablePattern <-  path[startStable:largeN]
33     empiricalStable <- rep(0, N+1)
34     for (i in seq_along(stateNames)) {
35         empiricalStable[i] <-  mean( stablePattern == (i-1) )
36     }
37     theorStable <- steadyStates(walkcircle)
38     cat("Empirical stable distribution: ",  empiricalStable, "\n")
39     cat("Theoretical stable distribution: ", theorStable, "\n")
40 }
```

.

R script for Ehrenfest Urn Model.

```r
1  library(markovchain)
2
3  N <- 7
4  stateNames <- as.character( 0:N )
5  transMatrix <- matrix(0, N+1,N+1)
6  transMatrix[1,2] <- 1
7  transMatrix[N+1, N] <- 1
8  for (row in 2:N) {  #row 2 is state 1, row 7 is state 6
9      transMatrix[row, row-1] <- (row-1)/N
10     transMatrix[row,row+1] <- (N-(row-1))/N
```

```
11  }
12  startState <- "3"
13
14  ehrenfest1 <- new("markovchain", transitionMatrix=transMatrix,
15                      states=stateNames, name="Ehrenfest1")
16  print(ehrenfest1)
17
18  print( summary(ehrenfest1) )
19
20  pathLength <- 10
21  path <- rmarkovchain(n=pathLength,
22                                 object = ehrenfest1, t0 = startState)
23  cat("A sample starting from ", startState, path, "\n")
24  path <- rmarkovchain(n=pathLength,
25                                 object = ehrenfest1, t0 = startState)
26  cat("Another sample starting from ", startState, path, "\n")
27
28  if ( length( transientStates(ehrenfest1) ) == 0 ) {
29      largeN <- 1000; startStable <- 100
30      path <- rmarkovchain(n=largeN,
31                             object = ehrenfest1, t0 = startState)
32      stablePattern <-  path[startStable:largeN]
33      empiricalStable <- rep(0, N+1)
34      for (i in seq_along(stateNames)) {
35          empiricalStable[i] <-  mean( stablePattern == stateNames[i] )
36      }
37      theorStable <- steadyStates(ehrenfest1)
38      cat("Empirical stable distribution: ",  empiricalStable, "\n")
39      cat("Theoretical stable distribution: ", theorStable, "\n")
40  }
```

.

R script for alternative Ehrenfest Urn Model.

```
 1  library(markovchain)
 2
 3  N <- 7; p <- 1/2; q <- 1 - p;
 4  stateNames <- as.character( 0:N )
 5  transMatrix <- matrix(0, N+1,N+1)
 6  transMatrix[1,1] <- q
 7  transMatrix[1,2] <- p
 8  transMatrix[N+1, N  ] <- q
 9  transMatrix[N+1, N+1] <- p
10  for (row in 2:N) {
11      transMatrix[row, row-1] <- ((N-(row-1))/N)*p
12      transMatrix[row,row] <- ((N-(row-1))/N)*q + ((row-1)/N)*p
13      transMatrix[row,row+1] <- ((row-1)/N)*q
14  }
15  startState <- "3"
16
17  ehrenfest2 <- new("markovchain", transitionMatrix=transMatrix,
18                      states=stateNames, name="Ehrenfest2")
19  print(ehrenfest2)
20
21  print( summary(ehrenfest2) )
22
23  pathLength <- 10
24  path <- rmarkovchain(n=pathLength,
25                                 object = ehrenfest2, t0 = startState)
26  cat("A sample starting from startState ", startState, path, "\n")
27  path <- rmarkovchain(n=pathLength,
28                                 object = ehrenfest2, t0 = startState)
29  cat("Another sample starting from ", startState, path, "\n")
30
31  if ( length( transientStates(ehrenfest2) ) == 0 ) {
32      largeN <- 1000; startStable <- 100
33      path <- rmarkovchain(n=largeN,
34                             object = ehrenfest2, t0 = startState)
35      stablePattern <-  path[startStable:largeN]
36      empiricalStable <- rep(0, N+1)
```

```
37      for (i in stateNames) {
38          empiricalStable[i] <-  mean( stablePattern == stateNames[i] )
39      }
40      theorStable <- steadyStates(ehrenfest2)
41      cat("Empirical stable distribution: ",  empiricalStable, "\n")
42      cat("Theoretical stable distribution: ", theorStable, "\n")
43  }
```

.

R script for another alternative Ehrenfest Urn Model.

```
1  library(markovchain)
2
3  N <- 7; p <- 1/2; q <- 1 - p;
4  stateNames <- as.character( 0:N )
5  transMatrix <- matrix(0, N+1,N+1)
6  transMatrix[1,1] <- 1
7  transMatrix[N+1, N+1] <- 1
8  for (row in 2:N) {
9      transMatrix[row, row-1] <- ((N-(row-1))/N)*p
10     transMatrix[row,row] <- ((N-(row-1))/N)*q + ((row-1)/N)*p
11     transMatrix[row,row+1] <- ((row-1)/N)*q
12  }
13  startState <- "3"
14
15  ehrenfest3 <- new("markovchain", transitionMatrix=transMatrix,
16                 states=stateNames, name="Ehrenfest3")
17  print(ehrenfest3)
18
19  print( summary(ehrenfest3) )
20
21  pathLength <- 10
22  path <- rmarkovchain(n=pathLength,
23                              object = ehrenfest3, t0 = startState)
24  cat("A sample starting from ", startState, path, "\n")
25  path <- rmarkovchain(n=pathLength,
26                              object = ehrenfest3, t0 = startState)
27  cat("Another sample starting from ", startState, path, "\n")
28
29  if ( length( transientStates(ehrenfest3) ) == 0 ) {
30      largeN <- 1000; startStable <- 100
31      path <- rmarkovchain(n=largeN,
32                          object = ehrenfest3, t0 = startState)
33      stablePattern <-  path[startStable:largeN]
34      empiricalStable <- rep(0, N+1)
35      for (i in stateNames) {
36          empiricalStable[i] <-  mean( stablePattern == i )
37      }
38      theorStable <- steadyStates(ehrenfest3)
39      cat("Empirical stable distribution: ",  empiricalStable, "\n")
40      cat("Theoretical stable distribution: ", theorStable, "\n")
41  }
```

.

R script for yet another alternative Ehrenfest Urn Model.

```
1  library(markovchain)
2
3  N <- 7
4  stateNames <- as.character( 0:N )
5  transMatrix <- matrix(0, N+1,N+1)
6  transMatrix[1,1] <- 1
7  transMatrix[N+1, N+1] <- 1
8  for (row in 2:N) {  #row 2 is state 1, row 7 is state 6
9      transMatrix[row, row-1] <- (row-1)*(N-(row-1))/N^2
10     transMatrix[row,row] <- (row-1)^2/N^2 + (N-(row-1))^2/N^2
11     transMatrix[row,row+1] <- (row-1)*(N-(row-1))/N^2
12  }
13  startState <- "3"
```

```
14
15 ehrenfest4 <- new("markovchain", transitionMatrix=transMatrix,
16                   states=stateNames, name="Ehrenfest4")
17 print(ehrenfest4)
18
19 print( summary(ehrenfest4) )
20
21 pathLength <- 10
22 path <- rmarkovchain(n=pathLength,
23                       object = ehrenfest4, t0 = startState)
24 cat("A sample starting from ", startState, ":", path, "\n")
25 path <- rmarkovchain(n=pathLength,
26                       object = ehrenfest4, t0 = startState)
27 cat("Another sample starting from ", startState, ":", path, "\n")
28
29 if ( length( transientStates(ehrenfest4) ) == 0 ) {
30     largeN <- 1000; startStable <- 100
31     path <- rmarkovchain(n=largeN,
32                           object = ehrenfest4, t0 = startState)
33     stablePattern <-  path[startStable:largeN]
34     empiricalStable <- rep(0, N+1)
35     for (i in seq_along(stateNames)) {
36         empiricalStable[i] <-  mean( stablePattern == stateNames[i] )
37     }
38     theorStable <- steadyStates(ehrenfest4)
39     cat("Empirical long-time distribution: ",  empiricalStable, "\n")
40     cat("Theoretical stable distribution: ", theorStable, "\n")
41 }
```

.

4.10. Problems to Work for Understanding

1: For each of the urn models, express the transition probability P_{ij} of going to state j from state i in a general form, using the cases construction.

2: Prove that the stationary distribution for the first Ehrenfest Urn Model has entry $\pi_i = \binom{k}{i}/2^k$.

3: For the third and fourth Ehrenfest urn models with $k = 7$ and $p = q = 1/2$ which have absorbing states 0 and k, find the absorption probabilities and the expected waiting times to absorption from each transient state.

4: A certain system depends on a single machine that breaks down on any given day with probability p. It takes two days to restore the failed machine to normal service. Form a Markov chain by taking as states the pairs (x, y) where x is 1 if the machine is in operating condition at the end of a day, 0 otherwise and y is the number of days, 1 or 2 the machine has been broken. What is the long-term system availability?

5: Given m white balls and m black balls which are mixed thoroughly and then equally distributed in two urns. A and B. From each urn one ball is randomly drawn and placed in the other urn. Find the transition matrix and the probability that after a very large number of turns the first urn will have j white balls.

6: Unknown to public health officials, a person with a highly contagious disease enters the population. During each time period the person infects a new person with probability p or his symptoms appear and public health officials discover him, occurring with probability $1 - p$ and quarantine him so that he is unable to infect additional people. Compute the probability distribution of the number of infected

but undiscovered people in the population at the time of first discovery of a carrier. Assume each infected person behaves like the first.

The Game of Monopoly as a Markov Chain

This chapter assumes familiarity with the standard layout and rules of the board game Monopoly, in particular, the classic U.S. version based on Atlantic City NJ. The goal is to model the game as a Markov chain to find the stationary distribution to determine the real estate squares a player token lands on most frequently. Those squares on the board are "more valuable" in the sense that other players will land on them more frequently and pay rent more frequently. Deciding which color group squares are "best" is more complicated because of the variable costs of rent and improvement costs around the board and that aspect is not considered here. The main intention of this chapter is to show how to successively make changes to a simple Markov chain to create a more nuanced model incorporating more transitions among states. Another intention is to analyze a Markov chain larger than introductory Markov chains.

Student: contains scenes of mild algebra or calculus that may require guidance.

5.1. Chapter Starter Question

While playing the game of Monopoly, how often do you "take a walk on the Boardwalk"? That is, what is the fraction of time spent on the most expensive property in the game? How would you estimate this fraction?

5.2. A Simplified Four State Monopoly Game

The basic transition probability matrix around the Monopoly board is simply determined by the rolls of two fair six-sided dice. However making distant jumps due to the "Go To Jail" square and the drawing of certain *Community Chest* and *Chance* cards complicate the transition probability matrix. To understand this complication, first consider a much smaller example with just 4 spaces: Jail is state 1, Policeman is state 2, Community Chest is state 3, and Go is state 4, see Figure 1.
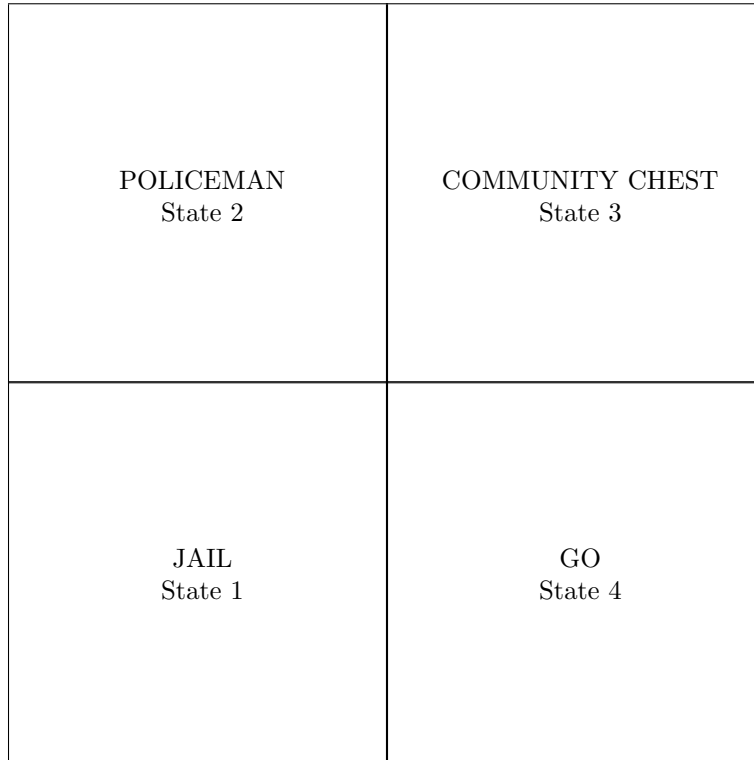
Figure 1. A Monopoly board with 4 spaces.

A fair two-sided "die" (i.e. a fair coin) determines the 1 or 2 step forward motion of a player token around the board. Then the "dice roll" transition matrix R is

$$R = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \end{pmatrix}.$$

(The source of this chapter, [**AR97**], uses the alternative convention that P_{ij} represents the probability of moving *to* state i *from* state j. Thus all matrices here are the transpose of matrices in [**AR97**].) This transition probability matrix is regular, and that the stationary distribution is $(1/4, 1/4, 1/4, 1/4)$.

The "Jail" matrix represents the transition from landing on the Policeman:

$$J = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

This matrix differs from the identity matrix only in the second row (Policeman) where, with probability 1, a player goes directly to state 1, Jail.

The last element is the Community Chest state. Of the 16 Community Chest cards (including the Get Out of Jail Free card), 14 penalize or reward some peculiar accomplishment, e.g. \$10 for second prize in a beauty contest. However, one card sends the player to Jail, i.e. state 1, and the other card advances the player to Go, state 4. To form the Community Chest matrix, again start with the identity matrix and edit the third row to get

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{16} & 0 & \frac{14}{16} & \frac{1}{16} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The transition matrix for the simplified 4-space board is then computed by the product

$$RJC = \begin{pmatrix} \frac{17}{32} & 0 & \frac{7}{16} & \frac{1}{32} \\ \frac{1}{32} & 0 & \frac{7}{16} & \frac{17}{32} \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

The order of the factors reflects the process of the play: A player rolls first and then follows the actions of where the token lands, i.e. go to Jail, or draw a Community Chest card. Now the second row of the matrix gives the transition probabilities for a token on the Policeman's square. However, this square is never occupied, the second column of the product is entirely 0. Note that the conditional probabilities of going from Policeman state 2 to other states are not 0. In fact, those conditional probabilities are exactly what would be expected starting from square 2. But the probability of being at the Policeman state is 0, so these correctly calculated probabilities are irrelevant. In terms of Markov chain states, the Go, Jail, and Community Chest states are communicating and the Policeman state is not accessible from the other states. Deleting the second column and row from the matrix gives the transitions between the remaining 3 occupiable states. The stationary distribution is $(\frac{16}{27}, \frac{7}{27}, \frac{4}{27})$. The high frequency of landing in Jail in this truncated game is not surprising.

5.3. Full 40 State Monopoly Game

The states of the Markov chain representing the full game are the 40 spaces around the board, numbered 1 through 40 with Mediterranean Avenue as 1 and Go as 40.

The 40×40 Rolling matrix has as its first row

$$[0, 0, \frac{1}{36}, \frac{2}{36}, \frac{3}{36}, \frac{4}{36}, \frac{5}{36}, \frac{6}{36}, \frac{5}{36}, \frac{4}{36}, \frac{3}{36}, \frac{2}{36}, \frac{1}{36}, 0, 0, \ldots, 0].$$

The second row is a rotated version of this row, add a zero at the front and shift the nonzero entries one column to the right. Form each succeeding row the same way until eventually the first entry is nonzero, as the rolls start to wrap around the board. This creates a **circulant matrix**, a square matrix in which each row vector is rotated one element to the right relative to the preceding row vector. Considering just this transition probability matrix, the steady state distribution is uniformly $\frac{1}{40} = 0.025$ which is natural.

For now, continue to ignore the more complicated game rules about exiting from the Jail square, and let the J matrix be just the identity matrix with the 1 in row 30, Policeman, moved from column 30 to column 10, Jail.

The Community Chest spaces or states are 2, 17, and 33. Community Chest cards are most likely to give you money leaving the state unchanged. But two of the cards move the player either to Jail, state 10 or to Go, state 40. To form the Community Chest matrix, Cc change the 1s on the diagonal in row 30 of the identity matrix to $\frac{14}{16}$ and enter $\frac{1}{16}$ in each of column 10, Jail, and column 40, Go.

The Chance spaces or states are 7, 22 and 36. A Chance card is more likely than a Community Chest card to move players, 10 of the 16 cards directing a player to a different state. The existence of cards such as "Go back three spaces" and "Advance to the nearest railroad" means the transitions depend on which Chance a player lands on. The Chance matrix Ch differs from the identity in rows 7, 22 and 36. It makes sense to multiply the Community Chest matrix *after* the Chance matrix, since there is a probability $\frac{1}{16}$ of moving from Chance state 36 to Community Chest state 33 and then to Go or Jail, each with probability $\frac{1}{16}$.

Finally, the full transition probability matrix is $P = RJChCc$. The stationary distribution gives a preliminary look at the proportion of time a player spends in each state over the course of a long game. The stationary distribution as a bar graph is in Figure 2. Table 1 summarizes the results with the 14 most frequently visited states in this model. The R script to generate the first two columns of Table 1 from the stationary distribution (eigenvector) of the transition probability matrix is in the Scripts section below. The last two columns come from a simulation with 10^6 steps. All states except 30 are communicating and the Policeman state 30 is not accessible from the other states. The frequencies computed here are essentially the same as the corresponding table in [**AR97**], with differences due to the number of significant digits reported.

5.4. Modeling Rules for Jail

This first model and simulation ignores two finer aspects of the game. This model assumes a player sent to Jail immediately pays the \$50 fine or uses a Get Out of Jail card to exit Jail on the next roll. But late in a game, when the board is covered with hotels and houses with high rent, players may choose to stay in Jail for a couple of turns to avoid paying rents. The rules allow this for two turns unless doubles are rolled, in which case a player must leave, exposed to nearby high rents. Because one must leave Jail on the third roll, the model now needs a record of how many unsuccessful rolls of doubles have occurred. The solution is to define two new Jail states, one for being newly Jailed, now state 41 and the other for having failed at doubles exactly once, state 42. This is a small example of a general principle of turning a process which is not Markov into a Markov chain by adding states, for a general discussion, see [**Str16**]. The original Jail state 10 serves as both the Just-Visiting and twice-failed-rolling-doubles states, since in both cases, the player leaves on the next turn.

Table 2 gives the results of one simulation with 10^6 steps of this expanded Markov chain with the Jail states 10, 41 and 42 combined. State 10 now is a state

	State	Frequency	Abbott State	Abbott Frequency	Simulation State	Simulation
1	10	0.05896420	10	0.05896	10	0.058909
2	24	0.03187795	24	0.03188	24	0.031934
3	40	0.03113803	40	0.03114	40	0.031183
4	19	0.03071024	19	0.03071	19	0.030969
5	25	0.03063697	25	0.03064	25	0.030858
6	5	0.02993127	5	0.02993	5	0.029701
7	15	0.02918612	15	0.02919	15	0.029198
8	18	0.02916517	18	0.02917	18	0.029070
9	20	0.02874826	20	0.02875	20	0.028704
10	21	0.02830354	21	0.02830	28	0.028230
11	28	0.02810737	28	0.02811	21	0.028152
12	16	0.02776751	16	0.02777	16	0.028065
13	23	0.02738577	23	0.02739	23	0.027349
14	11	0.02735991	11	0.02736	11	0.027226

Table 1. The 14 most-frequently visited states in the simple 40-state model.
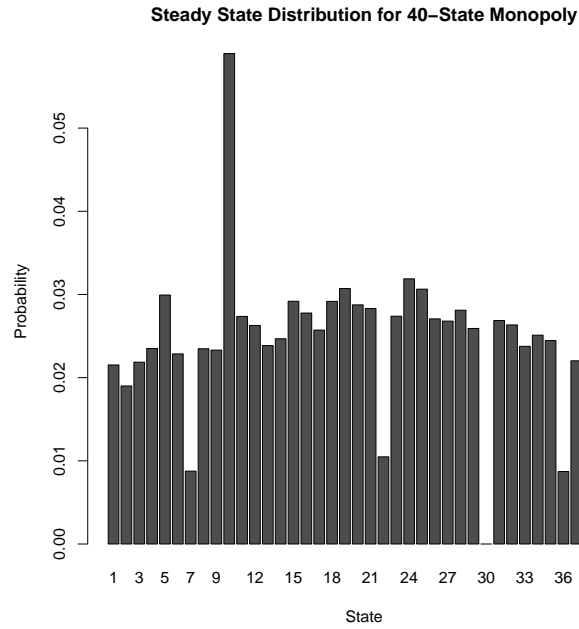
Figure 2. stationary distribution for the first 40-state model of Monopoly.

	State	Frequency	Abbott State	Abbott Frequency	Simulation State	Simulation
1	Jail	0.10799193	Jail	0.10800	Jail	0.108605
2	24	0.03012586	24	0.03013	24	0.030191
3	40	0.02948697	40	0.02949	40	0.029409
4	25	0.02905414	25	0.02905	25	0.029237
5	5	0.02845183	5	0.02845	19	0.028443
6	20	0.02821648	20	0.02822	5	0.028404
7	19	0.02821330	19	0.02821	20	0.028175
8	18	0.02811892	18	0.02812	18	0.028076
9	16	0.02674112	16	0.02674	28	0.026648
10	28	0.02669419	28	0.02669	16	0.026637
11	15	0.02658757	15	0.02659	15	0.026344
12	12	0.02636156	12	0.02636	12	0.026183
13	21	0.02627735	21	0.02628	21	0.026183
14	11	0.02602722	11	0.02603	11	0.026039

Table 2. The 14 most-frequently visited states in the 42 state model with Jail states 10, 41 and 42 combined.

reached by either a roll of the dice or by aging out of Jail so a player can move on to other states. The stationary distribution of this expanded Markov chain as a bar graph is in Figure 3, also with the Jail states 10, 41, and 42 combined.

The specific order of most frequently visited states is not as important as it appears in Table 2, since differences in frequency generally appear in the fourth significant digit. More useful is to consider *sets of most frequently visited states*. The most frequently visited set of states is 10, 41, 42, 24 and 40.

There is still one more detailed rule to consider, any player who rolls three doubles in a row goes to Jail. One way to handle this detail is to add 80 new states to the model as before, keeping track of history in a model with a total of 122 states. Again this illustrates the principle of turning a process which is not Markov into a Markov chain by adding states. An alternative way to handle this is to make an approximation by adjusting the Jail matrix J so that it sends a player not already in Jail to Jail with probability $1/216$ since the probability of rolling three doubles in a row is $\frac{1}{216}$. Replace each 1 on the diagonal with $\frac{215}{216}$ and add $\frac{1}{216}$ to column 41. After making these last adjustments to the probability transition matrix, Table 3 summarizes the results of simulation. The stationary distribution for this final Markov chain model as a bar graph is in Figure 4.

5.5. Additional Modeling Considerations

A few modeling assumptions are not explicitly considered in this model. One assumption is that Community Chest and Chance card decks are shuffled after each draw so that drawing the cards is a random event, possibly even repeating. The game actually has 17 Community Chest cards including a "Get Out of Jail Free" card, which can be saved by the player who draws it. Assuming that card has been

	State	Frequency
1	10	0.117841272
2	24	0.029795914
3	40	0.029055135
4	18	0.028235759
5	20	0.028148582
6	19	0.028019441
7	25	0.027108791
8	16	0.026894000
9	5	0.026694388
10	28	0.026549407
11	12	0.026091348
12	21	0.025963983
13	11	0.025481659
14	23	0.025471064
15	26	0.025384224
16	27	0.025192088
17	31	0.025076589
18	15	0.025064206
19	39	0.024766786
20	32	0.024499247
21	29	0.024398944
22	14	0.024354843
23	34	0.023334506
24	17	0.022984619
25	35	0.022713579
26	33	0.022086039
27	4	0.021986015
28	8	0.021786854
29	13	0.021756287
30	9	0.021606785
31	6	0.021335917
32	37	0.020478819
33	3	0.020476878
34	38	0.020438135
35	1	0.020087639
36	2	0.017758354
37	22	0.012115464
38	7	0.009514394
39	36	0.009452046
40	30	0.000000000

Table 3. The frequency of visited states in the final 42-state model with the approximation for going to Jail by rolling three doubles.

Steady State Distribution for 42–State Monopoly, 10,41,42 Combined
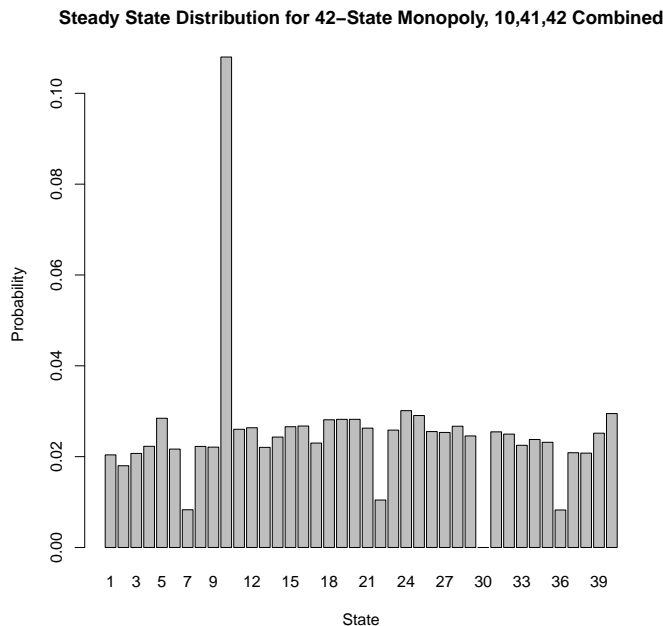
Figure 3. stationary distribution for the expanded 42-state model of Monopoly with Jail states 10, 41 and 42 combined.

drawn and retained gives the denominator of 16 used in the probabilities of the Community Chest matrix. Likewise, Monopoly uses 16 Chance cards but one is another "Get Out of Jail Free" card which can be saved. Assuming that card is still in the Chance deck gives the denominator of 16 used in the probabilities of the Chance matrix. This pair of mixed assumptions makes uniform denominators in the two matrices. Also, the final model assumes that appearance of a third consecutive roll of doubles is equally likely to occur at any space on the board, which is not strictly correct. Some squares are already known to be visited more than others and a third consecutive doubles could occur at an even number of squares further on the board from where the doubles run began.

In [**AR97**], Abbott and Richey use the stationary distribution to find the expected rental value and break-even times of the various color group properties. However, since illustration of the modeling and simulation of large Markov chains is the focus of these notes, these notes do not follow that investigation.

Many sites on the Internet make essentially the same Markov chain analysis of Monopoly in varying degrees of detail and in slightly different ways. For example, some use a 120-state chain to model the "rolling doubles" rule, and then condense the chain to the 40 board states. All reach essentially the same conclusions. Many of those sites go further and make an expected Return on Investment analysis to

Steady State Distribution for 42–State Monopoly, 10,41,42 Combined
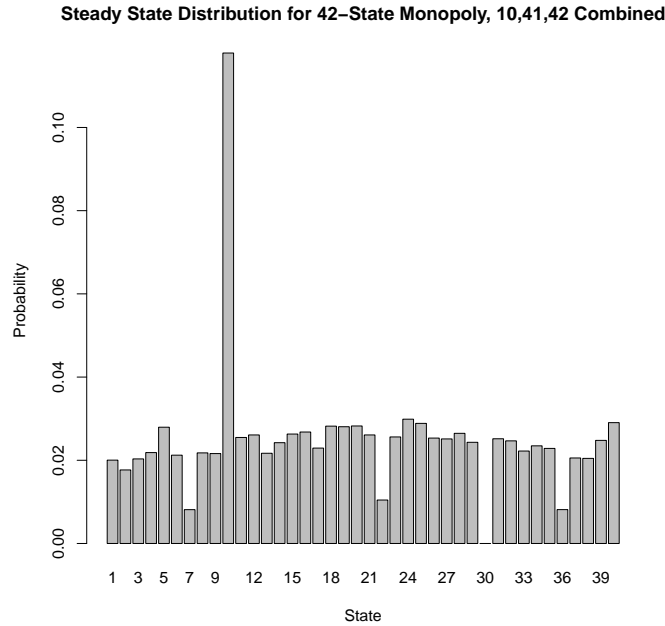
Figure 4. stationary distribution for the final 42-state model of Monopoly.

determine the most valuable spaces on the board, not just the most frequently visited. The Return on Investment analysis is not pursued here because the intention is to illustrate the Markov chain aspects.

5.6. Chapter Ending Answer

Since the board has 40 spaces, a first guess is that a token spends about $1/40 = 0.025$ of the time on Boardwalk. In fact, using the most detailed Markov chain here, 0.02431 of the time is spent at Boardwalk. It is less than the easy estimate because the model shows that a much larger fraction of time is spent in Jail, and in fact, not all spaces are equally likely to be occupied.

5.7. Chapter Summary

Key Concepts.

(1) Modeling the game of Monopoly as a Markov chain finds the stationary distribution to find the properties landed on most frequently.

(2) The modeling proceeds as a sequence of increasingly detailed Markov chains, each modeling more of the rules of the game.

(3) The transition probability matrix is composed as a product of simpler transition matrices, each modeling a specific type of moves around the board.

Vocabulary.

(1) A **circulant matrix** is a square matrix in which each row vector is rotated one element to the right relative to the preceding row vector.

Notation.

(1) R – the "dice roll" transition probability matrix for the four state Monopoly

(2) J – 'Jail" matrix represents the transition from landing on the Policeman

(3) C – four state Community Chest matrix

(4) Cc – full game Community Chest matrix

(5) Ch – full game Chance matrix

(6) P – full transition probability matrix

5.8. Sources

The section on Monopoly as a Markov chain is based on [**AR97**] with more information from [**AB72**]. Matthew Sheby pointed out an error in the Chance probability matrix in the scripts, now corrected so the stationary calculations agree with [**AR97**]. The simulation script is adapted from *Efficient R*, Chapter 7 by Colin Gillespie and Robin Lovelace.

5.9. Algorithms and Scripts

Data: States and basic transition probabilities
Result: stationary distribution for Monopoly game

Algorithm.

1 **function** `rotvec` *(v)*

2 Cyclically rotate vector elements one place to right

3 **return** rotated vector

4 *Inititialization of Matrices*

5 Load Markov Chain library

6 Set number of states and state names

7 Fill diceRoll matrix using `rotvec`

8 Fill diagonal Jail matrix with move from Police to Jail

9 Fill diagonal Community Chest matrix with moves to other states

10 Fill diagonal Chance matrix with moves to other states

11 Transition probability matrix as product of diceRoll, Jail, Chance and Community Chest

12 *Markov chain calculations*

13 Create Monopoly Markov chain object

14 Find stationary distribution of Monopoly Markov chain

15 **return** *Selected results in table or graph*

Algorithm 6: Monopoly Game Markov Chain

Data: Number of directly simulated Monopoly games
Result: Empirical stationary distributions from simulations

1 function moveSquare *(c, J)*

2 Simulate roll of 2 dice and

3 keep track of jail time

4 return new state c and jail time J

5 function checkSquare *(c)*

6 Cyclically reduce state about state 40

7 return state number

8 function updateStateVector *(c, J, l)*

9 Keep track of number of times a state is visited

10 return updated landings l

11 function communityChest *(c, J)*

12 Move state according to random selection

13 of Community Chest card

14 return new state c and jail time J

15 function chance *(c, J)*

16 Move state according to random selection of Chance card

17 return new state c and jail time J

18 main mySimulateMonopoly *(n)*

19 Initialize landings vector to hold occupancy of states

20 Initialize jail time J

21 Initialize first state at Go

22 for $i = 1$ *to numberOfTurns* **do**

23 | Move by moveSquare

24 | **if** *Chance square* **then**

25 | | move by chance

26 | **end**

27 | **if** *Community Chest square* **then**

28 | | move by communityChest

29 | **end**

30 | Update landings

31 end

32 Set number of simulations n

33 Run mySimulateMonopoly n times, divide by n

34 return *Relative Frequency of States from simulation*

Algorithm 7: Simulation of Monopoly game

Scripts.

R: R script for 40-state Monopoly.

```
library("markovchain")

rotvec <- function(vec) vec[ c(length(vec), 1:(length(vec)-1)) ]

```

```r
 5  nStates <- 40
 6  monopolyStates <- as.character( c(1:nStates) )
 7
 8  diceRoll <- matrix(0, nStates, nStates)
 9  p1 <- c(0, 0,
10          1/36, 2/36, 3/36, 4/36, 5/36, 6/36, 5/36, 4/36, 3/36, 2/36, 1/36,
11          numeric(nStates-13)
12          )
13  for (i in 1:nStates) {
14      diceRoll[i, ] <- p1
15      p1 <- rotvec(p1)
16  }
17
18  Jail <- diag(nStates)
19  Jail[30,30] <- 0
20  Jail[30, 10] <- 1
21
22  Ch <- diag(nStates)
23  chanceStates <- c(7, 22, 36)
24  chanceDests <- c(5, 10, 11, 24, 39, 40)
25  Ch[ cbind( chanceStates, chanceStates ) ] <- 7/16
26  Ch[ cbind( chanceStates, chanceStates-3 ) ] <- 1/16 #Go Back 3 spaces
27  Ch[ chanceStates, chanceDests] <- 1/16
28  Ch[7,12] <- 1/16; Ch[22,28] <- 1/16; Ch[36, 12] <- 1/16 #nearest utility
29  Ch[7,15] <- 1/16; Ch[22,25] <- 1/16; #nearest Railroad,
30  Ch[36, 5] <- 2/16 # note 2 ways to get to Reading Railroad
31
32  Cc <- diag(nStates)
33  chestStates <- c(2, 17, 33)
34  chestDests <-  c(10, 40)
35  Cc[ cbind(chestStates, chestStates) ] <- 14/16
36  Cc[chestStates, chestDests] <- 1/16
37
38  monopolyMatrix <- diceRoll %*% Jail %*% Ch %*% Cc
39
40  mcMonopoly <- new("markovchain", states = monopolyStates, byrow = TRUE,
41                    transitionMatrix = monopolyMatrix, name = "Monopoly")
42  ss <- steadyStates(mcMonopoly)
43
44  top14ofAbbott <- (1/100)*c(5.896, 3.188, 3.114, 3.071, 3.064, 2.993, 2.919,
          2.917, 2.875, 2.830, 2.811, 2.777, 2.739, 2.736)
45  top14ofAbbottIndices <- c(10, 24, 40, 19, 25, 5, 15, 18, 20, 21, 28, 16,
      23, 11)
46
47  sstop14 <- sort(ss, decreasing=TRUE)[1:14]
48  sstop14Indices <- order(ss, decreasing=TRUE)[1:14]
49
50  maxDiff <- max( abs( sstop14 - top14ofAbbott) )
51  cat("Maximum Difference: ", maxDiff, "\n")
52  maxDiffIndex <- which.max( abs( sstop14 - top14ofAbbott) )
53  cat("Maximum Difference Index: ", maxDiffIndex, "\n")
54  maxRelDiff <- maxDiff/sstop14[maxDiffIndex]
55  cat("Relative Maximum Difference: ", maxRelDiff, "\n")
56
57  ssdf <- data.frame( sstop14Indices, sstop14)
58
59  barplot(ss, names.arg=1:40,
60          main="Steady State Distribution for 40-State Monopoly",
61          xlab="State", ylab="Probability")
```

.

R script for 42-state Monopoly.

```r
 1    library("markovchain")
 2
 3  rotvec <- function(vec) vec[ c(length(vec), 1:(length(vec)-1)) ]
 4
 5  nStates <- 42
 6  monopolyStates <- as.character( c(1:nStates) )
 7
```

```
 8  diceRoll <- matrix(0, (nStates-2), (nStates-2))
 9  p1 <- c(0, 0,
10          1/36, 2/36, 3/36, 4/36, 5/36, 6/36, 5/36, 4/36, 3/36, 2/36, 1/36,
11          numeric((nStates-2)-13)
12          )
13  for (i in 1:(nStates-2)) {                #regular states 1 through 40
14      diceRoll[i, ] <- p1
15      p1 <- rotvec(p1)
16  }
17  ## Next, no probability of going from board state to extra Jail states
18  diceRoll  <- cbind(diceRoll, matrix(0, (nStates-2), 2))
19  ## add rows from extra Jail states
20  diceRoll  <- rbind(diceRoll, matrix(0, 2, nStates))
21  ## fill transitions for rolling doubles to get out of Jail
22  diceRoll[41, 41] <- 0; diceRoll[41, 42] <- 5/6 #stay in jail one turn
23  diceRoll[41, c(12, 14, 16, 18, 20, 22)] <- 1/36 #roll doubles
24  diceRoll[42, 42] <- 0; diceRoll[42, 10] <- 5/6 #stay in Jail one more turn
25  diceRoll[42, c(12, 14, 16, 18, 20, 22)] <- 1/36 #roll doubles
26
27  Jail <- diag(nStates)
28  Jail[30,30] <- 0
29  Jail[30, 41] <- 1                         #go from Police to Jail
30
31  Cc <- diag(nStates)
32  chestStates <- c(2, 17, 33)
33  chestDests <-  c(40, 41)
34  Cc[ cbind(chestStates, chestStates) ] <- 14/16
35  Cc[chestStates, chestDests] <- 1/16
36
37  Ch <- diag(nStates)
38  chanceStates <- c(7, 22, 36)
39  chanceDests <- c(5, 11, 24, 39, 40, 41)
40  Ch[ cbind( chanceStates, chanceStates ) ] <- 7/16
41  Ch[ cbind( chanceStates, chanceStates-3 ) ] <- 1/16 #Go Back 3 spaces
42  Ch[ chanceStates, chanceDests] <- 1/16
43  Ch[7,12] <- 1/16; Ch[22,28] <- 1/16; Ch[36, 12] <- 1/16 #nearest utility
44  Ch[7,15] <- 1/16; Ch[22,25] <- 1/16; #nearest Railroad,
45  Ch[36, 5] <- 2/16 # note 2 ways to get to Reading Railroad
46
47  monopMat <- diceRoll %*% Jail %*% Ch %*% Cc
48
49  mcMonopoly42 <- new("markovchain", states = monopolyStates, byrow = TRUE,
50                      transitionMatrix = monopMat, name = "Monopoly")
51  ss <- steadyStates(mcMonopoly42)
52
53  ssComb   <- ss
54  ssComb[10]   <- ssComb[10] + ssComb[41] + ssComb[42]
55  ssCombTop14 <- sort(ssComb[1:40], decreasing=TRUE)[1:14]
56  ssCombTop14Indices <- order(ssComb[1:40], decreasing=TRUE)[1:14]
57  ssCombdf <- data.frame( ssCombTop14Indices, ssCombTop14)
58
59  top14ofAbbott <- (1/100)*c(10.800, 3.013, 2.949, 2.905, 2.845, 2.822,
        2.821, 2.674, 2.669, 2.659, 2.659, 2.636, 2.628, 2.603)
60  top14ofAbbottIndices <- c(10, 24, 40, 25, 5, 20, 19, 18, 16, 28, 15, 12, 21
        , 11)
61
62  maxDiff <- max( abs( ssCombTop14 - top14ofAbbott) )
63  cat("Maximum Difference: ", maxDiff, "\n")
64  maxDiffIndex <- which.max( abs( ssCombTop14 - top14ofAbbott) )
65  cat("Maximum Difference Index: ", maxDiffIndex, "\n")
66  maxRelDiff <- maxDiff/ssCombTop14[maxDiffIndex]
67  cat("Relative Maximum Difference: ", maxRelDiff, "\n")
68
69  barplot(ssComb[1:40], names.arg=1:40,
70          main="Steady State Distribution for 42-State Monopoly, 10,41,42
        Combined",
71          xlab="State", ylab="Probability")
```

.

R script to simulate 42-state Monopoly.

```
 1    ##  This function calculates where you will land on your next move.
 2  moveSquare <- function(current, jailTime) {
 3      dice <- sample(seq(1,6), 2, replace=TRUE)
 4
 5      if (current == 10 & jailTime > 0) { # in Jail
 6          if (dice[1] == dice[2]) {           # roll doubles!
 7              current <- 10 + sum(dice)        # get out of Jail
 8              jailTime <- 0
 9          } else {                             #don't move but...
10              jailTime <- (jailTime + 1) %% 3 # count up jail time
11          }
12      } else {
13          current <- current + sum(dice)       # regular roll
14      }
15    return(c(current, jailTime))
16  }
17
18  ## Helper function to avoid code replication
19  checkState <- function(current) {
20    if (current > 40) {
21      current <- current - 40
22    } else if (current < 1) {
23      current <- current + 40
24    }
25    return(current)
26  }
27
28  ## Helper function to record landings,
29  updateStateVector <- function(current, jailTime, landings) {
30      if (current == 10 & jailTime > 0) {
31          landings[40 + jailTime] = landings[40 + jailTime] + 1
32      } else {
33          landings[current] <- landings[current] + 1
34      }
35
36      return(landings)
37  }
38
39  ## Move according to community chest cards
40  ## Use 16 cards, Get Out of Jail is reserved
41  communityChest <- function(current, jailTime) {
42    u <- runif(1)
43    goto <- current # Default. Do nothing
44    if (u < 1 / 16) {
45      goto <- 40 # Go
46    } else if (u < 2 / 16) {
47        goto <- 10 # Jail
48        jailTime <- 1
49    }
50    return( c(goto, jailTime) )
51  }
52
53  ## Move according to chance cards
54  ## Use 16 cards, Get Out of Jail is NOT reserved
55  chance <- function(current, jailTime) {
56    u <- runif(1)
57    goto <- current # Default. Do nothing
58    if (u < 1 / 16) {
59        goto <- 40 # Go
60    } else if (u < 2 / 16) {
61        goto <- 24 # Illinois Ave.
62    } else if (u < 3 / 16) {
63        goto <- 11 # St. Charles Place
64    } else if (u < 4 / 16) {
65        goto <- 10 # Jail
66        jailTime <- 1
67    } else if (u < 5 /16 ) {
68        goto <- 5 #Take ride on Reading
69    } else if (u < 6 / 16) {
```

```
70        goto <- 39 # Boardwalk
71    } else if (u < 7 / 16) {
72        goto <- checkState(current - 3) # Must check, since goto maybe
       negative!
73    } else if (u < 8 / 16) {
74        if (current > 28  | current < 12) { # Advance to nearest Utility
75            goto <- 12 # Water Works
76        } else {
77            goto <- 28 # Electricity
78        }
79    } else if (u < 9 / 16) { # Advance to nearest Railroad
80        if (current > 35  | current < 5) {
81            goto <- 5                    # Reading
82        } else if (current > 5  & current < 15) {
83            goto <- 15 #Pennsylvania
84        } else if (current > 15  & current < 25) {
85            goto <- 25 # B & O
86        } else if (current > 25  & current < 35) {
87            goto <- 35 # Short Line
88        }
89  }
90    return(c(goto, jailTime))
91  }
92
93  ## Main, record where landings occur
94  mySimulateMonopoly <- function (numberOfTurns)
95  {
96      landings <- numeric(42) # vector to hold occupancy of states,
97      # 1 is Med Ave, 40 is Go
98      # 41 is in Jail for one turn, 42 is Jail for two turns
99      # 10 is Just Visiting Jail or Jail for three turns, ready to get out
100     jailTime <- 0
101     current <- 40           # start at Go
102     for (i in 1:numberOfTurns) {
103         moveState <- moveSquare(current, jailTime)
104         current <- moveState[1]
105         jailTime <- moveState[2]
106         current <- checkState(current)
107         if (current == 30) {
108             current <- 10
109             jailTime <- 1
110         }
111         if (current == 7 | current == 22 | current == 36) {
112             move <- chance(current, jailTime)
113             current <- move[1]
114             jailTime <- move[2]
115         }
116         if (current == 2 | current == 17 | current == 33) {
117             move <- communityChest(current, jailTime)
118             current <- move[1]
119             jailTime <- move[2]
120         }
121         landings <- updateStateVector(current, jailTime, landings) # must
       do update
122     }
123  return(landings)
124  }
125
126  n   <- 1000
127  ss  <- mySimulateMonopoly(n)/n
```

.

5.10. Problems to Work for Understanding

1: Create a small Monopoly game with just 8 spaces:

 (1) Reading Railroad is state 1,

(2) Jail is state 2,

(3) Pennsylvania Railroad is state 3,

(4) Policeman is state 4,

(5) B. & O. Railroad is state 5,

(6) Chance is state 6,

(7) Short Line is state 7 and

(8) Go is state 8.

A fair two-sided "die" (i.e. a fair coin) determines the 1 or 2 step forward motion of a player token around the board. Considering this as a Markov Chain, find the stationary distribution. Assume the simple rule that a player leaves Jail immediately, no staying for 2 more turns. Be sure to include the effect drawing an "Advance to the Nearest Railroad", "Take a Ride on the Reading", "Go to Jail", "Advance to Go", or "Go back 3 spaces" card from the Chance deck. All other Chance movement cards (e.g. Illinois Avenue, Boardwalk, Utility) have no effect on movement.

The Game of Risk as a Markov Chain

This chapter uses Markov chains to analyze the stochastic progress of a war between two players in the board game *Risk*. Standard Markov chain methods give the probabilities of victory and additionally the expected losses in a war.

Mathematically Mature: may contain mathematics beyond calculus with proofs.

6.1. Chapter Starter Question

When rolling three fair six-sided dice with outcomes Y_1, Y_2, Y_3, what are the order statistics of the outcome?

6.2. The Game of Risk

The board game *Risk* for two or more players has a world map separated into 42 territories. Each player has a number of tokens used as armies. Each token represents one army. A player may use some or all but one of the armies to attack the armies in an adjacent opponent-controlled territory. The goal of a player is either to conquer the world through wars or to fulfill a task assigned at the beginning of the game. This task can be destroying a specific enemy or conquering a specific area.

The game consists of wars between two players, and each war consists of one or more rounds of combat. At each turn, a player must decide whether or not to attack a territory. The first question is the following: *If a player attacks a territory with some armies, what is the probability the player will capture this territory?* Of course, the probability that a player will capture a territory could be high while the expected loss is also high. Therefore, the second question is: *If a player engages in a war, how many armies should the player expect to keep, depending on the number of armies the opponent has on that territory?*

	Number of Armies		Number of Dice	
Case	Attacker	Defender	Attacker	Defender
1	1	1	1	1
2	2	1	2	1
3	≥ 3	1	3	1
4	1	≥ 2	1	2
5	2	≥ 2	2	2
6	≥ 3	≥ 2	3	2

Table 1. The number of dice each player rolls according to the number of attacker and defender armies.

At each turn, a player occupying some region may attack a neighboring territory. An attacker must leave at least one army on the occupied territory, and use at least one army in an attack. Thus a player who has at least two armies (one to keep on the territory and one to attack a territory) may attack a neighbor territory. A player who decides to attack a territory declares war on the opponent and the two players engage in a sequence of combats with random outcomes determined by dice. Outcomes of the defender's and the attacker's dice decide each battle and the ultimate outcome of the war. An attacker may withdraw before the war ends. An attacker who does not withdraw either destroys all the defender armies in that territory and occupies that territory, or loses all the attacking armies and fails to conquer the territory.

The number of armies an attacker or a defender possesses at the start of a each battle determines the number of dice each player rolls. To begin a round of war, the attacker will roll one die if attacking with a single army, two dice if the attacker has two armies present, and three dice if the attacking player has three or more armies involved in the war. A defender rolls one die with one army and two dice with two or more armies in the embattled territory. Six different cases result from the number of dice each opponent rolls, see Table 1.

At each turn, after rolling the dice, each player sets the dice in descending order and then pairs off the two sets of dice, matching the highest attacking die roll against the highest defending die roll. If each player is using at least two dice, then the second highest die rolls are similarly matched. The attacker loses one army for each die that is less than or equal to the corresponding defender's die. The defender loses one army for each die that is strictly less than the corresponding attacker's die. Note that ties favor the defender, so each tie results in the loss of an attacking army. After taking away the armies lost at that turn, the players roll the dice again. This sequence continues until one side loses all of its armies. Table 2 shows the progress of a typical war for a country. This war took four battles. At the end of the fourth turn, the attacker lost all of the attacking armies and the defender wins the war. This table also serves as a reminder of the number of dice rolled in various situations, never more than three for the attacker, and never more than two for the defender.

It may be disadvantageous to take control of a new territory by winning a war with only one army remaining, emphasizing the importance of the second question:

Roll	Armies		Dice Rolled		Dice Outcome		Losses	
	Att	Def	Att	Def	Att	Def	Att	Def
1	4	3	3	2	5,4,3	6,3	1	1
2	3	2	3	2	5,5,3	5,5	2	0
3	1	2	1	2	6	4,3	0	1
4	1	1	1	1	5	6	1	0
5	0	1						

Table 2. An example of a war with 4 battles

.

If a player engages in a war, how many armies should the player expect to lose depending on the number of armies the opponent has on that territory?

6.3. Risk States

Let A be the number of attacking armies and D the number of defending armies. The values of A and D characterize the state of the system at a given time. Let X_n be the state of the system at the beginning of the nth turn:

$$X_n = (a_n, d_n), \quad 0 \le a_n \le A, 0 \le d_n \le D$$

where a_n and d_n are the number of attacking and defending armies respectively. The initial state of the system is $X_0 = (A, D)$. If one side loses all its armies then that side loses the war; i.e., if $X_n = (0, d_n)$ with $d_n > 0$, then the attacker has lost the war at the end of the turn. Similarly, if $X_n = (a_n, 0)$, with $a_n > 0$, then the attacker has won the war at the end of the turn. Given the number of armies each side has at a given round, then it is possible to calculate the probability that each side wins or loses the war without knowing the states of the system before that round. In other words, the war process has the Markov property.

The AD states with both a and d positive are transient. The $A+D$ states where either $a = 0$ or $d = 0$ are absorbing. Order the $AD + A + D$ possible states so that the AD transient states precede the $A + D$ absorbing states. Order the transient states with the attacker armies decreasing along a row for constant defenders, and the defender armies decreasing row by row, see Figure 1. The transient states are, in order

$$(A, D), (A - 1, D), (A - 2, D), \ldots, (1, D),$$
$$(A, D - 1), (A - 1, D - 1), \ldots, (1, D - 1),$$
$$\ldots$$
$$(A, 1), (A - 1, 1), \ldots, (1, 1).$$

Order the absorbing states by decreasing desirability of outcome for the attacker, including the absorbing states where the defender wins

$$(A, 0), (A - 1, 0), \ldots, (1, 0), (0, 1), (0, 2), \ldots, (0, D).$$

Note that this order is the reverse of the state order in [**Osb03**]. The order here proceeds in the general direction of the war, with the first state for the initial numbers of armies, then decreasing number of armies for both players. See Figure 1
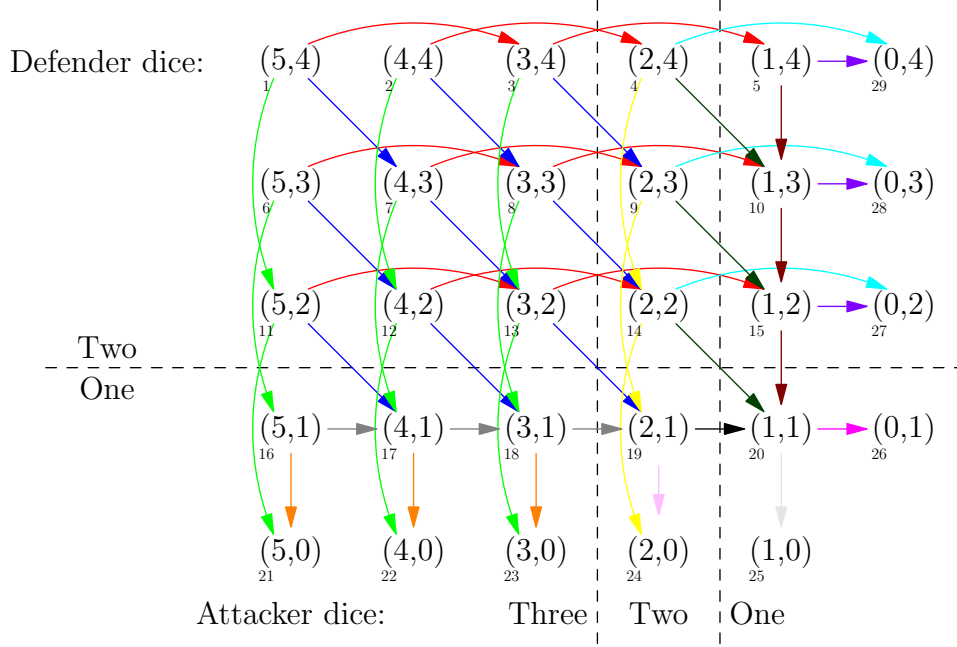
Figure 1. The state space and transitions of the Markov chain for a war in *Risk*. Transitions of the same color have the same probability.

for a diagram of the possible state transitions when $A = 5$ and $D = 4$. This figure is key for understanding the Markov chain. The small number below each ordered pair state is the index of the state for the transition probability matrix. The state indices increase from left to right and top to bottom, so the state of the system flows from the upper left corner to the bottom or right margin. The diagram divides into 6 regions, 3 for the attacker, and 2 for the defender depending on the number of dice each is rolling in a battle, as already outlined in Table 1. For example, the lower right region corresponds to Case 1 in Table 1 and the transition probabilities ρ_{111} and ρ_{110} detailed below in Table 3. The largest region in the upper left corresponds to Case 6 in Table 1 and the transition probabilities $\rho_{32\ell}$ in Table 3. Likewise for the other regions. Figure 1 labels the possible state transitions with arrows and colors corresponding to the transition probabilities in Table 3.

The states and the coordinates of states on the diagram can be confusing. For instance, the initial armies state $(5, 4)$ is at location $(x = 1, y = 4)$ on the diagram. As the attacking armies decrease, the first coordinate of the location of the state increases. As the defending armies decrease, the second coordinate of the location of the state decreases. This means:

- writing computer code for the chart requires close attention to indices, and

- do not confuse location with state.

6.4. Transition Probabilities

Under this ordering, the transition probability matrix takes the block-matrix form

$$P = \begin{pmatrix} Q & R \\ 0 & I \end{pmatrix}$$

where the $(AD) \times (AD)$ matrix Q has the probabilities of going from one transient state to another, and the $(AD) \times (A+D)$ matrix R has the probabilities of going from a transient state into an absorbing state. Note that this is the opposite order than the order in the chapter on Waiting Time to Absorption. The transition probability matrix P contains only 14 distinct probability values, based on the number of dice rolled and the armies lost as a result. Let $\rho_{ij\ell}$ denote the probability that the defender loses ℓ armies when rolling j dice against an attacker rolling i dice. See Table 3. To calculate $\rho_{ij\ell}$, find the joint probabilities associated with the best and second best roll from 2 or 3 six-sided fair dice. Let Y_1, Y_2, Y_3 denote the unordered outcome for an attacker when rolling three dice and let W_1, W_2 denote the unordered outcome for an attacker when rolling two dice. Let Z_1, Z_2 denote the unordered outcome for a defender rolling two dice. Then Y_1, Y_2, Y_3, W_1, W_2 and Z_1, Z_2 are random samples from the discrete uniform distribution on the integers 1 through 6. For example,

$$\mathbb{P}\left[Y_j = y\right] = \begin{cases} \frac{1}{6} & y = 1,2,3,4,5,6 \\ 0 & \text{else.} \end{cases}$$

Given a sample of n identically distributed random values, the ℓth **order statistic** is the ℓth largest value. Using superscripts, the ordered random variables $Y^{(1)} \geq Y^{(2)} \geq Y^{(3)}$ are the order statistics for the 3-dice attacker roll, the ordered random variables $W^{(1)} \geq W^{(2)}$ are the order statistics for the 2-dice attacker roll, and the ordered random variables $Z^{(1)} \geq Z^{(2)}$ are the order statistics for the 2-dice defender roll. The joint distributions of the order statistics for specification of $\rho_{ij\ell}$ use straightforward techniques of enumeration. When rolling two dice, the joint distribution of $(Z^{(1)}, Z^{(2)})$ is

$$\mathbb{P}\left[Z^{(1)} = z^{(1)}, Z^{(2)} = z^{(2)}\right] = \begin{cases} \frac{1}{36}, & z^{(1)} = z^{(2)} \\ \frac{2}{36}, & z^{(1)} > z^{(2)} \\ 0, & \text{else.} \end{cases}$$

See Figure 2 for the probability of $Z^{(1)} = 5, Z^{(2)} = 3$. The *marginal distribution* of the best roll $Z^{(1)}$ sums the joint distribution over possible values of $z^{(2)} \leq z^{(1)}$:

$$\mathbb{P}\left[Z^{(1)} = z^{(1)}\right] = \frac{2z^{(1)} - 1}{36}$$

for $z^{(1)} = 1,2,3,4,5,6$. See Figure 3 for the probability of $Z^{(1)} = 5$. When rolling three dice, the distribution of the best two rolls is

$$\mathbb{P}\left[Y^{(1)} = y^{(1)}, Y^{(2)} = y^{(2)}\right] = \begin{cases} \frac{3y^{(1)} - 2}{216}, & y^{(1)} = y^{(2)} \\ \frac{6y^{(2)} - 3}{216}, & y^{(1)} > y^{(2)} \\ 0, & \text{else.} \end{cases}$$

	1	2	3	4	5	6
1						
2						
3					$\frac{1}{36}$	
4						
5			$\frac{1}{36}$			
6						

Figure 2. Visualization of $Z^{(1)} = 5$, $Z^{(2)} = 3$.

	1	2	3	4	5	6
1	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$
2	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$
3	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$
4	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$
5	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$
6	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$

Figure 3. Visualization of $Z^{(1)} = 5$.

See Figure 4 for the probability of $Y^{(1)} = 3$, $Y^{(2)} = 3$. See the exercises for the derivation of the entry for $Y^{(1)} = y^{(1)}$, $Y^{(2)} = y^{(2)}$ with $y^{(1)} > y^{(2)}$. The marginal distribution of the best roll is

$$\mathbb{P}\left[Y^{(1)} = y^{(1)}\right] = \frac{1 - 3y^{(1)} + 3\left(y^{(1)}\right)^2}{216}$$

for $y^{(1)} = 1, 2, 3, 4, 5, 6$. All of the probabilities are 0 for arguments that are not positive integers less than or equal to 6. The joint distribution of $W^{(1)}$ and $W^{(2)}$ is the same as that for $Z^{(1)}$ and $Z^{(2)}$.
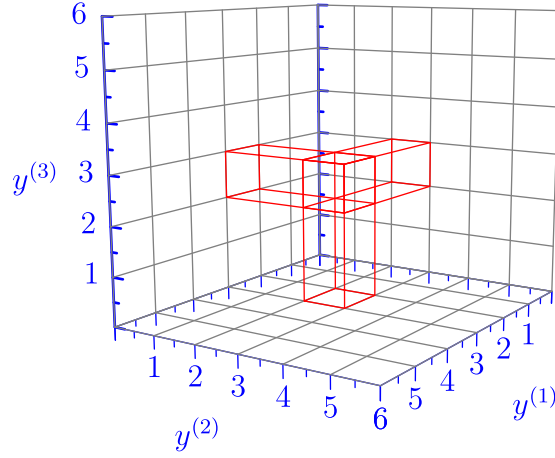
Figure 4. Visualization of $Y^{(1)} = 3, Y^{(2)} = 3$.

Now it is possible to calculate specific transition probabilities using these marginal distributions. For example

$$\rho_{322} = \mathbb{P}\left[Y^{(1)} > Z^{(1)}, Y^{(2)} > Z^{(2)}\right]$$

$$= \sum_{z_1=1}^{5} \sum_{z_2=1}^{z_1} \mathbb{P}\left[Y^{(1)} > z_1, Y^{(2)} > z_2\right] \mathbb{P}\left[Z^{(1)} = z_1, Z^{(2)} = z_2\right]$$

$$= \sum_{z_1=1}^{5} \sum_{z_2=1}^{z_1} \sum_{y_1=z_1+1}^{6} \sum_{y_2=z_2+1}^{y_1} \mathbb{P}\left[Y^{(1)} = y_1, Y^{(2)} = y_2\right] \mathbb{P}\left[Z^{(1)} = z_1, Z^{(2)} = z_2\right]$$

$$= \frac{2890}{7776}$$

$$\approx 0.372$$

Note that those events in this quadruple sum for which an argument with a subscript of 2 exceeds an argument with the same letter and subscript 1 have probability zero.

Obtain the probabilities ρ of the transition probability matrix P similarly using the joint distributions for $Y^{(1)}$, $Y^{(2)}$, for $Z^{(1)}$, $Z^{(2)}$, and for $W^{(1)}$, $W^{(2)}$. The exact probabilities, rounded to 3 decimal places, are in Table 3. The colors refer to the colors of the arrows representing transitions in Figure 1.

6.5. Absorption Probabilities

Given any initial state, the system *eventually* makes a transition to an absorbing state. For a transient state i, call $f_{ij}^{(n)}$ the probability that the first (and final) visit to absorbing state j occurs on the nth turn:

$$f_{ij}^{(n)} = \mathbb{P}\left[X_n = j, X_\nu \neq j \text{ for } \nu = 1, 2, \ldots, n-1 \mid X_0 = i\right].$$

Denote the $AD \times (A+D)$ matrix of these first transition probabilities by $F^{(n)}$. Then $F^{(n)} = Q^{n-1}R$. The probability that the system eventually goes from transient

Att Dice	Def Dice	Outcome	Symbol	Exact Prob.	Approx. Prob.	Color
1	1	Def loses 1	ρ_{111}	15/36	0.417	light gray
1	1	Att loses 1	ρ_{110}	21/36	0.583	magenta
1	2	Def loses 1	ρ_{121}	55/216	0.255	brown
1	2	Att loses 1	ρ_{120}	161/216	0.745	purple
2	1	Def loses 1	ρ_{211}	125/216	0.579	pink
2	1	Att loses 1	ρ_{210}	91/216	0.421	black
2	2	Def loses 2	ρ_{222}	295/1296	0.228	cyan
2	2	Each lose 1	ρ_{221}	420/1296	0.324	darkgreen
2	2	Att loses 2	ρ_{220}	581/1296	0.448	yellow
3	1	Def loses 1	ρ_{311}	855/1296	0.660	orange
3	1	Att loses 1	ρ_{310}	441/1296	0.340	gray
3	2	Def loses 2	ρ_{322}	2890/7776	0.372	green
3	2	Each lose 1	ρ_{321}	2611/7776	0.336	blue
3	2	Att loses 2	ρ_{320}	2275/7776	0.293	red

Table 3. Probabilities in the transition probability matrix. The colors correspond to the arrows in the diagram of state transitions.

Att Def	10	9	8	7	6	5	4	3	2	1
10	0.568	0.480	0.380	0.287	0.193	0.118	0.057	0.021	0.001	0.000
9	0.650	0.558	0.464	0.357	0.258	0.162	0.086	0.033	0.003	0.000
8	0.724	0.646	0.547	0.446	0.329	0.224	0.123	0.054	0.005	0.000
7	0.800	0.726	0.643	0.536	0.423	0.297	0.181	0.084	0.011	0.000
6	0.861	0.808	0.730	0.640	0.521	0.397	0.253	0.134	0.021	0.000
5	0.916	0.873	0.818	0.736	0.638	0.506	0.359	0.206	0.049	0.002
4	0.954	0.930	0.888	0.834	0.745	0.638	0.477	0.315	0.091	0.007
3	0.981	0.967	0.947	0.910	0.857	0.769	0.642	0.470	0.206	0.027
2	0.994	0.990	0.980	0.967	0.934	0.890	0.785	0.656	0.363	0.106
1	1.000	1.000	1.000	0.999	0.997	0.990	0.972	0.916	0.754	0.417

Table 4. Probability that the attacker wins with A armies against D defending armies.

state i to absorbing state j is $F = (I - Q)^{-1}R$. If the system ends in one of the A absorbing states then the attacker wins; if it ends in one of the D absorbing states, the defender wins. Since the initial state of a war is the first state using the order established previously, the probability that the attacker wins is just the sum of the entries in the first row of the submatrix of the first A columns of F:

$$\mathbb{P}\left[\text{Attacker wins} \mid X_0 = (A, D)\right] = \sum_{j=1}^{A} f_{1,j}$$

and

$$\mathbb{P}\left[\text{Defender wins} \mid X_0 = (A, D)\right] = \sum_{j=A+1}^{A+D} f_{1,j}.$$
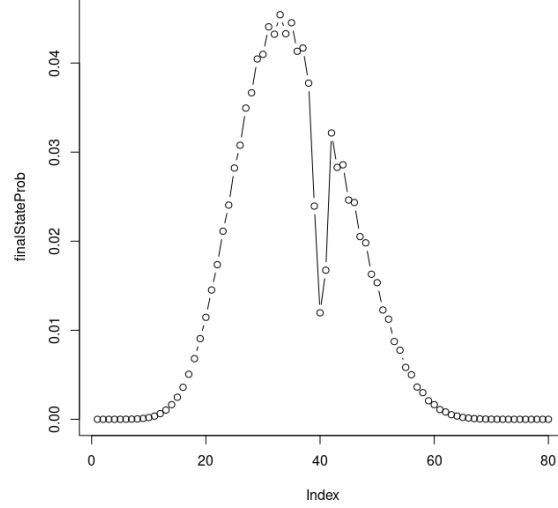
Figure 5. The probability mass function of absorbing states for $A = 40$, $D = 40$. The absorbing states are in decreasing order of preference for the attacker.

Because the state order here is the reverse of the order given in [**Osb03**], the table of winning probabilities here is the reverse of Table 3 in that paper. Note also by the Markov property, by calculating the winning probability for say $A = 10$ and $D = 10$, the winning probability for any war starting with $A < 10$ and $D < 10$ is also automatically calculated so that the entire Table 4 comes from $F = (I - Q)^{-1} R$ for $A = 10$ and $D = 10$.

Large engagements occur towards the end of an entire game of *Risk*. The absorbing probabilities for large wars have an interesting "saw-tooth" feature noticed in [**PW15**]. Figure 5 shows the probability mass function when the attacker and defender each start with 40 armies. The horizontal axis shows all the absorbing states of a $(40, 40)$ war and the vertical axis is the probability of the corresponding absorbing state. The outcome states along the horizontal axis are in decreasing order of preference of the attacker. One distinctive feature of this probability mass function is the local minimum at 40 in the probability masses. The outcomes $(1, 0)$ and $(0, 1)$ are less likely than their neighbors $(2, 0)$ and $(0, 2)$. The outcomes $(2, 0)$ and $(0, 2)$ can occur when a player with a single army will roll only one die against two dice and is less likely to survive. Also interesting is the jagged or "saw-tooth" phenomenon near the local maxima. A final outcome of $(8, 0)$ or $(6, 0)$ is each more likely than an outcome of $(7, 0)$. In the largest part of the transient states the probability of removing *two* armies from one player is about double the probability of each player losing *one* army. So a path that begins at $(40, 40)$ is more likely to visit only states (a, d) in which a and d have the same parity, except when one player keeps just one army.

20	19	18	17	16	15	14	13	12	11
0.633	0.628	0.623	0.617	0.611	0.605	0.599	0.592	0.584	0.576
10	9	8	7	6	5	4	3	2	1
0.568	0.558	0.547	0.536	0.521	0.506	0.477	0.470	0.363	0.417

Table 5. Probability of attacker winning when the number of armies are equal.

6.6. Expected Losses

For convenience, recall the definition of the normal matrix as $N = (I - Q)^{-1}$. Using the matrix NR allows calculation of the expected values and variances for the losses that the attacker and defender will suffer in a war. For example, suppose $A = 5$ and $D = 4$, as illustrated in Figure 1. In this case, the first row of the 20×9 matrix NR gives the probabilities for the $A + D = 9$ absorbing states: $(NR)_{1,\cdot} = (0.138, 0.165, 0.179, 0.107, 0.050, 0.070, 0.124, 0.104, 0.064)$. Let S_A and S_D denote the number of armies surviving for the attacker and defender respectively given the initial state $X_0 = (A, D)$. The probability distributions for S_A and S_D are in the first row of NR:

$$\mathbb{P}[S_A = \ell] = \begin{cases} \sum_{\nu=A+1}^{A+D} (NR)_{1,\nu} & \ell = 0, \\ (NR)_{1,A+1-\ell} & \ell = 1, 2, \ldots A \end{cases}$$

and

$$\mathbb{P}[S_D = \ell] = \begin{cases} \sum_{\nu=1}^{A} (NR)_{1,\nu} & \ell = 0, \\ (NR)_{1,\ell} & \ell = A + 1, A + 2, \ldots A + D. \end{cases}$$

For $A = 5$ and $D = 4$ the mean and standard deviation for the attacker's surviving armies are $\mathbb{E}[S_A] = 2.15$ and $\sqrt{\text{Var}[S_A]} = 1.37$. The defender's mean and standard deviation in this case are $\mathbb{E}[S_D] = 0.885$ and $\sqrt{\text{Var}[S_D]} = 1.11$. The attacker has an advantage in the sense that expected losses, 2.85, are lower than for the defender, 3.115. This is generally true provided the initial number of attacking armies is not too small.

To get an overall sense of the number of armies remaining after a war, Figure 6 has heatmaps of expected surviving armies for values of A and D from 1 to 20. A heatmap is a graphical representation of data where the individual values contained in a matrix are represented as colors. Here the lighter colors represent larger numbers of armies surviving, the darker colors smaller number of armies. The matrix is the number of armies surviving after a war between a number of attackers along a column and defenders along a row. The scripts calculate the expected number of armies surviving.

When the number of attacking and defending armies is equal, the probability that the attacker ends up winning the territory exceeds 50%, provided the initial stakes are high enough, at least 5 armies each, initially, see Table 5. With 40 armies attacking 40 defending armies, the expected armies surviving are 7.5 attacking armies and 2.1 defending armies. The overall conclusion is that the chances of winning a war are more favorable for the attacker. The logical recommendation is for the attacker to be aggressive.
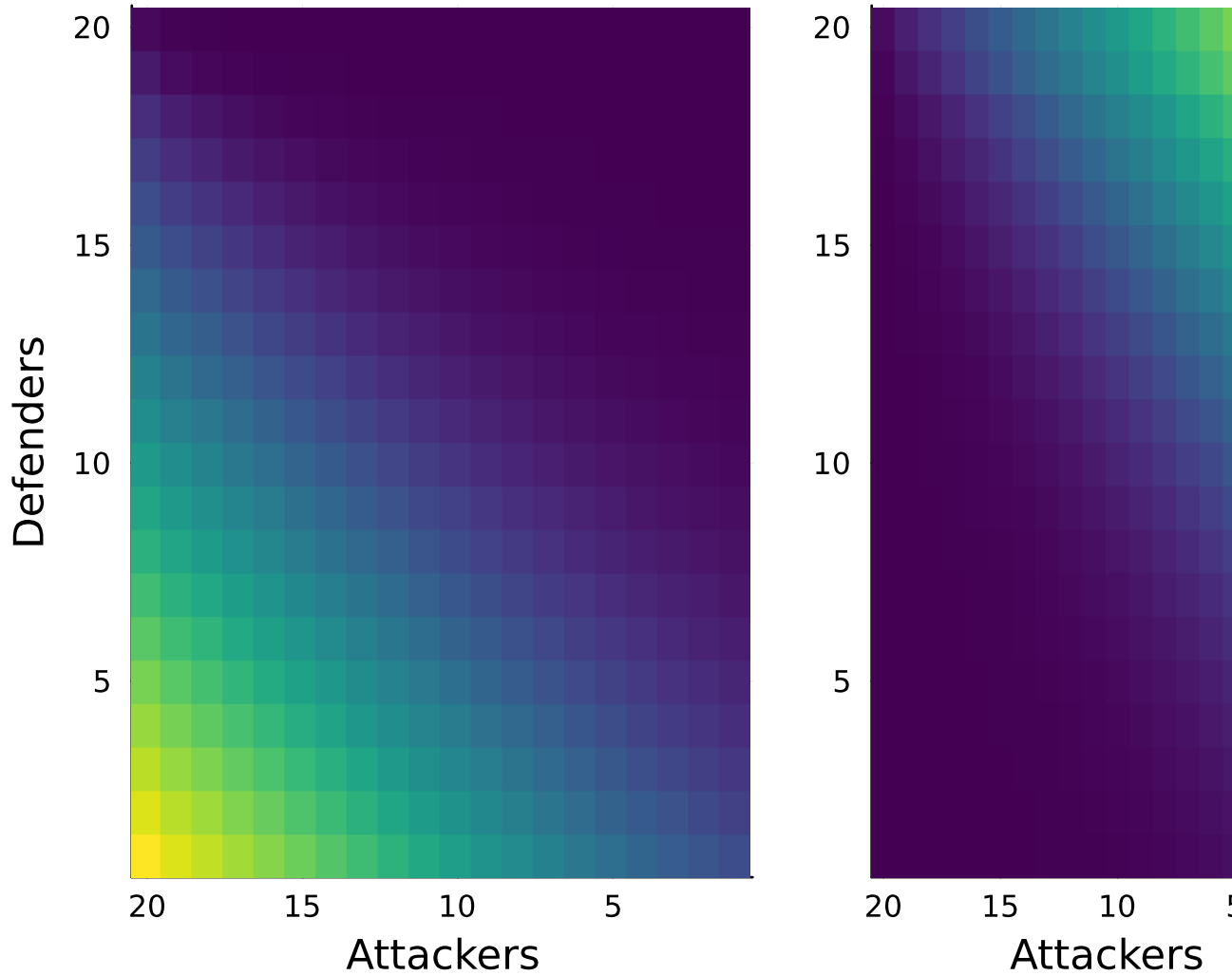
Figure 6. Expected surviving attacking armies on the left and surviving defending armies on the right after starting with A attacking armies and D defending armies. Lighter colors represent larger numbers of armies surviving, the darker colors smaller numbers of armies. Note the axis scales, the number of attackers decrease left to right, the number of defenders decrease top to bottom, as in Figure 1

Using the absorption probabilities for the states $(a, 0)$ where the attacker wins allows a more detailed summary of the outcomes of the war. As an example, suppose the attacker wants at least a probability of 0.75 of winning the war with at least four armies surviving. With how many armies should the attacker start a war to create this favorable situation? Refer to Figure 7 for an example of how to answer this important strategic question. Each facet in the figure indicates the

number of defending armies from 1 to 4. The horizontal axis in each facet lists the number of starting attacking armies, while the vertical axis in each facet displays how many attacking armies survive after the war. Respectively, the blue, green and red curves with points indicate the first-quartile, median, and third-quartile states in the distribution of outcomes based on the number of the attacker's starting armies. The outcomes are in descending order of preference for the attacker. For example, consider the lower right facet corresponding to the defender starting with 4 armies. If the attacker has seven armies, the most desired outcome is the absorbing state $(7, 0)$, while the least desired is $(0, 4)$. In this case, the median outcome is $(4, 0)$ in the sense that $(a, 0)$ with $a \geq 4$ has a probability of at least 0.50 (in fact, $\mathbb{P}[a \geq 4, d = 0] = 0.622$.) This means that with probability at least 0.50, the attacker will finish the war with 4 or more armies. The interquartile range is $(6, 0)$ through $(2, 0)$. This means that with probability at least 0.25 (precisely, the probability is 0.303) the attacker will win the war and have 6 or more armies surviving; with probability at least 0.50 (precisely, the probability is 0.503) the attacker will win and have 2 to 6 armies surviving; and the surviving 0.25 probability (precisely, the probability is 0.195) of battles will leave the attacker with 2 or fewer armies. Returning to the question of how many armies the player needs to start with to attack a defender with 4 armies with at least a probability of 0.75 of winning with at least four armies remaining, the attacker should start with at least 9 armies. In this case, the interquartile range is $(8, 0)$ through $(4, 0)$ and the median outcome is $(6, 0)$. The figure and the prior examples are illustrative, using up to 10 attacking armies and 4 defending armies. The R script below shows how to calculate these quartiles for any number of attackers and defenders.

The number of battles fought in a war is the number of dice rolls until absorption into one of the states $(a, 0)$ or $(0, d)$. This is the number of state transitions to absorption, also called the waiting time until absorption. The standard formula for calculating the expected waiting time to absorption is $(I - Q)^{-1}\mathbf{1}$. Using this, the expected number of battles for a war with 10 attacking armies against 4 defending armies is 3.994. The R script below shows how to calculate the expected waiting time to absorption for any number of attackers and defenders.

6.7. Chapter Ending Answer

The order statistics are $Y^{(3)}$, the largest value among Y_1, Y_2, Y_3; $Y^{(2)}$, the second largest value, including ties, among Y_1, Y_2, Y_3; and $Y^{(1)}$, the largest value, including ties, among Y_1, Y_2, Y_3.

6.8. Chapter Summary

Key Concepts.

(1) *Risk* is a board game of wars between two players, and each war consists of one or more rounds of combat between A attacking armies and D defending armies. The goal is to develop the probability of winning the war as a function of A and D using a Markov chain with $AD + A + D$ states.
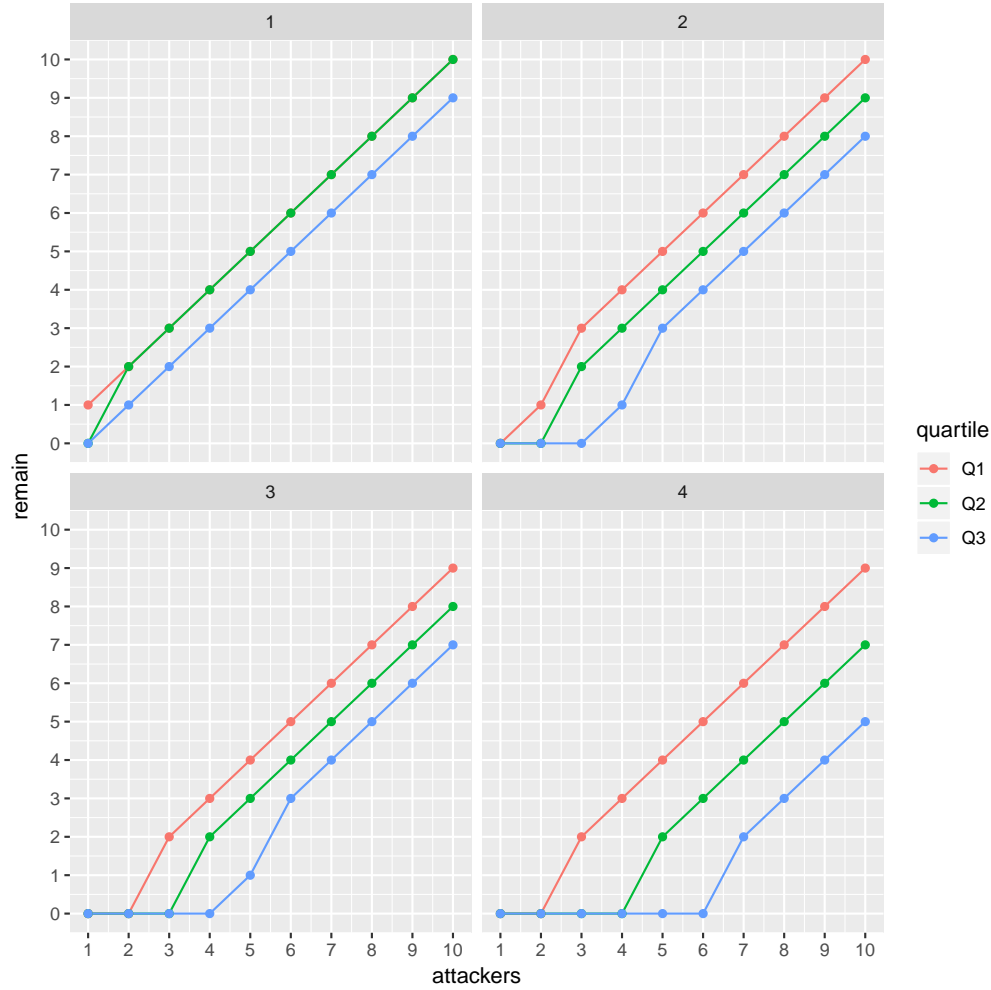
Figure 7. Quartile statistics for the number of attacking armies surviving in various scenarios.

Vocabulary.

(1) Given a sample of n identically distributed random values, the ℓth **order statistic** is the ℓth largest value.

Notation.

(1) A – the number of attacking armies

(2) D – the number of defending armies

(3) X_n – the state of the system at the beginning of the nth turn:

$$X_n = (a_n, d_n), 0 \leq a_n \leq A, 0 \leq d_n \leq D$$

where a_n and d_n are the number of attacking and defending armies respectively

(4) $\rho_{ij\ell}$ – the joint probabilities associated with the best and second best roll from 2 or 3 six-sided fair dice

(5) Y_1, Y_2, Y_3 – the unordered outcome for an attacker when rolling three dice

(6) W_1, W_2 – the unordered outcome for an attacker when rolling two dice

(7) Z_1, Z_2 – the unordered outcome for a defender rolling two dice

(8) $Y^{(1)} \geq Y^{(2)} \geq Y^{(3)}$ – the ordered random variables are the *order statistics*

(9) $f_{ij}^{(n)}$ the probability that the first (and final) visit form state i to absorbing state j occurs on the nth turn

(10) $F^{(n)}$ – $AD \times (A + D)$ matrix of these first transitions

(11) $F = (I - Q)^{-1}R$ – probability that the system eventually goes from transient state i to absorbing state j

(12) R_A and R_D – the number of armies remaining for the attacker and defender respectively given the initial state $X_0 = (A, D)$ probabilities

6.9. Sources

This chapter on *Risk* is adapted from [**Osb03**], [**PW15**] and [**Tan97**].

6.10. Algorithms and Scripts

q

Data: Maximum number of attacking and definding armies
Result: Absorption probabilities and win probabilities

Algorithm.

1 **function** `stateToIndex` (a, d)
2 Given a state, return the index for the transition probability matrix
   return index
3 **function** `indexToState` (n)
4 Given an index for the Markov chain, return the state (a, d)
5 **return** State
6 *Inititialization of Matrices*
7 Define 14 transition probabilities
8 Fill transition probability matrix
9 *Markov chain calculations*
10 Load Markov Chain library
11 Create *Risk* Markov chain object
12 Find absorption probabilities of *Risk* game chain
13 Sum up attacker and defender win probabilities

Algorithm 8: Risk Markov chain

Scripts.

R:

```
1
2  A <- 10
3  D <- 4
4  ## A <- 40
5  ## D <- 40
6
7  totalStates <- A * D + (A + D)
8
9  stateToIndex <- function(a,d) {
10     # given a state, return the index for the trans prob matrix
11     if (a > 0) {
12         ind <- (A+1 - a) + (D-d)*A
13     } else {
14         ind <- A*D + A + d
15     }
16     ind
17 }
18
19 indexToState <- function(ind) {
20     # given an index for a transient state, return the state as a pair
21     if (ind <= A*D + A) {
22         j <- A*D - ind
23         r <- j %% A
24         first <- 1 + r
25         second <- 1 + (j-r)/A
26     } else {
27         first <- 0
28         second <- ind - (A*D + A)
29     }
30     c(first, second)
31 }
32
33 rho111 <- 15/36                          # light gray
34 rho110 <- 21/36                          # magenta
35
36 rho121 <- 55/216                         # brown
37 rho120 <- 161/216                        # purple
38
39 rho211 <- 125/216                        # pink
40 rho210 <- 91/216                         # black
41
42 rho222 <- 295/1296                       # cyan
43 rho221 <- 420/1296                       # darkgreen
44 rho220 <- 581/1296                       # yellow
45
46 rho311 <- 855/1296                       # orange
47 rho310 <- 441/1296                       # gray
48
49 rho322 <- 2890/7776                      # green
50 rho321 <- 2611/7776                      # blue
51 rho320 <- 2275/7776                      # red
52
53 QR <- matrix(0, A * D, A * D + A + D)
54 ## R <- matrix(0, A * D, A + D)
55 I <- diag(A + D)
56 Z <- matrix(0, A + D, A * D)
57
58 att3dice <- seq(from=A, to = 3, by=-1)
59 def2dice <- seq(from=D, to = 2, by=-1)
60
61 # Cases 6, 5, 4
62 for (d in def2dice) {
63     for (a in att3dice) {
64         QR[stateToIndex(a,d), stateToIndex(a, d-2)] <- rho322    # green
65         QR[stateToIndex(a,d), stateToIndex(a-1, d-1)] <- rho321 # blue
66         QR[stateToIndex(a,d), stateToIndex(a-2, d)] <- rho320 # red
67     }
68     QR[stateToIndex(2, d), stateToIndex(2, d-2)] <- rho222 # cyan
69     QR[stateToIndex(2, d), stateToIndex(1, d-1)] <- rho221 # darkgreen
70     QR[stateToIndex(2, d), stateToIndex(0, d)] <- rho220 # yellow
```

```
71
72      QR[stateToIndex(1, d), stateToIndex(1, d-1)] <- rho121 # purple
73      QR[stateToIndex(1, d), stateToIndex(0, d)] <- rho120 # brown
74  }
75
76  # Case 3
77  for (a in att3dice) {
78      QR[stateToIndex(a, 1), stateToIndex(a, 0)] <- rho311 # orange
79      QR[stateToIndex(a, 1), stateToIndex(a-1, 1)] <- rho310 # gray
80  }
81
82  # Cases 2 and 1
83  QR[stateToIndex(2, 1), stateToIndex(2, 0)] <- rho211 # pink
84  QR[stateToIndex(2, 1), stateToIndex(1, 1)] <- rho210 # black
85  QR[stateToIndex(1, 1), stateToIndex(1, 0)] <- rho111 # lightgray
86  QR[stateToIndex(1, 1), stateToIndex(0, 1)] <- rho110 # magenta
87
88  M <- rbind(QR, cbind(Z, I))
89
90  library("markovchain")
91
92  riskChain <- new("markovchain", states = as.character(1:totalStates), byrow
        = TRUE,
93      transitionMatrix = M, name = "RISK")
94  absorpProb <- absorptionProbabilities(riskChain)
95  attackerWins  <- apply(absorpProb[ , 1:A], 1, sum)
96  attackerWinTable  <- t(matrix(formatC(attackerWins, digits=3, format="f"),
        A,D))
```

.

R:

```
1  A <- 20
2  D <- 20
3
4  totalStates  <- A * D + (A + D)
5
6  stateToIndex <- function(a,d) {
7      # given a state, return the index for the trans prob matrix
8      if (a > 0) {
9          ind <- (A+1 - a) + (D-d)*A
10     }  else {
11         ind <- A*D + A + d
12     }
13     ind
14 }
15
16 indexToAttackers <- function(ind) {
17     # for transient state index, return number of Attacking armies
18     j <- A*D - ind
19     r <- j %% A
20     attArmies <- 1 + r
21     attArmies
22 }
23
24 indexToDefenders <- function(ind) {
25     # given an index for a transient state, return number of Defending
         armies
26     j <- A*D - ind
27     r <- j %% A
28     defArmies <- 1 + (j-r)/A
29     defArmies
30 }
31
32 indexToState <- function(ind) {
33     # given an index for a transient state, return the state as a pair
34     if (ind <= A*D + A) {
35         j <- A*D - ind
36         r <- j %% A
37         first <- 1 + r
38         second <- 1 + (j-r)/A
```

```
39      } else {
40          first <- 0
41          second <- ind - (A*D + A)
42      }
43      c(first, second)
44 }
45
46 rho111 <- 15/36                          # light gray
47 rho110 <- 21/36                          # magenta
48
49 rho121 <- 55/216                         # brown
50 rho120 <- 161/216                        # purple
51
52 rho211 <- 125/216                        # pink
53 rho210 <- 91/216                         # black
54
55 rho222 <- 295/1296                       # cyan
56 rho221 <- 420/1296                       # darkgreen
57 rho220 <- 581/1296                       # yellow
58
59 rho311 <- 855/1296                       # orange
60 rho310 <- 441/1296                       # gray
61
62 rho322 <- 2890/7776                      # green
63 rho321 <- 2611/7776                      # blue
64 rho320 <- 2275/7776                      # red
65
66 QR <- matrix(0, A * D, A * D + A + D)
67 ## R <- matrix(0, A * D, A + D)
68 I <- diag(A + D)
69 Z <- matrix(0, A + D, A * D)
70
71 att3dice <- seq(from=A, to = 3, by=-1)
72 def2dice <- seq(from=D, to = 2, by=-1)
73
74 # Cases 6, 5, 4
75 for (d in def2dice) {
76     for (a in att3dice) {
77         QR[stateToIndex(a,d), stateToIndex(a, d-2)] <- rho322    # green
78         QR[stateToIndex(a,d), stateToIndex(a-1, d-1)] <- rho321 # blue
79         QR[stateToIndex(a,d), stateToIndex(a-2, d)] <- rho320 # red
80     }
81     QR[stateToIndex(2, d), stateToIndex(2, d-2)] <- rho222 # cyan
82     QR[stateToIndex(2, d), stateToIndex(1, d-1)] <- rho221 # darkgreen
83     QR[stateToIndex(2, d), stateToIndex(0, d)] <- rho220 # yellow
84
85     QR[stateToIndex(1, d), stateToIndex(1, d-1)] <- rho121 # purple
86     QR[stateToIndex(1, d), stateToIndex(0, d)] <- rho120 # brown
87 }
88
89 # Cases 3
90 for (a in att3dice) {
91     QR[stateToIndex(a, 1), stateToIndex(a, 0)] <- rho311 # orange
92     QR[stateToIndex(a, 1), stateToIndex(a-1, 1)] <- rho310 # gray
93 }
94
95 # Cases 2 and 1
96 QR[stateToIndex(2, 1), stateToIndex(2, 0)] <- rho211 # pink
97 QR[stateToIndex(2, 1), stateToIndex(1, 1)] <- rho210 # black
98 QR[stateToIndex(1, 1), stateToIndex(1, 0)] <- rho111 # lightgray
99 QR[stateToIndex(1, 1), stateToIndex(0, 1)] <- rho110 # magenta
100
101 M <- rbind(QR, cbind(Z, I))
102
103 library("markovchain")
104
105 riskChain <- new("markovchain", states = as.character(1:totalStates), byrow
        = TRUE,
106     transitionMatrix = M, name = "RISK")
107 absorpProb <- absorptionProbabilities(riskChain)
108
```

```R
109  ## Now I calculate the expected number of Attacking armies remaining
110  ## from each state.  This is the inner or dot product of the submatrix
111  ## absorpProb[ , 1:20] with the vector 20:1.  Note that the first column
112  ## of absorpProb[ , 1:20] is 20 Attacker armies remain, the second column
        of
113  ## absorpProb[ , 1:20] is 19 Attacking armies remain and so on.
114  ## This explains the use of the descending vector 20:1.
115  ## Furthermore, for example consider row 90, which corresponds to
116  ## state Att = 11 and Def = 16.  Then examination of the product shows
117  ## the first 9 entries of absorpProb[90, 1:20] are 0, positive
        probabilities
118  ## only start at entry 16, as described in the text, so the only
        contribution
119  ## to the expected number of attacking armies remaining will be from the
120  ## absorption probabilities of 11, 10, 9, ..., 1 armies remaining.
121
122  attArmiesRemain   <- absorpProb[ , 1:20] %*% 20:1
123
124  attackersRemain = data.frame(Att=indexToAttackers(1:400),
125                               Def=indexToDefenders(1:400),
126                               attRem=attArmiesRemain)
127
128  defArmiesRemain   <- absorpProb[ , 21:40] %*% 1:20
129
130  defendersRemain = data.frame(Att=indexToAttackers(1:400),
131                               Def=indexToDefenders(1:400),
132                               defRem=defArmiesRemain)
133
134  ## library("ggplot2")
135  ## ggplot(attackersRemain, aes(Att,Def)) +
136  ##     geom_raster(aes(fill=attRem)) +
137  ##     scale_x_reverse() +
138  ##     xlab("Attackers") +
139  ##     ylab("Defenders") +
140  ##     scale_fill_viridis_c()
141
142  ## ggplot(defendersRemain, aes(Att,Def)) +
143  ##     geom_raster(aes(fill=defRem)) +
144  ##     scale_x_reverse() +
145  ##     xlab("Attackers") +
146  ##     ylab("Defenders") +
147  ##     scale_fill_viridis_c()
```
. language=R

6.11. Problems to Work for Understanding

1: Given the values Y_1, Y_2, Y_3 from a roll of 3 dice, show that the probability for $Y^{(1)} = y^{(1)}$, $Y^{(2)} = y^{(2)}$ with $y^{(1)} > y^{(2)}$ is $\frac{6y^{(2)}-3}{216}$.

2: For the game of *Risk*, show that the probability of the attacker rolling three dice, the defender rolling one die and the defender losing one army is 855/1296. That is, calculate ρ_{311}, the orange arrow in Figure 1 and Table 3.

3: Find the expected number of attacking armies remaining when attacking an equal number of defending armies, for $n = 20$ to $n = 1$.

Games as Markov Chains

This chapter examines 4 additional games as Markov chains. The goal is to determine the expected number of moves or plays to reach a winning state. The games have from 11 states to 165 states and another goal is to show how to construct the states and the transition probability matrices in these cases.

Mathematically Mature: may contain mathematics beyond calculus with proofs.

7.1. Chapter Starter Question

What feature of a board game is necessary for it to be modeled as a Markov chain? Can you think of a board game that can be modeled as a Markov chain?

7.2. Tenzi

Tenzi is a simple dice game for multiple players. Each player has 10 six-sided fair dice in their hands. Taking turns, one player calls out, e.g., "Tenzi Four!" Everyone starts rolling their dice as quickly as possible. Whenever a "four" comes up, the player sets it aside and rolls the remaining dice. Eventually the player achieves all 10 "fours" and calls out "Tenzi!", becoming the winner of the round.

A Markov chain models the gameplay of *Tenzi* for a single player. Each state in the Markov chain represents the player's accumulated number of dice matching the target number. So at any point in the game, a player will be in one of the 11 states $\{0, 1, 2, 3, \ldots, 10\}$. A player in state 0 has not yet rolled any dice matching the target number, and a player in state 10 will call out "Tenzi!' Each time a player rolls the dice, the player has a chance to remain in the current state or to move up to a higher-numbered state.

Each entry in the transition probability matrix is a binomial probability. A player in state i rolls the remaining $10 - i$ dice. The player moves to state $j \geq i$ if exactly $j - i$ of the dice match the target number. So, the probability of transitioning from state i to state j in one roll of the dice is $\binom{10-i}{j-i} \left(\frac{1}{6}\right)^{j-i} \left(\frac{5}{6}\right)^{10-j}$ for $0 \leq i, j \leq 10$
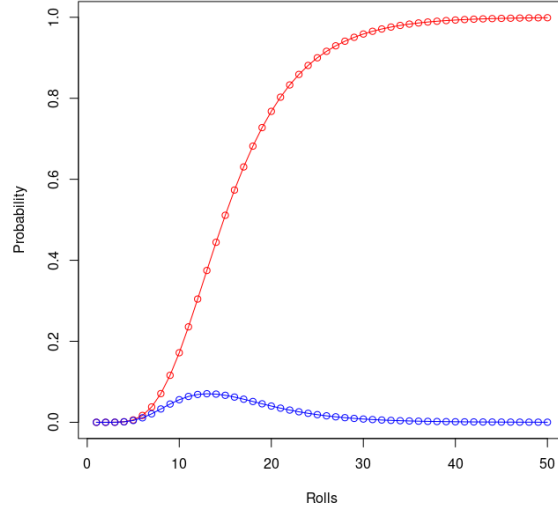
Figure 1. Probability density function (in blue) and cumulative distribution function (in red) of waiting time to win in Tenzi.

and $i \leq j$. A player will never go backwards to a previous state in the game, so the probability is 0 for $i > j$. Thus the states are not communicating, the Markov Chain is not recurrent, states 0 through 9 are transient and the state 10 is an absorbing state.

The full 11×11 transition probability matrix P is approximately

$$
\begin{pmatrix}
0.162 & 0.323 & 0.291 & 0.155 & 0.0543 & 0.013 & 0.00217 & 2.48 \cdot 10^{-4} & 1.86 \cdot 10^{-5} & 8.27 \cdot 10^{-7} & 1.65 \cdot 10^{-8} \\
0.0 & 0.194 & 0.349 & 0.279 & 0.13 & 0.0391 & 0.00781 & 0.00104 & 8.93 \cdot 10^{-5} & 4.47 \cdot 10^{-6} & 9.92 \cdot 10^{-8} \\
0.0 & 0.0 & 0.233 & 0.372 & 0.26 & 0.104 & 0.026 & 0.00417 & 4.17 \cdot 10^{-4} & 2.38 \cdot 10^{-5} & 5.95 \cdot 10^{-7} \\
0.0 & 0.0 & 0.0 & 0.279 & 0.391 & 0.234 & 0.0781 & 0.0156 & 0.00188 & 1.25 \cdot 10^{-4} & 3.57 \cdot 10^{-6} \\
0.0 & 0.0 & 0.0 & 0.0 & 0.335 & 0.402 & 0.201 & 0.0536 & 0.00804 & 6.43 \cdot 10^{-4} & 2.14 \cdot 10^{-5} \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.402 & 0.402 & 0.161 & 0.0322 & 0.00322 & 1.29 \cdot 10^{-4} \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.482 & 0.386 & 0.116 & 0.0154 & 7.72 \cdot 10^{-4} \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.579 & 0.347 & 0.0694 & 0.00463 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.694 & 0.278 & 0.0278 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.833 & 0.167 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0
\end{pmatrix} .
$$

The initial distribution is $X_0 = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$. The value in the last place of $X_n = X_0 P^n$ is the probability of reaching the absorbing or winning state for Tenzi in n or fewer rolls. For example, the value in the last entry of X_{20} is 0.7677. The player has a 76.77% chance the game finishes by the 20th roll. The player may have finished on the 20th roll, or possibly finished on some previous roll and is waiting in absorbing state 10. The resulting cumulative probability function $P(j)$ for a player getting Tenzi on or before the jth roll is in Figure 1. To find $p(j)$, the probability of getting Tenzi in exactly j rolls, subtract the probability that the player already had Tenzi by the $j-1$ roll. This gives the probability density function $p(j) = P(j) - P(j-1)$ also shown in Figure 1. The maximum of this probability function is at $j = 13$; a player has a 0.0705 probability of getting Tenzi on the 13th roll. Using these probabilities, the mean of the distribution is

0	1	2	3	4	5	6	7	8	9
1.19	0.48	0.67	0.79	0.92	1.10	1.37	1.83	2.74	5.48

Table 1. Expected number of times the chain will be in each state.

$\sum_{j=1}^{\infty} jp(j)$. Approximate this sum through $j = 50$ as 16.5. This is the average number of rolls needed for a player to get Tenzi.

The fundamental matrix $N = (I - Q)^{-1}$ provides another way of finding the mean. The transition probability matrix is already partitioned as

$$P = \begin{pmatrix} Q & R \\ 0 & I \end{pmatrix}$$

where Q is the 10×10 matrix containing the transition probabilities among the transient states, R is a 10×1 column matrix containing the probabilities of going from each of the non-absorbing states to state 10 in one roll, 0 is a 1×10 zero matrix, and I is the 1×1 identity matrix representing the absorbing state 10. Then N is approximately

$$\begin{pmatrix}
1.193 & 0.478 & 0.669 & 0.787 & 0.915 & 1.097 & 1.371 & 1.828 & 2.742 & 5.485 \\
0.000 & 1.240 & 0.564 & 0.771 & 0.917 & 1.097 & 1.371 & 1.828 & 2.742 & 5.485 \\
0.000 & 0.000 & 1.303 & 0.673 & 0.905 & 1.099 & 1.371 & 1.828 & 2.742 & 5.485 \\
0.000 & 0.000 & 0.000 & 1.387 & 0.815 & 1.091 & 1.373 & 1.828 & 2.742 & 5.485 \\
0.000 & 0.000 & 0.000 & 0.000 & 1.504 & 1.010 & 1.368 & 1.829 & 2.742 & 5.485 \\
0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 1.672 & 1.298 & 1.826 & 2.743 & 5.485 \\
0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 1.931 & 1.769 & 2.742 & 5.485 \\
0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 2.374 & 2.697 & 5.485 \\
0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 3.273 & 5.455 \\
0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 6.000
\end{pmatrix}.$$

The theorems from the chapter on Waiting Time to Absorption give a lot of information about the progress of the game. The entry N_{ij} (counting i and j from 0) represents the expected number of turns spent in state j given that the player started in state i. Notice that the diagonal entries are all greater than 1. If a player starts the game in state i then at least one roll, the first roll, is in state i. Entries in later states $j > i$ can be less than 1 because it is possible for a player to skip a state entirely. Notice the 6 in the bottom right corner gives the average number of rolls spent in state 9 given that the player started in state 9. This makes sense; in state 9, the player has just one die left to roll. It will take 6 rolls on average to roll the given target number, this is the mean of a geometric random variable with $p = 1/6$. Since players start the game of *Tenzi* in state 0 the first row of matrix N gives the expected number of rolls a player will spend in each state, see Table 1. This makes sense, for state 0, there is always the first roll, but usually the player moves to another state. For states 1 through 4, it is common to get more than one of the target number, so those states often get skipped over. For states 5 to 9, it takes increasingly longer to collect the target number.

The sum of the first row is the mean number of turns to go from state 0 to the absorbing state 10, specifically, 16.564849, agreeing with the approximate mean from the p.d.f. The standard deviation of the number of turns the player will spend

0	1	2	3	4	5	6	7	8	9
6.83	6.81	6.78	6.75	6.70	6.64	6.55	6.41	6.14	5.48

Table 2. Standard deviation of the number of times the chain will be in each state.

Figure 2. Starting configuration and a typical state in the Towers of Hanoi puzzle.

in state i before reaching the absorbing state 10 from state 0 is in Table 2. The scripts provide the calculations for these standard deviations. This also makes sense; in state 9, then a player has just one die left to roll. The entry 5.48 is the approximate standard deviation of a geometric random variable with $p = 1/6$.

7.3. Random Moves in the Towers of Hanoi

The Towers of Hanoi (also called the Tower of Brahma or Lucas' Tower) is a mathematical game or puzzle. It consists of three rods and some number of disks of different sizes, with holes in the center so each disk can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, making a conical shape. The objective of the puzzle is to move the entire stack to another rod, obeying 3 rules.

(1) Only one disk can be moved at a time.

(2) Each move consists of taking the top disk from one of the stacks and placing it on top of another stack or on an empty rod.

(3) No larger disk may be placed on top of a smaller disk.

Consider solving the small three disk case of Lucas' Tower puzzle with random legal moves. The disks start with the biggest disk at the bottom, the intermediate size disk in the middle, and the smallest disk on top. The goal is to move all three disks, one at a time, from one pole to another pole. At no point can a larger disk sit atop a smaller disk. See Figure 2

The minimum number of moves to solve the puzzle is well known: m disks can be solved in exactly $2^m - 1$ moves. The maximum number of legal moves to move all three disks from one pole to any other pole without repeating a configuration is $3^m - 1$. The question here is: on average, how many moves does it take to solve this puzzle with *random* legal moves of three disks?

Label the rods from left to right as a, b, c as in Figure 2. Represent the states for 3 disks as a triplet of values from $\{a, b, c\}$. The first entry in the triplet gets the rod label on which the smallest disk rests, the second entry in the triplet gets the rod label for the middle disk, and the third entry in the triplet gets the rod label on which the largest disk rests. So the start is *aaa*, the goal is either *bbb* or *ccc*. The configuration *aba* implies the smallest disk is on top of the largest disk, while
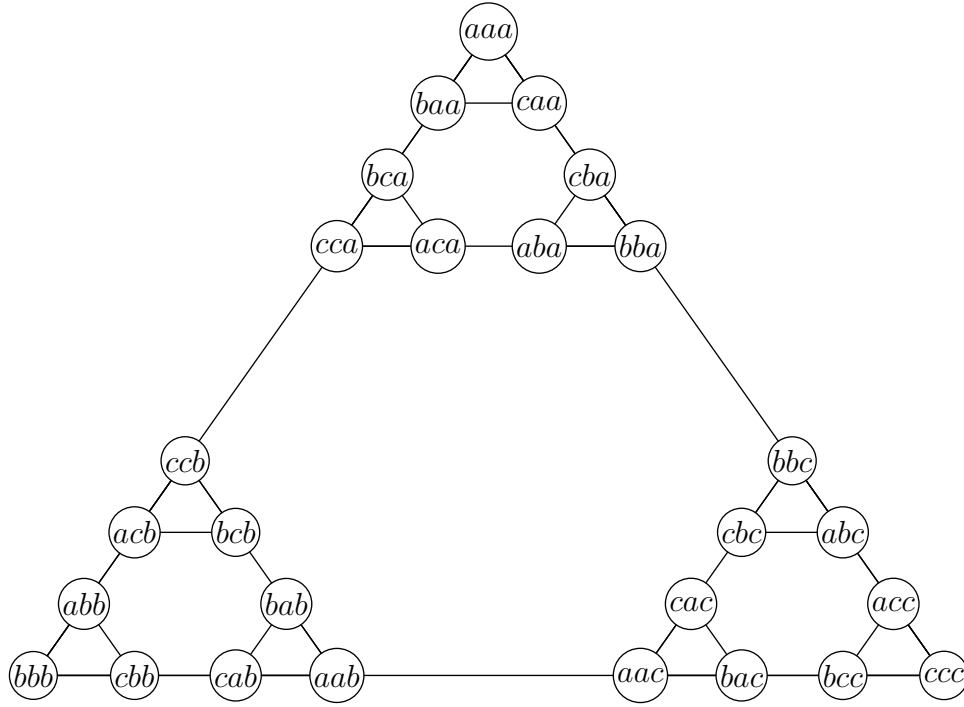
Figure 3. The puzzle graph for the three disk Towers of Hanoi.

the medium disk is on the middle rod labeled b. The puzzle with 3 disks has 27 legal configurations.

The triplet labeling leads to an example of a *puzzle graph*, in which the configurations of a puzzle or game are the nodes of a graph and legal moves between states are the edges of the puzzle graph. Figure 3 is the puzzle graph for the Towers of Hanoi. For example, the edge connecting *aaa* with *baa* represents the initial move of the smallest disk to the middle rod. The puzzle graph for the Towers of Hanoi has a recursive ternary structure useful for analyzing the shortest and longest paths from the start *aaa* to the goals *bbb* or *ccc*. For this small random game just number the states from the source, top to bottom, left to right, as in Figure 3

Selecting each move uniformly randomly from the set of all valid moves from any node to any adjacent node (except for the goal nodes which are absorbing) defines a random walk on the graph or a Markov chain. The two goals, now states 20 and 27, are absorbing states. Most transition probabilities between states are $1/3$, except for the start and end states. The recursive ternary labeling of the nodes is not immediately useful for analysis of this 27-state Markov chain. A script for this Markov chain calculates the mean absorption time as 70.77 moves, see the exercises. T. Berger and M. Alekseyev [**AB15**] give two proofs that the mean time to absorption for the m-disk Tower of Hanoi puzzle with random moves is

$$\frac{(3^m - 1)(5^m - 3^m)}{4 \cdot 3^{m-1}}.$$

One proof uses recurrence relations, the other proof uses an equivalence of Markov chains to electrical networks.

7.4. Chutes and Ladders

The childrens' game *Chutes and Ladders* (known in the United Kingdom as *Snakes and Ladders*) derives from an old Hindu game *Leela* or *Moksha Patam* teaching moral lessons. The game described here is the modified U.S. version. The game is a 100 state absorbing Markov chain and the goal is to investigate the expected playing time.

The game is for two or more players. The game play is on a 10 by 10 board with squares numbered 1 to 100 that are the states of the game. Rather than add an additional state 0 as the starting position, instead make an initial uniform distribution on the 6 possible starting states, see below for details. (The scripts in the R statistical computing language which indexes vectors from 1 in contrast to say, Python, which indexes from 0 motivates this choice of 100 states with an initial distribution. With 101 states including a starting state, the state indices in R scripts would be one greater than the spaces marked on the board, leading to potential confusion.) Since the goal is to find the expected playing time, add 1 to the playing time for the initial move to be in one of the starting states. Players roll a fair six-sided die (or use a fair six-section spinner) to determine how many squares to advance their token on each turn. Two or more players may occupy the same square simultaneously. The first player to reach square 100 exactly wins. In the final 5 states, if a move would take a player beyond the last square, the player does not move. The game board is also marked with chutes and ladders connecting various squares on the board. The special feature of the game is that if a token lands on a square at the bottom of a ladder the token immediately moves to the top of the ladder. Similarly if the token lands on a square at the top of a chute, it immediately moves to the bottom of the chute. The game has 10 chutes and 9 ladders with beginning and end states given by the following ordered pairs

- Ladders: $(1, 38)$, $(4, 14)$, $(9, 31)$, $(21, 42)$, $(28, 84)$, $(36, 44)$, $(51, 67)$, $(71, 91)$, $(80, 100)$;
- Chutes: $(16, 6)$, $(47, 26)$, $(49, 11)$, $(56, 53)$, $(62, 19)$, $(64, 60)$, $(87, 24)$, $(93, 73)$, $(95, 75)$, $(98, 78)$.

This means that 19 states, the beginnings of the chutes and ladders, are inaccessible states. A player is never actually in state 1, for if the token lands on square 1, the ladder $(1, 38)$ immediately deposits the player into state 38. Although the game is a race among two or more players, the players move independently, so it is sufficient to consider just one player moving and ask "How long does it typically take a single player to finish a game by reaching square 100?"

Although the game has 100 squares, it is possible to model the game as a Markov chain with fewer states, see [**AKS93**]. Adding a starting state 0, but deleting each inaccessible state corresponding to a chute or a ladder leaves 82 states. The transition probability matrix is an 82×82 matrix with most rows containing 6 consecutive probabilities of 1/6 each. There are two kinds of exceptions. A row corresponding to any state reachable from a chute or ladder must account for the

long-distance transitions by the chute or ladder. The last 5 rows show the fact that a player must reach state 100 exactly, a die roll causing the player to go past state 100 results in no move.

Here the model uses a 100-state Markov chain with a 100×100 transition probability matrix derived from matrices analogous to those in the Monopoly model. First use a 100×100 circulant (almost, excepting the last 5 rows) dice roll matrix R followed by sparse 100×100 chutes and ladders transition matrices C and L. The matrix L is just the identity matrix but, for instance the 1 in row 1 and column 1, moves to row 1 and column 38 to indicate ladder $(1, 38)$. Likewise for the 18 other ladders and chutes. The result RLC is a 100×100 transition probability matrix with transient states 1 through 99 and absorbing state 100. The game starts in states 38, 2, 3, 14, 5 and 6, each with probability $\frac{1}{6}$. The expected waiting time to absorption is the average over the expected waiting time to absorption from each state, $\frac{1}{6} \left(W(38) + W(2) + W(3) + W(14) + W(5) + W(6) \right) = 38.22512$. Adding 1 for the initial move to the starting states gives 39.22512, agreeing with [**AKS93**]. Using Theorem 3 from Waiting Time to Absorption the standard deviation of the waiting time to absorption is approximately 25.2 turns. As a final note, observe that the waiting time to absorption from state 99 is 6 with standard deviation $\sqrt{30} \approx 5.48$, exactly as expected for a geometric random variable with $p = 1/6$, waiting to roll a 1 to enter the last state.

In [**AKS93**] the authors investigate the sensitivity of the waiting time to adding or removing chutes and ladders. For example, the expected waiting time to absorption for a game with no chutes and ladders is 31.1 moves. See the exercises for more examples.

7.5. Count Your Chickens

Count Your Chickens!, produced by Peaceable Kingdom, is a simple game for preschoolers. In this cooperative game, play begins with forty chicks spread throughout the board and a mama chicken at the start square. The end square is the mama chicken's coop, and the mama chicken moves progressively closer to this square as the game unfolds. The object of the game is to have all 40 chicks in the chicken coop when the mama chicken arrives there. The game board has 41 squares numbered in order, with 30 squares containing one of five characters: a cow, a sheep, a pig, a dog, or a tractor. The remaining 11 squares are empty and neutral. Five of the character squares have a blue background. The final square, the chicken coop, contains all 5 farm characters. Players take turns spinning a spinner with 6 segments of equal area. Five segments match the farm characters from the board and a sixth segment has a fox character. When a player spins the spinner, one of two things can happen. If the player spins one of the five characters drawn on the board, she moves the mama chicken token along the path toward the coop to the next instance of the character. The player counts the squares moved, and adds this number of chicks to the chicken coop. Additionally, if the square landed on has a blue background, the player adds one additional chick to the coop. There are only a total of 40 chicks on the board though, so if the mama chicken moves two squares but there are already 39 chicks in the coop, only one chick gets added. If a fox

0 Start	1	2 Sheep	3 Cow	4	5 Cow	6	7 Sheep	8 Coop Sheep Cow

Figure 4. Smaller game board for the example *Count Your Chickens!*

is spun, the player must remove one chick from the coop and place it back on the board. If there are no chicks in the coop then spinning a fox has no effect.

Suppose the mama chicken is at the start square. If the player spins a sheep, then the mama chicken moves to square 2 (the first sheep) and places two chicks in the coop. If the player next spins a tractor then the mama chicken moves 2 squares to square 4 and places 3 more chicks in the coop because square 4 is blue. If the first spin is a fox then nothing happens because at the start of the game there are no chicks in the coop. The game continues in this manner until the mama chicken reaches the coop. The players win if all 40 chicks are in the coop when the mama chicken arrives there; otherwise the players lose, this is a cooperative game.

The goal here is to answer the following questions:

(1) What is the probability of winning a game of *Count Your Chickens!*? That is, what is the probability of landing on square 40 with 40 chicks in the coop?

(2) What is the expected number of chicks in the coop at the end of the game?

(3) What effect do the blue squares have on the probability of winning the game? For example, the original game has five particular squares colored blue; are there other choices of five blue squares that increase the probability of winning the game?

The mathematical interest here is creating the state space, leading to a large transition probability matrix. A state in *Count Your Chickens!* is an ordered pair (a, b) where a is the mama chicken's position on the board and b is the number of chicks in the coop. For win probabilities only, to minimize the number of states and cut down on computation, include a "loss" state and a "win" state. The idea is that while $(39, 35)$, for example, is a possible state of the game, it is impossible to win from such a configuration. Lump together all such ordered pairs from which it is impossible to win as a loss state. The win state is $(40, 40)$, the mama chicken must reach the last square of the board with all of the chicks in the coop.

The following illustrates construction of a chain from a smaller board, see Figure 4.

This smaller game uses only two farm characters – a sheep and a cow. There are 8 squares on the board after the start square and 8 chicks to return to the coop. The spinner has three equal-sized segments illustrated with a cow, a sheep, and a fox. Number only the squares with a character, since it is not possible to land on a blank square.

A state of the game is an ordered pair (i, j) where i is the position of the mama chicken and j is the number of chicks in the coop. Identify some pairs with the loss state because if j is small enough relative to the mama chicken's position, it will be impossible for all 8 chicks to be in the coop at the end of the game. The

state $(2,0)$ is such an example: if the mama chicken is on square 2 and there are no chicks in the coop, the game will end with at most 7 chicks in the coop. Let b_i^R be the number of blue squares after square i on the path and b_i^L the number of blue squares that come before square i, including square i. Note $b_i^R + b_i^L$ is the total number of blue squares in the game. Then for square i, the states are (i,j) with $\max(i - b_i^R, 0) \le j \le \min(i + b_i^L, 8)$. For example, for square 3, $b_3^R = 1$ and $b_3^L = 1$ since there is one blue square to the right of square 3 and one blue square to the left of square 3. Thus square 3 corresponds to possible states $(3,2)$, $(3,3)$, $(3,4)$. It is not possible to have more than 4 chicks in the coop on square 3, and if the coop has 0 or 1 chicks on square 3 it is impossible to win the game. All the states are

$$\{(0,0), (2,1), (2,2), (2,3), (3,2), (3,3),$$
$$(3,4), (5,5), (5,6), (5,7), (7,7), (7,8), \text{win}, \text{loss}\}$$

for this board. For the purposes of building a transition matrix think of $(0,0)$ as state 1, $(2,1)$ as state 2, $(2,2)$ as state 3, ..., and "loss" as state 14. Entry $p_{i,j}$ in the transition matrix for the game is the probability of moving from state i to state j. For example, in row 1 of the transition matrix, start in position $(0,0)$ on the game board. There is a $1/3$ probability of rolling a fox; this spin keeps the token in position $(0,0)$, so $p_{1,1} = 1/3$. Spinning a sheep is the next possibility, putting the token in position $(2,3)$, or a cow, putting the token in position $(3,3)$. This gives $p_{1,4} = 1/3$ and $p_{1,6} = 1/3$. All other entries in row 1 are 0. Starting in state $(2,1)$, then spinning the sheep puts the token in position $(7,6)$, or the loss state, because if there are only 6 chicks in the coop upon reaching square 7, it is impossible to win. Spinning the fox also puts the game in the loss state, so entry $p_{2,14} = 2/3$. Spinning the cow puts the game in state $(3,2)$, so $p_{2,5} = 1/3$. All other entries in row two are 0. Continuing in this way, the transition matrix for this game is

$$\begin{pmatrix}
\frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2}{3} \\
0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & \frac{2}{3} \\
0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2}{3} & \frac{1}{3} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{2}{3} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}.$$

The state and transition probability matrix models the small game as an absorbing Markov chain. The loss and win states are both absorbing since each of these states transitions to itself with probability 1, and it is possible to reach one of these two states from any other state. The transition matrix is in the canonical

form

$$\begin{pmatrix} Q & R \\ 0 & I \end{pmatrix}$$

where I is the identity matrix. With the transition matrix in canonical form, the fundamental matrix is

$$N = (I - Q)^{-1} = \begin{pmatrix}
\frac{3}{2} & \frac{1}{18} & \frac{1}{6} & \frac{1}{2} & \frac{2}{9} & \frac{11}{18} & \frac{1}{6} & \frac{4}{27} & \frac{2}{9} & \frac{1}{18} & \frac{67}{162} & \frac{17}{54} \\
0 & 1 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{9} & 0 & 0 & \frac{1}{27} & 0 \\
0 & \frac{1}{3} & 1 & 0 & \frac{2}{9} & \frac{1}{3} & 0 & \frac{1}{9} & \frac{1}{9} & 0 & \frac{40}{81} & \frac{1}{27} \\
0 & \frac{1}{9} & \frac{1}{3} & 1 & \frac{1}{9} & \frac{2}{9} & \frac{1}{3} & \frac{2}{27} & \frac{1}{9} & \frac{1}{9} & \frac{31}{81} & \frac{14}{27} \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{9} & 0 \\
0 & 0 & 0 & 0 & \frac{1}{3} & 1 & 0 & \frac{2}{9} & \frac{1}{3} & 0 & \frac{4}{9} & \frac{1}{9} \\
0 & 0 & 0 & 0 & \frac{1}{9} & \frac{1}{3} & 1 & \frac{1}{9} & \frac{2}{9} & \frac{1}{3} & \frac{26}{81} & \frac{14}{27} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \frac{1}{3} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 1 & 0 & \frac{2}{9} & \frac{1}{3} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{9} & \frac{1}{3} & 1 & \frac{5}{27} & \frac{4}{9} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 1
\end{pmatrix}.$$

Let $B = NR$. If the game is in the ith non-absorbing state then the probability that the game terminates by reaching the jth absorbing state is b_{ij}. Then

$$B = \begin{pmatrix}
\frac{305}{486} & \frac{181}{486} \\
\frac{5}{81} & \frac{76}{81} \\
\frac{104}{243} & \frac{139}{243} \\
\frac{170}{243} & \frac{73}{243} \\
\frac{5}{27} & \frac{22}{27} \\
\frac{5}{9} & \frac{4}{9} \\
\frac{190}{243} & \frac{53}{243} \\
\frac{5}{9} & \frac{4}{9} \\
\frac{22}{27} & \frac{5}{27} \\
\frac{73}{81} & \frac{8}{81} \\
\frac{2}{3} & \frac{1}{3} \\
\frac{8}{9} & \frac{1}{9}
\end{pmatrix}.$$

From the first row of B, this game has a win probability of $305/486 \approx 0.6276$ and a loss probability of $181/486 \approx 0.3724$ when a player begins from the start square. One of the nice aspects of using a Markov chain for these calculations is that a player can update the win probability moving through the game. For example, suppose in two turns the player starts by spinning a sheep and then spins a fox. The coop would first gain 3 chicks, then lose a chick. This state of the game corresponds to the state $(2, 2)$, which corresponds to the third row of B. Therefore in such a situation the player would know that her win probability is now only $104/243 \approx 0.4780$.

The computations for the general game are similar except that the matrices Q and R are much larger. Q is 163×163 and R is 163×2 and therefore B is 163×2. It is impractical to enter and compute with such large matrices by hand. The Scripts section has an R program to build the matrices and compute the win probability. The matrix Q in the full game looks similar to the matrix Q in the small example:

the matrix is mostly zeroes punctuated sparsely with small diagonals of entries of value $1/6$, instead of $1/3$. The top left entry of B in this case is 0.6410, and the probability of winning a game of *Count Your Chickens!* is 0.641. The length of the game from the start state to absorption in the win or loss states is 10.83 moves, with a varaince of 4.43 moves.

7.6. Chapter Ending Answer

The future development of the game at a turn must be independent of the past to model the game as a Markov chain. Games depending on a randomizing device, such as a dice rolls or a spinner, can often be modeled as a Markov chain. Examples in this section are the games of *Tenzi*, *Chutes and Ladders*, and *Count Your Chickens!*.

7.7. Chapter Summary

Key Concepts.

(1) *Tenzi* is a simple dice game for multiple players. The gameplay of Tenzi for a single player can be modeled with a Markov chain with 10 transient states and 1 absorbing state. Markov chain methods give the distribution of the number of rolls necessary to complete the game, along with the mean and variance.

(2) The *Towers of Hanoi* is a mathematical game or puzzle. It consists of three rods and some number of disks of different sizes, with holes in the center so each disk can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, making a conical shape. The objective of the puzzle is to move the entire stack to another rod, obeying 3 rules.

(3) The childrens' game *Chutes and Ladders* is a 100 state Markov chain with a final absorbing state and Markov chain methods give the expected playing time.

(4) *Count Your Chickens!* is a cooperative counting game aimed at preschoolers. Absorbing Markov chains give the probability of winning a game of *Count Your Chickens!* and the expected number of chicks at the end of the game. The main interest is classifying and counting the states, leading to a 165×165 transition probability matrix.

Vocabulary.

(1) *Tenzi* is a simple dice game for multiple players.

(2) The *Towers of Hanoi* is a mathematical game or puzzle. It consists of three rods and some number of disks of different sizes, with holes in the center so each disk can slide onto any rod.

(3) The childrens' game *Chutes and Ladders* is a board game that can be modeled as a 100 state Markov chain with a final absorbing state.

(4) *Count Your Chickens!* is a cooperative counting board game aimed at preschoolers.

Notation.

(1) i, j – typical states in a game Markov chain

(2) $P(j)$ – cumulative probability function for a player getting Tenzi on or before the jth roll

(3) $p(j) = P(j) - P(j-1)$ – probability density function for a player getting Tenzi on the kth roll

(4)

$$P = \begin{pmatrix} Q & R \\ 0 & I \end{pmatrix}$$

transition probability matrix is already partitioned

(5) N matrix of expected number of rolls a player will spend in each state

(6) m – number of disks in Tower of Hanoi problem

(7) $aaa, aba, \ldots$ – representation of the states for 3

(8) (a, b) – A state in *Count Your Chickens!* as an ordered pair where a is the mama chicken's position on the board and b is the number of chicks in the coop

(9) b_i^R – the number of blue squares after square i on the path and

(10) b_i^L – the number of blue squares that come before square i, including square i

7.8. Sources

The section on *Tenzi* is adapted from [**BP20**]. The *Towers of Hanoi* example is adapted from https://fivethirtyeight.com/features/can-you-cross-like-a-boss/
Background information is from http://www.cut-the-knot.org/recurrence/hanoi.shtml
Puzzle graphs are examined in more detail in https://www.cut-the-knot.org/do_you_know/graphs2.shtml
The labeling of the states is from https://en.wikipedia.org/wiki/Tower_of_Hanoi
The reference to the Alekseyev-Berger paper is [**AB15**] The section on Chutes and Ladders is adapted from [**AKS93**]. The section on Count Your Chickens is adapted from [**MM19**].

7.9. Outside Readings and Links:

(1)

(2)

(3)

(4)

7.10. Algorithms and Scripts

Algorithm.

Data: States and transition probabilities for Tenzi
Result: Mean and standard deviation of waiting time to win
Result: Plots of p.d.f. and c.d.f. of waiting time

1 *Initialization of Matrices*
2 Load transition probability matrix
3 for $i = 2 : 50$ **do**
4 | Iteratively compute powers of P
5 | Add (1,11) entry to c.d.f. list
6 end
7 Differences of c.d.f. are the p.d.f.
8 Check total probability and approximate mean

9 Extract transient probability submatrix
10 Compute fundamental matrix with matrix inversion
11 Calculate mean from fundamental matrix
12 Calculate variance of waiting times
13 Print mean and standard variation
14 Plot p.d.f. and c.d.f. functions

Algorithm 9: Tenzi Ladders Game Markov Chain.

Data: States and transition probabilities for Chutes and Ladders
Result: Mean and standard deviation of waiting time to win

1 function `rotvec` *(v)*

2 Cyclically rotate vector elements one place to right
3 return rotated vector

4 *Inititialization of Matrices*
5 Load Markov Chain library
6 Set number of states and state names
7 Fill (almost) diceRoll matrix using `rotvec`
8 Fill diagonal Ladder matrix with moves from ladders
9 Fill diagonal Chutes matrix with moves from chutes
10 Transition probability matrix as product of diceRoll, Ladders, Chutes
   Transition matrix is product of dice roll, ladders and chutes
11 *Markov chain calculations*
12 Create Chutes and Ladders Markov chain object
13 Compute mean absorption time with library function
14 Extract transient probability submatrix
15 Compute fundamental matrix with matrix inversion
16 Calculate mean from fundamental matrix
17 Calculate variance of waiting times
18 Print mean and standard variation

Algorithm 10: Chutes and Ladders Game Markov Chain.

Scripts.

R: R script for Tenzi.

```r
x   <-   c(0.1615055828898457 , 0.3230111657796915 , 0.2907100492017223 ,
           0.1550453595742519 , 0.05426587585098816 , 0.01302381020423716 ,
           0.002170635034039527 , 2.48072575318803*10^-4 ,
           1.860544314891023*10^-5 , 8.2690858439601*10^-7 ,
           1.65381716879202*10^-8 , 0.0 , 0.1938066994678149 ,
           0.3488520590420667 , 0.2790816472336534 , 0.1302381020423716 ,
           0.03907143061271148 , 0.007814286122542296 ,
           0.001041904816338973 , 8.930612711476907*10^-5 ,
           4.465306355738455*10^-6 , 9.922903012752121*10^-8 , 0.0 , 0.0 ,
           0.2325680393613778 , 0.3721088629782046 , 0.2604762040847432 ,
           0.1041904816338973 , 0.02604762040847432 , 0.004167619265355891 ,
           4.167619265355891*10^-4 , 2.381496723060509*10^-5 ,
           5.953741807651273*10^-7 ,0.0 , 0.0 , 0.0 , 0.2790816472336534 ,
           0.3907143061271148 , 0.2344285836762689 , 0.07814286122542295 ,
           0.01562857224508459 , 0.001875428669410151 ,
           1.250285779606767*10^-4 , 3.572245084590763*10^-6 ,0.0 , 0.0 ,
           0.0 , 0.0 , 0.3348979766803841 , 0.4018775720164609 ,
           0.2009387860082305 , 0.05358367626886145 , 0.008037551440329218 ,
           6.430041152263375*10^-4 , 2.143347050754458*10^-5 ,0.0 , 0.0 ,
           0.0 , 0.0 , 0.0 , 0.4018775720164609 , 0.4018775720164609 ,
           0.1607510288065844 , 0.03215020576131687 , 0.003215020576131687 ,
           1.286008230452675*10^-4 ,0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,
           0.4822530864197531 , 0.3858024691358025 , 0.1157407407407407 ,
           0.0154320987654321 , 7.716049382716049*10^-4 ,0.0 , 0.0 , 0.0 ,
           0.0 , 0.0 , 0.0 , 0.0 , 0.5787037037037037 , 0.3472222222222222 ,
           0.06944444444444445 , 0.004629629629629629 ,0.0 , 0.0 , 0.0 , 0.0 ,
           0.0 , 0.0 , 0.0 , 0.0 , 0.6944444444444444 , 0.2777777777777778 ,
           0.02777777777777778 ,0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 ,
           0.0 , 0.8333333333333334 , 0.1666666666666667 ,0.0 , 0.0 , 0.0 ,
           0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 1.0)
P   <-   t( matrix(x, 11, 11) )

PPow   <- P
tenzicdf   <- c(PPow[1,11])
for (i in 2:50) {
  PPow   <- PPow %*% P;
  tenzicdf   <- c(tenzicdf , PPow[1,11])
}

tenzipdf   <-   tenzicdf[2:50] - tenzicdf[1:49]
tenzipdf   <- c(tenzicdf[1], tenzipdf)

sum(tenzipdf)
tenzimean   <- (1:50) %*% tenzipdf

Q   <-   P[1:10, 1:10]
I10   <-   diag(10)
N   <- solve( I10 - Q, I10)

mean   <- N %*% rep(1,10)
Varw   <- ((2 * N - I10) %*% N) %*% rep(1, 10) - ( N %*% rep(1, 10) )^2
sdw   <- sqrt(Varw)
cat("Mean waiting time: ", mean[1], "\nSD of waiting time:", sdw[1], "\n");

plot(x = 1:50, tenzicdf, col="red", xlab="Rolls", ylab="Probability")
points(1:50, tenzicdf, type="l", col="red")
points(1:50, tenzipdf, col="blue")
points(1:50, tenzipdf, type="l", col="blue")
```

.

R: R script for Chutes and Ladders.

```r
library("markovchain")

rotvec <- function(vec) vec[ c(length(vec), 1:(length(vec)-1)) ]
```

```
 4
 5  nStates <- 100
 6  chutesladdersStates <- as.character( c(1:nStates) )
 7
 8  diceRoll <- matrix(0, nStates, nStates)
 9  p1 <- c(0,
10          1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
11          numeric(nStates-7)
12          )
13  for (i in 1:(nStates-5)) {
14      diceRoll[i, ] <- p1
15      p1 <- rotvec(p1)
16  }
17  diceRoll[100, 100] <- 1
18
19  diceRoll[99, 99] <- 5/6; diceRoll[99, 100] <- 1/6
20
21  diceRoll[98, 98] <- 4/6; diceRoll[98, 99] <- 1/6; diceRoll[98, 100] <- 1/6
22
23  diceRoll[97, 97] <- 3/6; diceRoll[97, 98] <- 1/6
24  diceRoll[97, 99] <- 1/6; diceRoll[97, 100] <- 1/6
25
26  diceRoll[96, 96] <- 2/6; diceRoll[96, 97] <- 1/6
27  diceRoll[96, 98] <- 1/6; diceRoll[96, 99] <- 1/6; diceRoll[96, 100] <- 1/6
28
29  ladders <- diag(nStates)
30  ladderIn <-  c( 1,  4,  9, 21, 28, 36, 51, 71,  80)
31  ladderOut <- c(38, 14, 31, 42, 84, 44, 67, 91, 100)
32  ladders[ cbind(ladderIn, ladderIn) ] <- 0
33  ladders[ cbind(ladderIn, ladderOut) ] <- 1
34
35  chutes <- diag(nStates)
36  chuteIn <-  c(16, 47, 49, 56, 62, 64, 87, 93, 95, 98)
37  chuteOut <- c( 6, 26, 11, 53, 19, 60, 24, 73, 75, 78)
38  chutes[ cbind(chuteIn, chuteIn) ] <- 0
39  chutes[ cbind(chuteIn, chuteOut) ] <- 1
40
41  chutesladdersMat <-  diceRoll %*% ladders %*% chutes
42
43  chutesladders <- new("markovchain", states = chutesladdersStates, byrow =
        TRUE,
44                       transitionMatrix = chutesladdersMat, name = "
        ChutesLadders")
45  meanMC  <- meanAbsorptionTime(object=chutesladders)
46
47  Q   <- chutesladdersMat[1:99, 1:99]
48  I99  <-  diag(99)
49  N   <- solve( I99 - Q, I99)
50  absorptionTimeMat   <-  N %*% rep(1, 99)
51  meanMat  <- mean( absorptionTimeMat[ c(38, 2, 3, 14, 5, 6)] )
52  varAbsorptionTimeMat  <- ((2 * N - I99) %*% N) %*% rep(1, 99) - ( N %*% rep
        (1, 99) )^2
53  sdAbsorptionTime  <-  sqrt( mean( varAbsorptionTimeMat[ c(38, 2, 3, 14, 5,
        6) ] ) )
54
55  cat("Mean waiting time from markovchain library: ", meanMC[1],
56  "\nMean waiting time from fundamental matrix: ", meanMat + 1,
57  "\nSD of waiting time:", sdAbsorptionTime[1], "\n");
```

·

R: R script for *Count Your Chickens!*.

```
 1
 2  boardLength <- 40
 3
 4  animalList <- c("C", "D", "P", "S", "T")
 5  NAnimals <- length(animalList) + 1  # one more for fox
 6
 7  blueSquares <- numeric(boardLength)
```

```
 8 blueSquares[4] <- 1
 9 blueSquares[8] <- 1
10 blueSquares[22] <- 1
11 blueSquares[35] <- 1
12 blueSquares[39] <- 1
13
14 totalNumberOfBlues <- sum(blueSquares)
15
16 # Setting up the board
17 board <- character(boardLength)
18
19 board[1] <- "B"
20 board[2] <- "S"
21 board[3] <- "P"
22 board[4] <- "T"
23 board[5] <- "C"
24 board[6] <- "D"
25 board[7] <- "P"
26 board[8] <- "C"
27 board[9] <- "D"
28 board[10] <- "S"
29 board[11] <- "T"
30 board[12] <- "B"
31 board[13] <- "C"
32 board[14] <- "P"
33 board[15] <- "B"
34 board[16] <- "B"
35 board[17] <- "B"
36 board[18] <- "T"
37 board[19] <- "B"
38 board[20] <- "T"
39 board[21] <- "D"
40 board[22] <- "S"
41 board[23] <- "C"
42 board[24] <- "D"
43 board[25] <- "P"
44 board[26] <- "T"
45 board[27] <- "B"
46 board[28] <- "S"
47 board[29] <- "C"
48 board[30] <- "B"
49 board[31] <- "B"
50 board[32] <- "T"
51 board[33] <- "P"
52 board[34] <- "S"
53 board[35] <- "D"
54 board[36] <- "B"
55 board[37] <- "S"
56 board[38] <- "C"
57 board[39] <- "P"
58 board[40] <- "CDPST"
59
60 # Count how many states
61 statesIndex <- 1  #initial state (0,0)
62 statesCount <- 1
63 statesHolder <- matrix(0, boardLength * totalNumberOfBlues, 2)
64
65 for (i in 1:(boardLength - 1)) {
66     # don't use last space, the 'coop'
67     if (board[i] != "B") {
68         bLi <- sum(blueSquares[1:i])  #blueSquares to Left of i, including
       i
69         bRi <- sum(blueSquares[(i + 1):boardLength])  #blueSquares to Right
        of i
70
71         maxChickens <- min(boardLength, i + bLi)
72         minChickens <- max(0, i - bRi)
73
74         statesCount <- statesCount + maxChickens - minChickens + 1
75         for (j in minChickens:maxChickens) {
76             statesIndex <- statesIndex + 1
```

```r
77               statesHolder[statesIndex, ] <- c(i, j)
78           }
79       }
80  }
81  statesHolder[statesCount + 1, ] <- c(Inf, Inf)   # win state
82  statesHolder[statesCount + 2, ] <- c(-Inf, -Inf)  # lose state
83  statesCount <- statesCount + 2  #add final won and loss states
84  states <- statesHolder[1:statesCount, ]
85  rm(statesHolder)
86
87  Q <- matrix(0, statesCount - 2, statesCount - 2)
88  R <- matrix(0, statesCount - 2, 2)
89  I <- diag(2)
90  Z <- matrix(0, 2, statesCount - 2)
91
92  for (i in 1:(statesCount - 2)) {
93      currentSquare <- states[i, 1]
94      chicksInCoop <- states[i, 2]
95      currentState <- i
96
97      # This sets transition probabilities from spinning a fox
98      if (chicksInCoop == 0) {
99          Q[i, i] <- 1/NAnimals  # stay in current state, prob 1/NAnimals
100     } else if (states[i - 1, 1] == currentSquare) {
101         Q[i, i - 1] <- 1/NAnimals  # remove 1 chick, fill trans prob
102     } else {
103         # too few chicks, must lose
104         R[i, 2] <- R[i, 2] + 1/NAnimals  # so add to lose probability
105     }
106
107     # Now set transition probabilities from other five animals.
108     for (animal in animalList) {
109
110         nextSquare <- grep(animal, board[(currentSquare + 1):boardLength])
        [1] + currentSquare
111
112         if (blueSquares[nextSquare] == 0) {
113             # not Blue, just add advance
114             updatedChicksInCoop <- chicksInCoop + (nextSquare -
        currentSquare)
115             if (updatedChicksInCoop > boardLength)
116                 updatedChicksInCoop <- boardLength
117         } else {
118             # Blue square, so add extra chick
119             updatedChicksInCoop <- chicksInCoop + (nextSquare -
        currentSquare) +
120                 1
121             if (updatedChicksInCoop > boardLength)
122                 updatedChicksInCoop <- boardLength
123         }
124
125         in_states <- FALSE
126
127         for (p in 1:statesCount) {
128             if (states[p, 1] == nextSquare && states[p, 2] ==
        updatedChicksInCoop) {
129                 in_states <- TRUE
130                 newLocation <- p
131                 break
132             }
133         }
134
135         if (in_states == TRUE) {
136             Q[i, newLocation] <- 1/NAnimals
137         } else {
138             if (nextSquare == boardLength && updatedChicksInCoop ==
        boardLength)
139                 {
140                     R[i, 1] <- R[i, 1] + 1/NAnimals
141                 }  # win
142  else {
```

```
143                    R[i, 2] <- R[i, 2] + 1/NAnimals   #lose
144               }
145          }
146      }
147 }
148
149 # Calculate win-lose probabilities with fundamental matrix: (I-Q)^{-1} R
150 IsC <- diag(statesCount - 2)
151 matrixProbs <- solve(IsC - Q, R)
```

.

7.11. Problems to Work for Understanding

1: Write a script in some computer language to replicate the calculations and graphs for the Markov chain for *Tenzi*.

2: The c.d.f. and p.d.f. functions $P(j)$ and $p(j)$ above respectively summarize the probability that a single player will get Tenzi in j rolls. What if n people are playing? This essentially taking n independent random samples from the probability distribution $p(j)$, and identifying the minimum value(s) of j as the roll in which the game is won. For example, if there are $n = 5$ players, take 5 independent random samples from $p(j)$. Suppose the arbitrary values of j are $[12, 20, 15, 9, 28]$, representing the number of rolls needed for each player to get Tenzi. Then the 4th player wins the game in 9 rolls.

(1) In general, write an expression for the probability that the minimum value from n random samples of $p(j)$ is less than or equal to j in terms of $P(j)$.

(2) Using the probability distributions from the scripts find the expected number of turns in a game of Tenzi with 2, 3 and 4 players/

3: Show that the shortest possible game of *Chutes and Ladders* consists of 7 moves by listing those moves.

4: Consider the effect of chutes and ladders on the waiting time to absorption.

(a) Find the expected waiting time to absorption in a game with no chutes or ladders.

(b) In the original game, insert an additional ladder from state 46 to state 94. What happens to the expected waiting time?

(c) In the original game, instead insert a chute from state 83 to state 7. What happens to the expected waiting time?

(d) Add a ladder to the original game from state 79 to state 81. What happens to the expected waiting time?

(e) Add a chute to the original game from state 29 to state 27. What happens to the expected waiting time?

5: Investigate the following modifications to the small *Count Your Chickens!* game.

(a) What is the win probability if only square 2 is blue? What is the win probability if squares 2, 3 and 5 are blue?

(b) What effect does adding additional animals to the board (i.e., increasing the size of the spinner) have?

(c) What if the fox removes more than one chick from the coop?

(d) What effect does changing the size of each character's space in the spinner do? For example, what if the fox covered $1/3$ of the spinner and the remaining five animals each took up $2/15$ of the spinner?

(e) What if some of the blue squares were colored green and had the effect of moving an additional two chicks into the coop?

(f) Create your own questions that can be analyzed using Markov chains and to modify the supplemental R code to answer the questions.

A Markov Chain Model of Baseball

This chapter assumes familiarity with the rules and terminology of American baseball and some of the statistics associated with baseball players. The goal is to model a half-inning of a baseball game as a relatively simple Markov chain to find the expected number of runs a player will score as a means of evaluating and comparing players. A secondary goal is to propose a simple Markov chain as a basis for more detailed and realistic models of the game of baseball.

Mathematically Mature: may contain mathematics beyond calculus with proofs.

8.1. Chapter Starter Question

What is the baseball statistic known as *OPS*? How is this statistic used?

8.2. States and Transition Probability Matrix

The Markov chain will be the sequence of situations arising in a half-inning of a baseball game. In this Markov chain model of baseball each step will be a plate appearance. A plate appearance is similar to an at-bat, but includes walks, hit batters, and sacrifices that recorded baseball does not regard as an official at-bat. In this baseball model, the states are the runners on base together with a count of the present outs. Each plate appearance has eight possibilities for the runners on base:

- the bases can be empty;
- there can be one runner on first, second, or third base;
- there can be two runners on base in three different ways; or
- the bases can be loaded.

Runners:	None	1st	2nd	3rd	1&2	1&3	2&3	1,2,&3
Outs								
0	1	4	7	10	13	16	19	22
1	2	5	8	11	14	17	20	23
2	3	6	9	12	15	18	21	24
3	25							

Table 1. States of the Markov chain model of a half inning of baseball
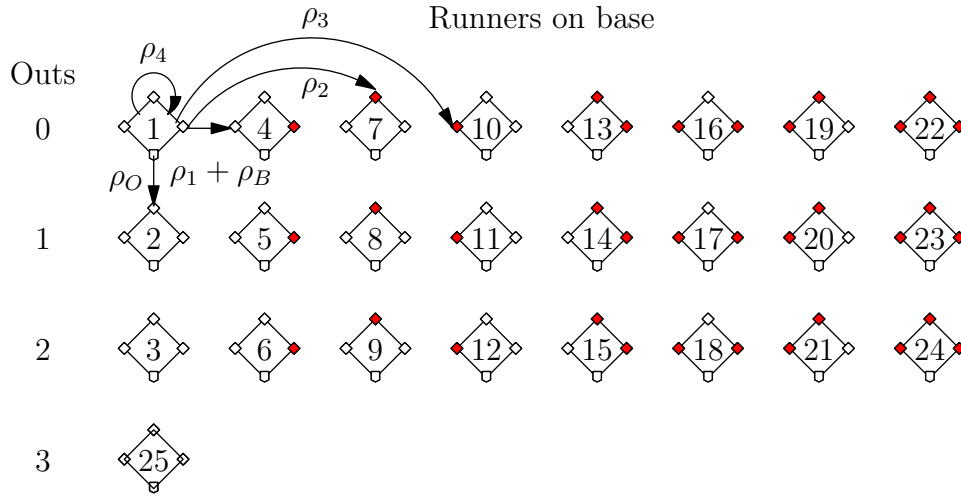
Figure 1. The state diagram for the Markov chain model of a half-inning of baseball, with only transitions from state 1 indicated as an example.

The number of outs can be 0, 1, or 2 to make twenty-four states. All situations with three outs are the same, and this situation is the final 25th absorbing state. Once the game reaches this state, the half-inning ends and the Markov chain is stopped. This simplified model takes no account of how many players are on base or where players are located when the third out is made. Table 1 below gives the labeling of the 24 possible runners and outs states. Figure 1 is a diagram of the states along with a few sample transitions. The model tracks runs scored separately, runs are not part of the states. Osbourne *et al.* [**OWD20**], use the same set of states with more descriptive state labels but ordering them left to right in a row, then row by row.

Denote the probability of moving from one state i to state j by $p_{i,j}$. Many transitions are impossible, such as a transition to a state with fewer outs, and hence have probabilities equal to zero. Using the numbering above, $p_{1,4}$ is the probability of going from no outs and no runners on base to a runner on first with no outs, which in turn is equal to the probability of a single plus the probability of a walk. In real baseball, there are other possibilities for this transition such as a wild pitch, but this simplified model ignores them. In certain situations different actions by the batter produce the same effect. As can be seen in Figure 1, if no one

is on base, then a single and a walk have the same effect. Sum the probability of each event to give the state transition probability.

For this model imagine that the player bats with plate outcomes distributed randomly according to his statistical profile. Plate appearances are independent. In other words, a player's chance of getting a specific kind of hit or making a specific kind of out is independent of how many runners are on and the number of outs. Based on this profile, compute the transition probabilities for each pair of states, obtaining the transition matrix. The transition probabilities could come from player statistics from the previous year, the current year, or the career, but must use the same profile each time. The transition probability matrix P, whose entries are $p_{i,j}$ governs the evolution of the half inning. By the properties of matrix multiplication, P^2 is the transition matrix for sequences of two plays or batters, P^3 is the transition matrix for sequences of three batters, and so on, for a string of identical batters. This assumption is useful for computing *average* performance which is the goal here, but not for specific game situations.

8.3. Assumptions of the Model

Encoding the half inning of the game in the transition matrix like this requires simplifying assumptions.

(1) The Markov property assumption means that this model does not take into account how it arrived at a particular situation. For example, if there is a runner on first and no outs, the Markov chain model is not concerned about whether there was a walk or a single.

(2) More specifically, the Markov property assumption deliberately eliminates the possibility of a *clutch hitter*. Such a player allegedly hits better when the situation matters more, but numerical studies indicate that the concept is an illusion.

(3) A related question is whether players get "hot" or "cold", the model ignores this possibility.

(4) This model ignores situations such as stolen bases or pick-offs that could occur during the plate appearance, changing the state without action from the player in his plate appearance.

(5) The model ignores double plays, sacrifice bunts, and sacrifice flies.

(6) The model ignores runs scored while incurring the third out.

(7) An important simplifying assumption is that the matrix is time independent, and each plate appearance is an independent event. In real baseball, as pitchers and batters face each other multiple times, each adjusts to the other, changing the probabilities. Fatigue may also be a factor as a game progresses.

(8) Another related assumption is that the batter's outcomes do not depend on the pitcher and what the pitcher does is not dependent on the hitter. The probabilities remain constant, regardless of the current match-up and previous encounters between pitcher and hitter.

(9) The model assumes that a single advances runners already on base by two bases. For example, in states 4, 5, or 6 with runner on first, a single by the

hitter moves the base runner to third base, leaving the chain in states 16, 17, or 18 respectively. Also, assume that a double advances each runner two bases, so that in states 4, 5, or 6 with runner on first, a double by the hitter moves the base runner to third base, leaving the chain in states 19, 20, or 21. In contrast, in [**CK77**] Cover and Keilers explicitly make the assumption that "All singles and doubles are assumed to be long. That is, a single advances a baserunner two bases, and a double scores a runner from first base." On the other hand, Pankin [**Pan**] is not explicit about this assumption, but Markov chain simulations imply that more detailed base running plays have nonzero probability. For instance, Pankin implies the possibility that bases loaded with one out (state 23) can change to state 5 or 8 meaning that a runner on first, as well as the two runners on second and third, can score on a single or a double. Likewise Pankin implies that a change from state 23 to 23 is possible which would be a single allowing one run from third, advancing the runners on first and second one base. As another example, Osbourne *et al.* [**OWD20**] show how to estimate the probabilities that with no outs and a baserunner on first, after a single the baserunner advances to second or third, or even scores all the way from first. The simpler Markov chain model here does not assume these additional baserunning and scoring refinements.

(10) Bukiet *et al.* [**BHP97**] present an algorithm to carry Markov calculations for one inning forward across a full game, keeping track of the probability distribution of runs scored, instead of just a half-inning considered here.

Each assumption can be relaxed at the cost of more states and a larger transition probability matrix, or using more detailed player statistics or both. For example, Osbourne *et al.* [**OWD20**] show how to estimate the probability of a sacrifice fly or grounding into a double play. However they specifically neglect events such as a four-base error in transition $P_{1,1}$, or reaching base by error, or by hit-by-pitch, also ignored here. The intent here is to show how to model baseball as a relatively simple Markov chain, in the process creating a new player evaluation statistic.

8.4. Player Statistics

A particular batter H determines a twenty-five by twenty-five matrix M_H from his statistical profile. For convenience, write the player's profile as follows, deviating slightly from the standard listing. Here AB denotes at-bats, S denotes singles, D denotes doubles, T denotes triples, HR denotes home runs, BB denotes walks plus hit batters, AVE denotes batting average $(S+D+T+HR)/AB$, SA denotes slugging average $(1 \cdot S + 2 \cdot D + 3 \cdot T + 4 \cdot HR)/AB$, a measure of the batting productivity of a player, OB denotes on-base average $(BB + S + D + T + HR)/(AB + BB)$, and $OPS = OB + SA$ measuring the ability of a player both to get on base and to hit for power, two important offensive skills. Note that the last four elements depend on the first six, and hence are not needed. For this model, the number of plate appearances is $AB + BB$, ignoring sacrifice flies as in the official score-keeping. In this simplified scenario, all outs are equivalent, there is no practical difference between a strikeout and a pop-out. As a result, values of OB and OPS here differ slightly from the official records. These statistics give the complete profile of H for

	AB	S	D	T	HR	BB	AVE	SA	OB	OPS
X	500	100	25	5	30	100	.320	.570	.433	1.003
MT	470	63	27	2	45	110	,291	.645	.426	1.071
CB	558	86	34	3	47	95	.305	.629	.406	1.035

Table 2. Statistics for representative players.

the model. In other words, the player statistics determine the probabilities that H makes each kind of hit or out, draws a walk, gets hit by a pitch, and so on.

For example, Table 2 gives a fictional line for a hypothetical star player X. Table 2 also includes 2019 season profiles for Mike Trout (Los Angeles Angels, American League MVP for 2019) and Cody Bellinger (Los Angeles Dodgers, National League MVP for 2019). Player X has 500 at-bats but 600 plate appearances. In a given plate appearance player X has a $\frac{1}{6}$ chance to hit a single, a $\frac{1}{6}$ chance to reach first base by a walk or hit batter, a $\frac{1}{120}$ chance to hit a triple, and so on. In this way, from the 6 basic player statistics, compute a probability distribution $(\rho_O, \rho_B, \rho_1, \rho_2, \rho_3, \rho_4)$, where ρ_O is the probability of an out, ρ_B is the probability of a base on balls, and ρ_i is the probability of an i-base hit.

8.5. Constructing the Transition Probability Matrix

A state can change in several ways, for instance state 16 with runners on first and third with no outs can change into state

- 17 with an out,
- 22 with a single advancing the runner on first,
- 16 back to itself while scoring a run with a single,
- 19 with a double scoring both base runners,
- 10 with a triple, or
- 1 with a home run.

A convenient way to build the transition probability matrix is to compose it from 6 simpler matrices with entries either 0 or ρ_x, corresponding to the 6 probabilities in the hitter's distribution. For example, for ρ_O, the probability of an out, the number of outs increases by one and the base runners remain the same, which makes a regularly shaped transition matrix P_0, where states with an index divisible by 3 transition to the absorbing state 25 of three outs. All other states advance by 1 index with probability ρ_O. Taking P_3 as another example, a triple scores all runners already on base, leaves the hitter on third base and the numbers of outs stays the same. Thus in P_3 each state transitions to state 10, 11 or 12 according to the existing number of outs. The other 4 matrices are similar, but the advance of states is slightly more complicated because of the assumption that runners on base advance by 2 bases on a hit. Finally, the full probability transition matrix is the sum of the 6 elementary transition matrices: $P = P_O + P_B + P_1 + P_2 + P_3 + P_4$. Set $P_{25,25} = 1$ to account for the absorbing end state. The transition probability

matrices are too large to display here, but the scripts below build the matrices given the values for $(\rho_O, \rho_B, \rho_1, \rho_2, \rho_3, \rho_4)$.

Keeping track of runs scored will be an important outcome of the model. The new statistic called *Markov runs* is how many runs a team could score by using this player for every plate appearance. The complete definition follows. Cover and Keilers [**CK77**] call this the *Offensive Earned-Run Average* or OREA but that term seems to be no longer used. This statistic, normalized using nine innings, for all players each day of the season is available at Sagarin ratings labeled as "markov RPG" (Runs Per Game). Given the player's statistical profile, the player bats according to the Markov chain model until making three outs. In principle the test is run thousands of times, and from it the model determines the average number of runs scored per nine innings. However, simple results from Markov chain theory calculate the statistics without needing thousands of simulations.

8.6. Calculating Markov Runs

Let R be the 25×1 column vector containing the expected or average runs scored after each state on one play only. Denoting the elements of R by R_1, R_2, ..., R_{25} some example calculations are:

$$R_1 = p_{1,1}(\text{ based on a home run})$$
$$R_2 = p_{2,2}(\text{ based on a home run})$$
$$R_6 = p_{6,12} + 2p_{6,3}(\text{ based on triple or a home run})$$
$$R_{17} = p_{17,17} + p_{17,20} + 2p_{17,11} + 3p_{17,1}$$
$$R_{23} = 2p_{23,17} + 2p_{23,20} + 3p_{23,11} + 4p_{23,2}$$
$$R_{25} = 0.$$

The vector of runs calculated here will be componentwise slightly less than the run vector for Cover and Keilers because advances on singles and doubles are less than they assume. Likewise, the vector of runs calculated here will be componentwise less than the run vector for Pankin because some state transitions he considers are ignored here.

The key output of the Markov chain baseball model is the computation of the expected runs in the rest of the inning after any runners and outs state. Let E be the 25×1 column vector containing these values. Then,

$$E = R + PR + P^2R + P^3R + \cdots .$$

This equation says that the expected runs after any state is the sum of the expected runs after one plate appearance, the expected runs after two plate appearances, and so on theoretically forever. Of course, almost surely (in the probability sense!) the half-inning will end sometime in the 25th absorption state of three outs. Because the vector of runs calculated here is less than the other run vectors, the expected run vector, or Markov runs will be slightly less. Since the 25th state with 3 outs is the only absorbing state and the other 24 states are transient, the 25×25 probability transition matrix partitions into the block matrix

$$P = \begin{pmatrix} Q & Q' \\ 0 & 1 \end{pmatrix}$$

where Q is 24×24, and Q' is 24×1. This block matrix has the typical form for games because it is natural to order the absorbing state as the last state 25. Since no runs are scored from the 25th absorbing state, just consider

$$E' = R' + QR' + Q^2 R' + Q^3 R' + \cdots$$

where R' and E' are the 24×1 subvectors of E and R corresponding to the first 24 entries. By standard matrix theory, this is equal to

$$E' = (I_{24} - Q)^{-1} R'.$$

The expected runs vector E' has useful interpretations. For example, if P uses statistics for all events for a league, then E' contains the league average expected runs scored in the rest of the inning from each runners-and-outs state. With enough statistics, similar values for teams, a team's home and road games, or for an individual player can be calculated. The last case is the per inning estimate of run scoring if that player batted all the time. Also, restricting the transitions to those not influenced by strategies such as stolen bases or sacrifice bunts allows comparison of expected run values with games using those strategies.

Note that E'_1 is the expected runs after the no runners, no out state in which all innings begin. Thus **Markov runs per game** $9E'_1$ is the expected number of runs per 9 half-innings, or per game. This computation is especially interesting when applied to an individual player's statistics. The value for the mythical player X is 9.49. This statistics provides another way of estimating how many runs per game a player would score by batting all the time. As noted above, some baseball statisticians report this number for players, but much more common is the **On-base plus slugging** statistic or OPS. For OPS, simply add the player's slugging average to his on-base average. This simply computed player statistic empirically correlates fairly well with the expected number of runs scored using the Markov chain model. More details are in the article by D'Angelo [**D'A10**].

8.7. Changes to the Model

The elementary model here assumes a series of identical batters, of course this does not happen in real games. This has been the major criticism of the applicability of Markov models. But it is possible to have a Markov model with different transition matrices for each batter, see [**OWD20**] for an example. Then the assumption of stationarity is dropped and the transition probability matrix changes with each batter according to that batter's statistics. Instead of powers of one transition matrix P, use products of the matrices for each batter: $P_1 P_2$, $P_1 P_2 P_3$, etc. Also, the expected runs on the next play column vector R has to be modified for each batter. With such changes, the expected runs generalizes to

$$E = R_1 + P_1 R_2 + P_1 P_2 R_3 + P_1 P_2 P_3 R_4 + \cdots .$$

An additional potential complication is that it may be necessary to repeat the calculation for several sequences with different first batters and then weight the results by the probability of specific batters beginning the sequence. The model can be expanded in a number of ways and the utility is limited only by the amount of data available.

8.8. Chapter Ending Answer

OPS is the sum of "on-base percentage" and "slugging percentage" to get one number that unites the two. OPS is meant to combine how well a hitter can reach base with how well he can hit for average and for power. "On-base percentage" refers to how frequently a batter reaches base per plate appearance. Times on base include hits, walks and hit-by-pitches, but do not include errors, times reached on a fielder's choice or a dropped third strike. "Slugging percentage" represents the total number of bases a player records per at-bat.

8.9. Chapter Summary

Key Concepts.

(1) The goal is to model a half-inning of a baseball game as a Markov chain to find the expected number of runs a player will score as a means of evaluating and comparing players.

(2) The expected number of runs a player will score uses the standard partition of the transition probability matrix over the transient and absorbing states.

(3) A convenient way to build the transition probability matrix is to compose it from 6 simpler matrices with entries either 0 or ρ_x, corresponding to the 6 probabilities in the hitter's distribution.

Vocabulary.

(1) A new baseball statistic is how many runs will a team score if it uses a player for every plate appearance, called **Markov runs**.

(2) The **On-base plus slugging** statistic or OPS is the sum of a player's slugging average and his on-base average.

Notation.

(1) $p_{i,j}$ – the probability of moving from one state i to state j

(2) AB denotes at-bats, S denotes singles, D denotes doubles, T denotes triples, HR denotes home runs, BB denotes walks plus hit batters, AVE denotes batting average $(S + D + T + HR)/AB$, SA denotes slugging average $(1 \cdot S + 2 \cdot D + 3 \cdot T + 4 \cdot HR)/AB$, a measure of the batting productivity of a player, OB denotes on-base average $(BB + S + D + T + HR)/(AB + BB)$, and $OPS = OB + SA$ measuring the ability of a player both to get on base and to hit for power

(3) $(\rho_O, \rho_B, \rho_1, \rho_2, \rho_3\rho_4)$ – probability distribution where ρ_O is the probability of an out, ρ_B is the probability of a base on balls, and ρ_i is the probability of an i-base hit.

(4) R – the 25×1 column vector containing the expected or average runs scored after each state on one play only. The elements of R are $R_1, R_2, \ldots, R_{25}$

(5) E – the expected runs in the rest of the inning after any runners and outs state

(6) Q, Q' – submatrices of the transition probability matrix corresponding to transitions among the transients and the transients to the stationary state. E' corresponding expected runs subvector.

(7) I_{24} – 24×24 identity matrix.

8.10. Sources

This section is adapted from the articles [**CK77**], [**D'A10**], [**OWD20**] and [**Pan**]. Each of those articles has many further references. The online article The Markov Chain Model of Baseball has the same basic information as this article, but in a condensed form. The definition of OPS is adapted from On-base Plus Slugging

8.11. Algorithms and Scripts

Data: Basic player statistics
Result: Markov runs from each state for the player

Algorithm.

1 *Initialization*
2 Enter player statistics
3 Initialize with zero simple matrices for the player statistics
4 Fill simple matrices with corresponding player statistic
5 Transition probability matrix is sum of simple matrices
6 Create runs scored matrix
7 Expected runs as product of runs scored matrix with player statistics

8 *Calculation of Markov runs*
9 Direct matrix calculation of Markov runs from submatrix
10 Load markovchain library and initialize
11 Calculate Markov runs as expectedRewards()
12 **return** *Markov runs from each state*

Algorithm 11: Markov chain simulation.

Scripts.

```
1  # Player X Statistics
2  rho0 <- 340/(500 + 100)
3  rhoB <- 100/(500 + 100)
4  rho1 <- 100/(500 + 100)
5  rho2 <- 25/(500 + 100)
6  rho3 <- 5/(500 + 100)
7  rho4 <- 30/(500 + 100)
8
9  P0 <- matrix(0, nrow=25, ncol=25)
10 PB <- matrix(0, nrow=25, ncol=25)
11 P1 <- matrix(0, nrow=25, ncol=25)
12 P2 <- matrix(0, nrow=25, ncol=25)
13 P3 <- matrix(0, nrow=25, ncol=25)
14 P4 <- matrix(0, nrow=25, ncol=25)
15
16 ##  State         1    2    3    4    5    6    7    8   9   10   11   12   13   14   15   16
          17   18   19   20   21   22   23   24
17 transitionOut <- c(2,   3,  25,   5,   6,  25,   8,   9,  25,  11,  12,  25,  14,  15,  25,
       17,  18,  25,  20,  21,  25,  23,  24,  25)
18 transitionBB  <- c(4,   5,   6,  13,  14,  15,  13,  14,  15,  16,  17,  18,  22,  23,  24,
       22,  23,  24,  22,  23,  24,  22,  23,  24)
19 transition1B  <- c(4,   5,   6,  16,  17,  18,   4,   5,   6,   4,   5,   6,  16,  17,  18,
       16,  17,  18,   4,   5,   6,  16,  17,  18)
```

```
20 transition2B  <- c(7,  8,  9, 19, 20, 21,  7,  8,  9,  7,  8,  9, 19, 20, 21,
        19, 20, 21,  7,  8,  9, 19, 20, 21)
21 transition3B  <- c(10, 11, 12, 10, 11, 12, 10, 11, 12, 10, 11, 12, 10, 11, 12,
        10, 11, 12, 10, 11, 12, 10, 11, 12)
22 transitionHR  <- c(1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2,
        3, 1, 2, 3)
23
24 for (i in 1:24) {
25     P0[i, transitionOut[i] ] <- rho0
26     PB[i, transitionBB[i]  ] <- rhoB
27     P1[i, transition1B[i]  ] <- rho1
28     P2[i, transition2B[i]  ] <- rho2
29     P3[i, transition3B[i]  ] <- rho3
30     P4[i, transitionHR[i]  ] <- rho4
31 }
32
33 P <- P0 + PB + P1 + P2 + P3 + P4
34 P[25, 25] = 1
35
36 M <- matrix(0, nrow=25, ncol=5)
37 M[22:24,   1] <- 1
38 M[7:18,  2:3] <- 1
39 M[19:24, 2:3] <- 2
40 M[4:12,    4] <- 1
41 M[13:21,   4] <- 2
42 M[22:24,   4] <- 3
43 M[1:3,     5] <- 1
44 M[4:12,    5] <- 2
45 M[13:21,   5] <- 3
46 M[22:24,   5] <- 4
47
48 playerX <- c(rhoB, rho1, rho2, rho3, rho4)
49 runsX <- M %*% playerX
50
51 Q <-  P[1:24, 1:24]
52 cat( solve(diag(24) - Q, runs[X1:24]) )
53 cat(''\n\n'')
54
55 library("markovchain")
56 nStates <- 25
57 baseballStates <- as.character(c(1:nStates))
58 mcBaseball <- new("markovchain", states = baseballStates, byrow=TRUE,
        transitionMatrix = P, name="mcX")
59 cat( expectedRewards(mcBaseball, 30, runX) )
```

.

8.12. Problems to Work for Understanding

1: Steven Strasburg of the Washington Nationals was the 2019 World Series MVP. Discuss why this model may not useful for assessing Strasburg as a baseball player.

2:   With this model, what is the expected number of at-bats for (9 copies of) Mike Trout until the third out in a half-inning?

3:   Consider a player K who each time at bat either hits a home run with probability q or strikes out with probability $1-q$. Write out the nonzero probability transitions in the matrix M_K.

4: Find the player statistics for the MVP in the National League and American League for the previous complete baseball season and compute the Markov runs for each.

5: Use this model to calculate the Markov runs for Ted Williams and Babe Ruth. Does this settle the debate about who is the greatest hitter in baseball history?

Modeling Sports With Markov Chains

Tennis and basketball are modeled with Markov chains with 20 and 8 states respectively. For tennis, winning or losing are absorbing states of the Markov chain. For basketball, the sample path of the chain provides a score for the game. These two familiar sports illustrate previously derived methods for Markov chains.

Student: contains scenes of mild algebra or calculus that may require guidance.

9.1. Chapter Starter Question

What is your favorite sport? Does the sport have discrete "states" that can be modeled with a Markov chain? What sports analytic measures exist to measure the game?

9.2. Tennis

Tennis is a popular sport played as a sequence of "games" making up a "set" and several sets making a "match", determining the overall winner. One player in a game is the server, and the opposing player is the receiver. A coin toss determines the server in the first game. Service alternates game by game between the two players or teams.

A tennis game consists of a sequence of points played with the same player serving. The first player to gain 4 points in total and at least two points more than the opponent wins. The scoring of tennis is unusual for historical reasons that are not clear. Roughly speaking, a player's point score advances by 15 when a player scores because the opponent missed returning the ball or hits it out of bounds, with a few more technical rules. The maximum score to win a game is 60 so game scores should be 0, 15, 30, 45, and 60. But tradition abbreviates the third score to 40, and calls the 0 score "love" for reasons that are again historically unclear.
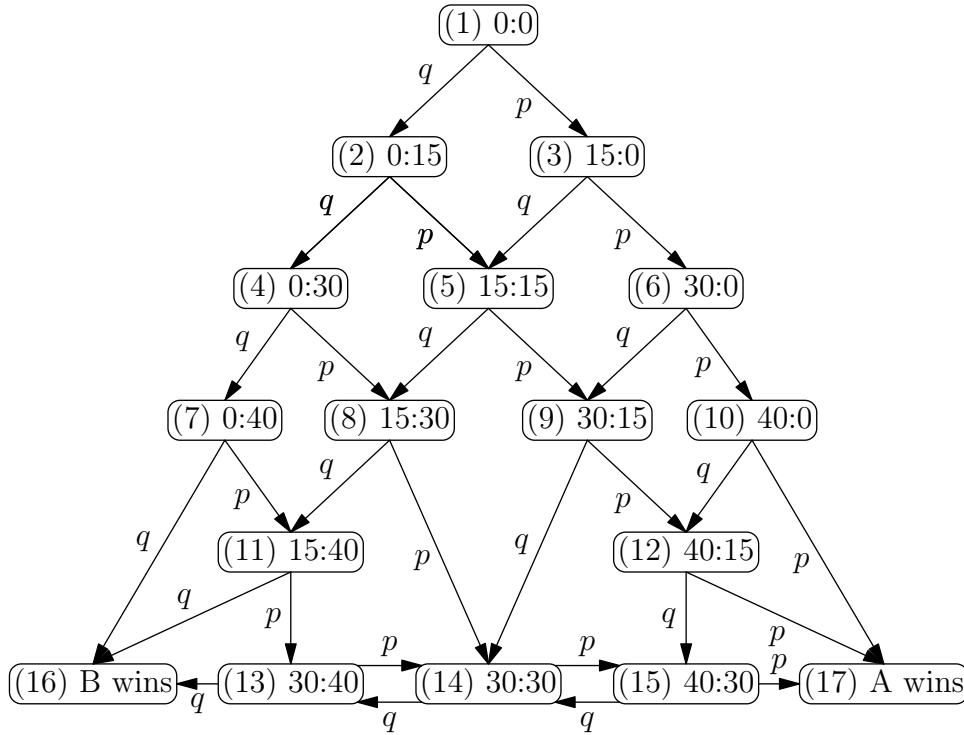
Figure 1. The state transition diagram for a game of tennis. The score for player A is first, separated by a colon from the score for player B. The numbers in parentheses are the row and column numbers of the state in the probability transition matrix. The probability labels are the same as the analogous random walk probabilities.

Thus, the running score of each game is special to tennis: scores from zero to three points are "love", "15", "30", and "40", respectively. If each player has at least three points, making the player's scores equal at 40, the score is not called out as "40–40", but rather as "deuce". If at least three points have been scored by each side and a player has one more point than the opponent, called "ad in" when the serving player is ahead, and "ad out" when the receiving player is ahead.

To model tennis as a Markov chain, take the states of the chain as the possible scores for the two players. For simplicity, assume that transition from one state or score to another is constant and depends only on the present score and not on the preceding states or scores. Figure 1 illustrates the transitions among the states. Clearly this assumption is idealizing the game, leaving out factors such as the order of service, psychological factors, fatigue, or adaptation to the opponent's style. Other factors not mentioned can also affect the transition probabilities.

As is natural for a game or a sport, this Markov chain has two absorbing states, "A wins" and "B wins". Because the victor of the game must win by at least 2 points, after 4 or 5 transitions the final set of states "B wins", "30:40", "30:30",

"40:30", "A wins" creates an absorbing random walk on 5 states. The exercises compute the probability of winning a game for specific values of p and q.

After the first game is over, a second game starts with service passing to the other side, until a *set* is completed. This happens when one side wins at least 6 games with a margin of at least 2 games. A set is completed as soon as the score is one of 6:0, 6:1, 6:2, 6:3, 6:4, 7;5, 8:6 and so on. One set follows another until one side wins the match by winning by at least 2 sets. Therefore, a new Markov chain represents the progress of the set. The absorbing probabilities of this higher level chain represent the probability of winning an individual game. In this new Markov chain for the set, after 11 or 12 games again there is an absorbing random walk on 5 states due to the requirement of winning by at least 2 games.

Finally, tennis games have a 3 or 5 set *match* to declare the overall winner by at least 2 sets, making a third-level Markov chain. Women and junior players play 3 set matches and tennis has additional rules for tiebreakers.

9.3. Basketball

This section assumes familiarity with the rules and terminology of basketball. The Markov chain model assumes the states and transition probabilities of a basketball game between teams A and B are the following.

AOff: Team A goes on offense.

APen: Penetrating offense by team A.

AScore: Basket scored by team A.

Bdef: Ball in possession of team B in defense.

BOff: Team B goes on offense.

BPen: Penetrating offense by team B.

BScore: Basket scored by team B.

ADef: Ball in possession of team A in defense.

$p_{\mathbf{eff,A}}$: Probability of organizing a penetrating offense by team A.

$p_{\mathbf{eff,B}}$: Probability of organizing a penetrating offense by team B.

$p_{\mathbf{d,A}}$: Probability of successful defense by team A.

$p_{\mathbf{d,B}}$: Probability of successful defense by team B.

$p_{\mathbf{r,A}}$: Probability of offensive rebound by team A.

$p_{\mathbf{r,B}}$: Probability of offensive rebound by team B.

$p_{\mathbf{s,A}}$: Probability of scoring a basket by team A as a result of a penetrating offense.

$p_{\mathbf{s,B}}$: Probability of a scoring a basket by team B as a result of a penetrating offense.

A diagram of the states and transition probabilities is in Figure 2. This model does not consider three-point shots and scoring by foul shots. This simple Markov chain model could be the basis for extending the model with more states and using advanced sports analytics for the transition probabilities.
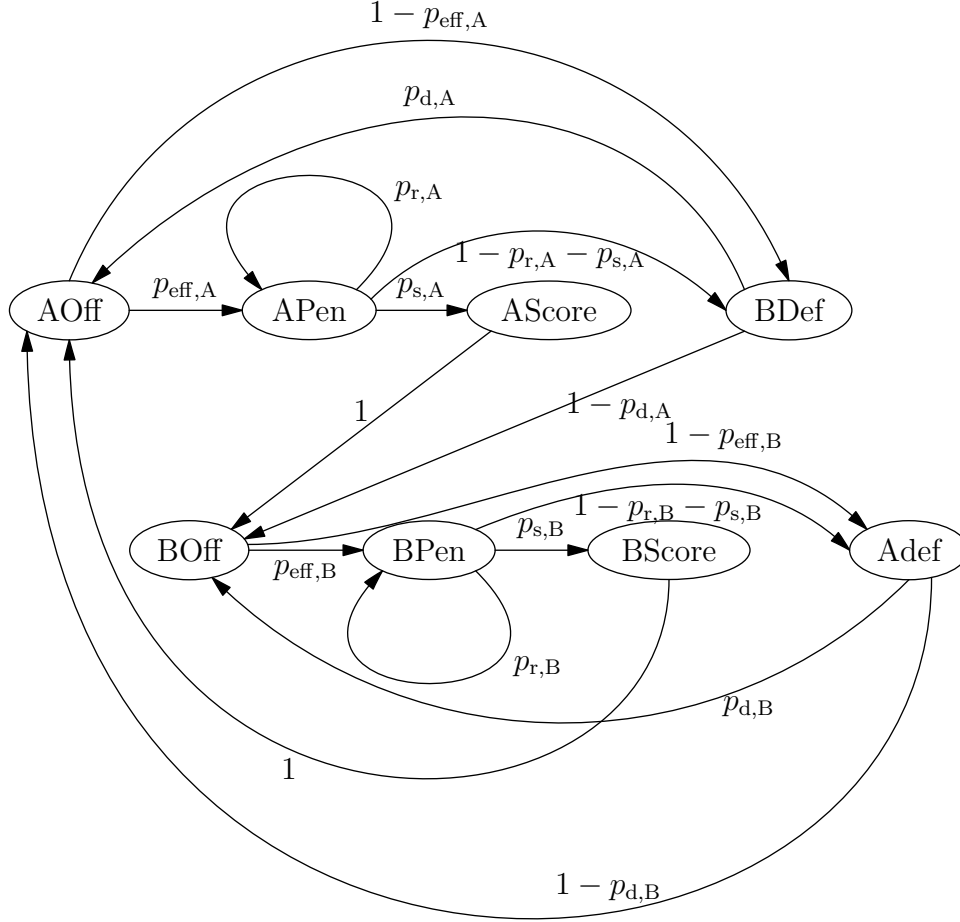
Figure 2. The state transition diagram for a game of basketball.

College basketball games have 40 minutes playing time for both the men's and women's leagues. The games last longer because of time-outs and foul shots stopping the clock, which this model does not take into account. In a typical game, each team will have about 70 to 80 possessions, measured here as being in the state "AOff" and "Boff", because starting in this state leads ultimately to either a score "AScore" or a turnover "BDef".

As an example, take $p_{\text{eff,A}} = 0.6$ and $p_{\text{eff,B}} = 0.75$, so team A is not as good offensively as team B. It is reasonable to set $p_{\text{d,A}} = 1 - p_{\text{eff,B}}$ and $p_{\text{d,B}} = 1 - p_{\text{eff,A}}$. Set $p_{\text{r,A}} = p_{\text{r,B}} = 0.3$. Set $p_{\text{s,A}} = 0.7$ and $p_{\text{s,B}} = 0.6$ so that while team A is not as effective on offense, once it gets near the basket team A scores better than team B near the basket. Then in sample paths with 450 steps in the Markov chain, A has about 75 possessions and an average final score around 89 and B also has about 75 possessions with an average final score around 99. With these parameters, having a more effective offense is more important for winning than better scoring percentage near the basket. Averaging over multiple samples can give a statistical look at

the influence of various parameters for winning this simple Markov chain model of basketball. This is an example where sample path properties are more important than stationary probabilities. See the exercises for more details.

9.4. Chapter Ending Answer

For tennis, the game points create a reasonable Markov chain model of a game. The probability a player can make a point against as opponent is the analytic measure. The sets and matches can be similarly modeled as Markov chains. For basketball, the teams are in successive states of offense, near the basket, scoring, and turnovers. Sports analytic measures can give the transition probabilities between states.

9.5. Chapter Summary

Key Concepts.

(1) Tennis can be modeled as a Markov chain with two absorbing states.

(2) Basketball can be modeled as a Markov chain. The sample path of the chain predicts the final score of the game.

Vocabulary.

(1)

(2)

Notation.

(1)

9.6. Sources

This section is adapted from [**SS93**], Sections 3.2, 3.5, 3.9, 3.10, 3.13.

9.7. Reading Suggestion:

9.8. Outside Readings and Links:

(1)

(2)

(3)

(4)

9.9. Algorithms and Scripts

Algorithm.

Data: Transition probability matrix P, starting state
Result: Either absorption probabilities and waiting time to absorption or sample path of the chain

1 *Initialization and sample paths*
2 Load Markov chain library
3 Set state names, set transition probability matrix, set start state
4 Determine absorption probabilities and waiting times OR
5 Set an example length and create a sample path of example length
6 Find sample paths and count scores.

Scripts.

R: R script for tennis.

```r
library("markovchain")

rows <- 17

p <- 0.51
q <- 1-p

P <- matrix(0, rows, rows)
P[1,  2] <- q; P[1,3] <- p;
P[2,  4] <- q; P[2,5] <- p;
P[3,  5] <- q; P[3,6] <- p;
P[4,  7] <- q; P[4,8] <- p;
P[5,  8] <- q; P[5,9] <- p;
P[6,  9] <- q; P[6,10] <- p;
P[7, 16] <- q; P[7,11] <- p;
P[8, 11] <- q; P[8,14] <- p;
P[9, 14] <- q; P[9,12] <- p;
P[10,12] <- q; P[10,17] <- p;
P[11,16] <- q; P[11,13] <- p;
P[12,15] <- q; P[12,17] <- p;
P[13,16] <- q; P[13,14] <- p;
P[14,13] <- q; P[14,15] <- p;
P[15,14] <- q; P[15,17] <- p
P[16,16] <- 1;
P[17,17] <- 1;

tennis  <- new("markovchain", transitionMatrix=P, name="Tennis")
print( summary(tennis) )

cat("Probability A wins game: ",
    absorptionProbabilities(tennis)[1,2], "\n")

T = P[1:15,1:15]
eye = diag(15)
N = solve(eye-T, eye)
ones = matrix(1, 15,1)
VarW = (2*N-eye) %*% N %*% ones - (N %*% ones)^2

cat("Expected length of game in serves: ",
      meanAbsorptionTime(tennis)[1],
    "Standard deviation: ", sqrt(VarW[1]), "\n")
```

.

R script for basketball.

```r
library("markovchain")

```

```
 3  rows <- 8
 4
 5  peffA <- 0.6
 6  peffB <- 0.75
 7
 8  pdA <-  1-peffB
 9  pdB <- 1 - peffA
10
11  prA <- 0.3
12  prB <- 0.3
13
14  psA <- 0.7
15  psB <- 0.6
16
17  stateNames <- c("AOff", "APen", "AScore", "Bdef",
18                  "BOff", "BPen", "BScore", "ADef")
19  P <- matrix(c(0,   peffA, 0,   1-peffA,   0,    0, 0, 0,
20                0,    prA,  psA, 1-prA-psA, 0,    0, 0, 0,
21                0,    0,    0,   0,         1,    0, 0, 0,
22                pdA,  0,    0,   0,         1-pdA, 0, 0, 0,
23                0,    0,    0,   0,         0,    peffB, 0,  1-peffB,
24                0,    0,    0,   0,         0,    prB,  psB, 1-prB-psB,
25                1,    0,    0,   0,         0,    0, 0, 0,
26                1-pdB,0,    0,   0,         pdB,  0, 0, 0),
27                nrow=rows, byrow=TRUE)
28  rownames(P) <- stateNames
29  colnames(P) <- stateNames
30
31  startState  <- "AOff"
32
33  bball   <- new("markovchain", transitionMatrix=P,
34                 states=stateNames, name="Basketball")
35  print( summary(bball) )
36
37  pathLength  <- 450
38  ngames <- 100
39  allAScores <- numeric(ngames)
40  allBScores <- numeric(ngames)
41
42  for (i in 1:ngames) {
43      bballHistory  <- rmarkovchain(n=pathLength, object=bball,
44                              t0=startState)
45      allAScores[i] <- 2 * sum(bballHistory == "AScore")
46      allBScores[i] <- 2 * sum(bballHistory == "BScore")
47  }
48
49  cat("Mean A Scores: ",      mean(allAScores),
50      "Standard Deviation: ", sd(allAScores), "\n")
51  cat("Mean B Scores: ",      mean(allBScores),
52      "Standard Deviation: ", sd(allBScores), "\n")
```

.

9.10. Problems to Work for Understanding

1: In tennis, assume A has probability $p = 0.6$ of winning a point. What is the probability of A winning a game? If $p = 0.51$, what is the probability of A winning a game? What is the expected length and the standard deviation of the length of the game in serves for each of these probabilities?

2: For the Markov chain model of basketball with the given transition probabilities, what are the expected scores of the two teams?

Markov Chain Epidemic Models

Markov chain **Susceptible-Infected-Removed (SIR) models**. have a population and a disease which infects members of the population for a fixed amount of time, after which the infected individuals are removed, in the sense that each one is no longer susceptible and cannot become infected again. The **Greenwood Model** assumes the number of Infectives in generation t is a binomial random variable with parameters S_t and infection success constant p, not depending on the number of Infectives. The **Reed-Frost model** includes an infective size dependency that the Greenwood model lacks.

Mathematically Mature: may contain mathematics beyond calculus with proofs.

10.1. Chapter Starter Question

Suppose a single person in a population of S susceptible individuals has a communicable disease. What is a simple probabilistic model of how this person can infect some number of susceptible individuals in a given time interval?

10.2. Introduction

This chapter is an introduction to elementary Markov chain **Susceptible-Infected-Removed (SIR) models**. These Markov chains model a population and a disease that infects members of the population for a fixed amount of time, after which each infected individual is removed, in the sense that they are no longer susceptible and cannot become infected again. The removed class includes individuals who are no longer susceptible either because of recovery and subsequent immunity or because of death. That is, after members of the population have been infected for one time unit, the model no longer considers them, as the removed indiduals cannot infect anyone else, nor can they become infected again. Therefore only two classes need
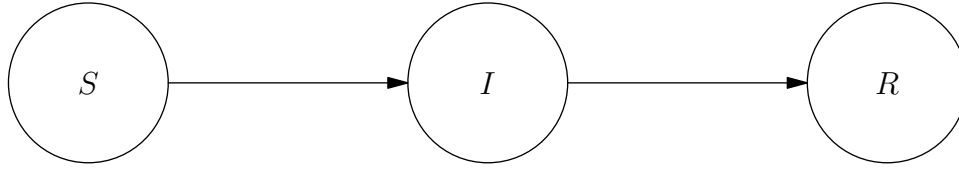
Figure 1. Schematic class diagram for the simple Susceptible-Infective-Removed epidemic model illustrating the classes of the epidemic and the movement from class to class and not the state transitions of the Markov chain.
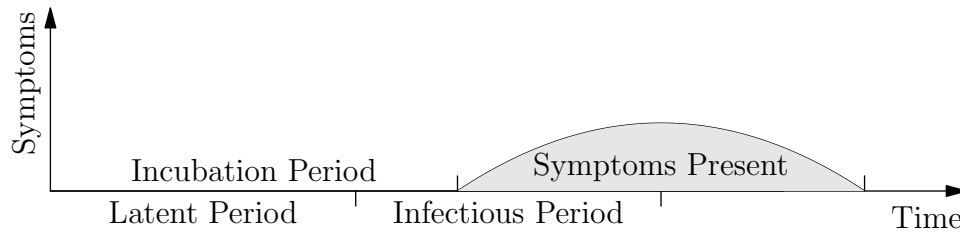
Figure 2. Possible relation of disease stages or periods.

to be considered, the Susceptibles and the Infectives. Figure 1 shows the movement between the classes.

Markov chain epidemic models use discrete time and a discrete state space. The discrete time will be in units of the infectivity period of the disease. Typically this unit time is about 7-8 days for measles, and about 7-9 days for influenza. Call each of these unit time periods a generation, enumerated by the integer variable t. (This is a slight change from the usual notation for discrete time Markov chains using the index n for the discrete time. Using t avoids confusion with the number of Infectives.) The infectivity period may either be distinct from, or overlap, the period when the infected person shows symptoms of the disease. The incubation period, the period between infection and onset of symptoms, and the latent period, the time from infection to infectiousness are different from the infectivity period. Figure 2 illustrates one possible configuration of these disease periods. The states will be the number of Susceptibles and Infectives in a homogeneous and well-mixed population. Assume the epidemic takes place over a short enough time interval that the population is a constant value N, that is, assume no births and no emigration or immigration. So it is enough to consider only Susceptibles and Infectives since the number of individuals in the Removed class is determined from the other two.

To keep track of each group, denote the number of Susceptibles at generation t by S_t, and the number of Infectives at generation t by I_t. These random variables may achieve any integer value such that $S_t + I_t$ is in the population range.

A natural first question for any epidemic model is the probability of having I_{t+1} Infectives at generation $t + 1$. In asking this question at generation t, the number of Susceptibles and Infectives at every generation up to t are known. Since the model is expressed in time units of infectivity periods where individuals move from one class to another, the main focus is on the number of Infectives at generation t,

ignoring the past, also keeping track of the Susceptibles. Then the model is a first-order Markov process through the states S_t, I_t with $S_t + I_t \leq N$. If the population is N, the number of states is $N(N+1)/2$ so this is a Markov chain with many states.

10.3. Greenwood Model

The **Greenwood Model** assumes the number of Infectives in generation t is a binomial random variable with parameters S_t and infection success constant p, not depending on the number of Infectives:

$$\mathbb{P}\left[I_{t+1} = i_{t+1} \mid S_0 = s_0, I_0 = i_0, \ldots, S_t = s_t, I_t = i_t\right] =$$

$$\mathbb{P}\left[I_{t+1} = i_{t+1} \mid S_t = s_t, I_t = i_t\right] = \binom{s_t}{i_{t+1}} p^{i_{t+1}} (1-p)^{s_t - i_t}.$$

The first equality represents the first-order Markov chain assumption. Consider the second equality in further detail. At generation t there are s_t Susceptibles. To find the probability that i_{t+1} members of this group become Infectives, choose i_t of the Susceptibles who will become infected. Similarly p is the probability that a given Susceptible will become infected in one generation. Thus the probability i_{t+1} Susceptibles become Infectives is $p^{i_{t+1}}$. Note that a Susceptible becoming Infective is a "success" (for the disease, not the population) in the binomial random variable sense. Remaining a Susceptible is a failure in the binomial sense. The transition for the Susceptibles is

$$S_{t+1} = S_t - I_{t+1}.$$

Because I_{t+1} is a binomial random variable

$$E[I_{t+1} \mid S_t] = pS_t.$$

This expression gives us the expected number of Infectives at generation $t+1$ given the number of Susceptibles at generation t. Extend this idea to say

$$\mathbb{E}\left[S_{t+1} \mid S_t\right] = \mathbb{E}\left[S_t\right] - \mathbb{E}\left[I_{t+1}\right] = (1-p)S_t.$$

By induction

$$\mathbb{E}\left[S_t \mid S_0 = s_0\right] = (1-p)^t s_0,$$
$$\mathbb{E}\left[I_t \mid S_0 = s_0\right] = p(1-p)^{t-1} s_0.$$

10.4. Reed-Frost Model

The Reed-Frost model is an SIR mathematical model of epidemics created in the 1920s by Lowell Reed and Wade Hampton Frost at Johns Hopkins University. Originally presented in a talk by Frost in 1928 and used in courses at Hopkins for two decades, the mathematical formulation was not published until the 1950s.

The Greenwood model has a fixed probability of infection, independent of the number of Infectives at the current generation step. The **Reed-Frost model** includes an infective size dependency that the Greenwood model lacks. The Reed-Frost model is based on the following assumptions:

(1) The infection is spread directly from Infectives to Susceptibles by a certain type of contact (termed "adequate contact") and in no other way.

(2) Any Susceptible in the group, after such contact with an Infective in a given generation, will develop the infection and will be infectious to others only within the following generation period. In subsequent generation periods, the infected individual is wholly and permanently immune and moves to the Removed class.

(3) Each individual has a fixed probability p of coming into adequate contact with any other specified individual in the group within one time interval, and this probability is the same for every member of the group.

(4) The population is constant with no births, immigration or emigration.

(5) These conditions stay constant during the epidemic.

Set the following initial parameters:

- the size of the population N,
- the number of individuals already immune, typically 0,
- the number of Infective cases, usually set at 1,
- the probability p of adequate contact.

If p is the probability of a given Susceptible *making adequate contact* with 1 specific Infective, then $q = 1 - p$ is the probability of *avoiding* adequate contact with that infective. Make the (vastly simplifying and probably unjustified) assumption that all interactions of the Susceptible with all Infectives are independent events. Then q^{i_t} is the probability the given Susceptible at generation t avoids adequate contact at generation t from *all* Infectives in the population at that generation and remains Susceptible at generation $t + 1$. This is a failure for the epidemic in the binomial random variable sense. The probability of a Susceptible being infected is then $(1 - q^{i_t})$, a success for the epidemic. Then as a binomial random variable, the probability of the number of Infectives in the next generation is

$$\mathbb{P}\left[I_{t+1} = i_{t+1} \mid S_t = s_t, I_t = i_t\right] = \binom{s_t}{i_{t+1}} \left((1 - q^{i_t})\right)^{i_{t+1}} \left(q^{i_t}\right)^{s_t - i_t}$$

and $S_{t+1} = S_t - I_{t+1}$. In comparison to the Greenwood model, the Reed-Frost model gives a higher probability to more new Infectives at generation $t + 1$ if the population already has a large number of Infectives at generation t than if there is only a small population of Infectives. The Reed-Frost model has a positive feedback effect on infectives.

A success for the epidemic is a failure for the Susceptibles and a failure for the epidemic is a success for the Susceptibles. Then an alternative expression for the Susceptibles in the Reed-Frost model is

$$\mathbb{P}\left[S_{t+1} = s_{t+1} \mid S_t = s_t, I_t = i_t\right] = \binom{s_t}{i_{t+1}} \left(q^{i_t}\right)^{i_{t+1}} \left(1 - q^{i_t}\right)^{s_t - i_t}$$

and $I_{t+1} = St - S_{t+1}$.

Define the **extinction time τ of an epidemic** to be the generation at which i_t is first zero. This is an absorbing state for the Markov Chain. Note that if the

number of Susceptibles becomes 0 in a generation t, then $i_{t+1} = 0$. Of more interest is the probability $S_\tau > 0$ and $I_\tau = 0$. Consider an epidemic for which the Infectives have progressed to extinction $i_1, i_2, \ldots, i_{\tau-1}, i_\tau = 0$. Note that $i_{\tau-1} > 0$. Thus

$$\mathbb{P}\left[I_1 = i_1, \ldots, I_k = i_k, I_{k+1} = 0 \mid S_0 = n, I_0 = m\right] =$$
$$\mathbb{P}\left[I_1 = i_1 \mid S_0 = n, I_0 = m\right] \times \mathbb{P}\left[I_2 = i_2 \mid S_1 = n - i_1, I_1 = i_1\right] \times \cdots \times$$
$$\mathbb{P}\left[I_\tau = 0 \mid S_{\tau-1} = s_{\tau-1}, I_{\tau-1} = i_{\tau-1}\right].$$

In this specific epidemic, the **total damage** of the epidemic is $K = \sum_{j=0}^{\tau} I_j$. As K is the sum of I_i's, treat it as a random variable. Thus consider:

$$\mathbb{P}\left[K = k \mid S_0 = n, I_0 = m\right] = \sum_{\vec{i}: |\vec{i}| = k} \mathbb{P}\left[I_1 = i_1, \ldots, I_\tau = 0 \mid S_0 = n, I_0 = m\right].$$

Some common asymptotics apply here. Let $m = i_0 = o(n)$ and suppose $q = \mathrm{e}^{-\lambda/n}$ so $p = 1 - q = O(n^{-1})$ and the expected number of adequate contacts q^{i_t} is approximately constant. At the beginning of the epidemic, using the first formulation of the Reed-Frost model

$$\mathbb{P}\left[I_{t+1}\right] = \mathrm{Bin}(S_t, 1 - \mathrm{e}^{-\lambda I_t/n})$$
$$\approx \mathrm{Bin}(n, \lambda I_t/n)$$
$$\approx \mathrm{Poisson}(\lambda I_t)$$

This means I_{t+1} is the sum of I_t independent Poisson random variables with parameter λ. That is, the number of Infectives is the result of a Galton-Watson process or branching process. Then from the standard theory for branching processes, the epidemic is *subcritical* if $\lambda \leq 1$ and *supercritical* if $\lambda > 1$. That is, the trajectories of this process either die out or explode in an exponential way, at least at the beginning of the epidemic while the asymptotic assumptions hold. This points to more general stochastic process models of epidemics as branching processes. Although the epidemic models are Markov chains, the tools of branching processes are more applicable and yield more detailed results.

10.5. Limitations

Both of these models assume a homogeneous population, i.e. everyone is equally likely to get sick assuming the same exposure. Also the Reed-Frost model assumes that the population mixes perfectly, that is, every Susceptible interacts with every Infective and thus has an equally likely chance to get sick from each one. Both models assume independence in these interactions. These assumptions apply best to a moderately-sized (reasonably) closed situations such as a dormitory, a nursing home under quarantine, a cruise ship, a factory, or perhaps a small town.

For large populations such as a city or an entire country, the assumptions of homogeneous population and perfect mixing don't apply. Also for large populations, direct calculation of the binomial coefficients at each stage is inconvenient, and so using continuum limits is appropriate, leading to differential equation epidemic models. However, one advantage of the stochastic process models is the possibility of obtaining a distribution for the extinction times. This is a more realistic property than the extinction time given by an asymptotic deterministic model.

More detailed models have multiple health and infection classes instead of just Infectives. More detailed models may also cross these with factors influencing mixing, such as age, social class, or geographical location. The mathematical modeling of epidemics has a huge literature, the references below are a starting point.

10.6. Chapter Ending Answer

One particularly simple model is that the infective transmits the disease to a fixed number of susceptibles, this is not probabilistic. A probabilistic model is that the infective transmits to each susceptible with probability p, leading to the binomial probability distribution of the Greenwood model. A third possibility is that the number of infected is a Poisson random variable with parameter λ. This would be appropriate if the population is large and the transmission rate is constant across various size groups of susceptibles.

10.7. Chapter Summary

Key Concepts.

(1) Markov chain **Susceptible-Infected-Removed (SIR) models**. have a population and a disease which infects members of the population for a fixed amount of time, after which the infected individuals are removed, in the sense that each one is no longer susceptible and cannot become infected again.

(2) The **Greenwood Model** assumes the number of Infectives in generation t is a binomial random variable with parameters S_t and infection success constant p, not depending on the number of Infectives.

(3) For the Greenwood model

$$\mathbb{E}\left[S_t \mid S_0 = s_0\right] = (1-p)^t s_0,$$
$$\mathbb{E}\left[I_t \mid S_0 = s_0\right] = p(1-p)^{t-1} s_0.$$

(4) The **Reed-Frost model** includes an infective size dependency that the Greenwood model lacks.

(5) In comparison to the Greenwood model, the Reed-Frost model gives a higher probability to more new Infectives at generation $t+1$ if there are already a large number of Infectives at generation t than if there is only a small population of Infectives. The Reed-Frost model has a positive feedback effect on infectives.

(6) Under some common asymptotic assumptions, is, the number of Infectives is the result of a Galton-Watson process or branching process. Then from the standard theory for branching processes, the epidemic is *subcritical* if $\lambda \leq 1$ and *supercritical* if $\lambda > 1$.

Vocabulary.

(1) Markov chain **Susceptible-Infected-Removed (SIR) models**. have a population and a disease which infects members of the population for a fixed amount of time, after which the infected individuals are removed, in the sense that each one is no longer susceptible and cannot become infected again.

(2) The **Greenwood Model** assumes the number of Infectives in generation t is a binomial random variable with parameters S_t and infection success constant p, not depending on the number of Infectives.

(3) The **Reed-Frost model** includes an infective size dependency that the Greenwood model lacks.

(4) The **extinction time τ of an epidemic** to be the generation at which i_t is first zero.

(5) The **total damage** of the epidemic is $K = \sum_{j=0}^{\tau} i_j$.

Notation.

(1) t – the integer variable enumerating unit time periods for each generation

(2) N – population constant value

(3) S_t – the number of Susceptibles at generation t

(4) I_t – the number of Infectives at generation t

(5) p – the probability that a given Susceptible will become infected in one generation

(6) $K = \sum_{j=0}^{\tau} I_j$ – the total damage of the epidemic

(7) $q = 1 - p$ – the probability of *avoiding* contact with that infective

(8) τ – extinction time of an epidemic to be the generation at which i_t is first zero

(9) s_t – specific numbers of Susceptibles.

(10) i_t – the specific number of Susceptibles who will become infected

(11) $m = i_0 = o(n)$

(12) λ – parameter such that $q = \mathrm{e}^{-\lambda/n}$

10.8. Sources

Details of the Greenwood and Reed-Frost models are adapted from "Criticality in Epidemic Models" by R. Dolgoarshinnykh, [**Dol**]. Comments on epidemics as branching processes and more detailed assumptions for epidemics are from "Branching Processes: Their Role in Epidemiology", by C. Jacob, [**Jac10**]. See also [**SR13**] for a survey of mathematical models of infectious diseases. The exercise is adapted from the Ohio Supercomputer Center Summer Institute, Reed-Frost Epidemic Model.

10.9. Reading Suggestion:

10.10. Outside Readings and Links:

(1) *Introduction to Probability*, Grinstead and Snell. Chapter 11 - Markov Chains - http://tinyurl.com/qw6sa

(2) Generalized Markov Models of Infectious Disease Spread

(3) Criticality in Epidemic Models

1:

(a) There are 4 states, D for dropping out, 1 for first-year, 2 for second-year, and G for graduating.

$$
P = \begin{array}{c} \\ D \\ 1 \\ 2 \\ G \end{array}
\begin{array}{c} \begin{array}{cccc} D & 1 & 2 & G \end{array} \\
\left(\begin{array}{cccc}
1 & 0 & 0 & 0 \\
3/20 & 1/4 & 3/5 & 0 \\
1/10 & 0 & 1/5 & 7/10 \\
0 & 0 & 0 & 1
\end{array} \right).
\end{array}
$$

(b) Absorbing states are D and G. Transient states are 1, 2.

(c)

$$
P^2 = \left(\begin{array}{cccc}
1 & 0 & 0 & 0 \\
99/400 & 1/16 & 27/100 & 21/50 \\
31/250 & 0 & 1/125 & 217/250 \\
0 & 0 & 0 & 1
\end{array} \right).
$$

Probability of graduating on time is $21/50$.

$$
P^3 = \left(\begin{array}{cccc}
1 & 0 & 0 & 0 \\
2271/8000 & 1/64 & 183/2000 & 609/1000 \\
31/250 & 0 & 1/125 & 217/250 \\
0 & 0 & 0 & 1
\end{array} \right).
$$

(d) Probability of graduating in 3 years is $609/1000$.

(e) Canonical form is

$$
\left(\begin{array}{cccc}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
3/20 & 0 & 1/4 & 3/5 \\
1/10 & 7/10 & 0 & 1/5
\end{array} \right)
$$

so

$$
N = \begin{pmatrix} 4/3 & 1 \\ 0 & 5/4 \end{pmatrix} \qquad B = NA = \begin{pmatrix} 3/10 & 7/10 \\ 1/8 & 7/8 \end{pmatrix}.
$$

The chance that a student will graduate eventually is $7/10$. The chance the student will drop out eventually is $3/10$.

(f) The expected time to graduation is $4/3 + 1 = 7/3$ years.

Remark 10.1. Problem created by LT Grant, 2006.

2:

(a)

$$
P = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array}
\begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\
\left(\begin{array}{cccccc}
0 & 1/3 & 1/3 & 0 & 1/3 & 0 \\
1/2 & 0 & 0 & 1/2 & 0 & 0 \\
1/2 & 0 & 0 & 1/2 & 0 & 0 \\
0 & 1/3 & 1/3 & 0 & 0 & 1/3 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{array} \right).
\end{array}
$$

(b) Absorbing states are E and F. Transient states are A, B, C, D.

(c)

$$
P' = \begin{array}{c} \\ E \\ F \\ A \\ B \\ C \\ D \end{array}
\begin{array}{c} \begin{array}{cccccc} E & F & A & B & C & D \end{array} \\
\left(\begin{array}{cccccc}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
1/3 & 0 & 0 & 1/3 & 1/3 & 0 \\
0 & 0 & 1/2 & 0 & 0 & 1/2 \\
0 & 0 & 1/2 & 0 & 0 & 1/2 \\
0 & 1/3 & 0 & 1/3 & 1/3 & 0
\end{array} \right) \end{array}.
$$

$$
N = \begin{pmatrix}
2 & 1 & 1 & 1 \\
\frac{3}{2} & 2 & 1 & \frac{3}{2} \\
\frac{3}{2} & 1 & 2 & \frac{3}{2} \\
1 & 1 & 1 & 2
\end{pmatrix}.
$$

(d) $(N\mathbf{1})_1 = 2 + 1 + 1 + 1 = 5$.

(e) From A, probability to E is $2/3$, to F is $1/3$. From D, probability to E is $1/3$, to F is $2/3$. Seems reasonable by symmetry.

Remark 10.2. Problem created by LT Grant, 2006.

3:

(a) Expect absorption probabilities from B to E to increase and absorption probability of B to F to decrease.

(b) From A, probability to E to increase, to F to decrease. From D, probability to E to increase, to F to decrease.

(c)

$$
\begin{pmatrix}
0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\
\frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\
\frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 \\
0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}.
$$

(d) From A, probability to E increases to 0.792, to F decreases to 0.208. From D, probability to E increases to approximately 0.458, to F decreases to approximately 0.542.

Remark 10.3. Problem created by LT Grant, 2006.

4:

$$
P_{ij} = \begin{cases}
(\frac{i}{N})^2 & j = i - 1, & i = 1, 2, \dots, N \\
(2\frac{i}{N})\frac{N-i}{N} & j = i, & i = 0, 1, 2, \dots, N \\
1 - (\frac{i}{N})^2 & j = i - 1, & i = 0, 1, 2, \dots, N - 1 \\
0 & \text{otherwise}
\end{cases}
$$

There are no absorbing states.

As an example, for $N = 3$, the transition probability matrix is

$$
\begin{array}{c}
 \\
0 \\
1 \\
2 \\
3
\end{array}
\begin{array}{cccc}
0 & 1 & 2 & 3 \\
\end{array}
$$

$$
\begin{array}{c}
0 \\
1 \\
2 \\
3
\end{array}
\left(
\begin{array}{cccc}
0 & 1 & 0 & 0 \\
1/9 & 4/9 & 4/9 & 0 \\
0 & 4/9 & 4/9 & 1/9 \\
0 & 0 & 1 & 0
\end{array}
\right)
$$

with stationary distribution $\pi = (1/20, 9/20, 9/20, 1/20)$ For $N = 4$, the transition probability matrix is

$$
\begin{array}{c}
0 \\
1 \\
2 \\
3 \\
4
\end{array}
\left(
\begin{array}{ccccc}
0 & 1 & 2 & 3 & 4 \\
0 & 1 & 0 & 0 & 0 \\
1/16 & 6/16 & 9/16 & 0 & 0 \\
0 & 4/16 & 8/16 & 4/16 & 0 \\
0 & 0 & 9/26 & 6/16 & 1/16 \\
0 & 0 & 0 & 1 & 0
\end{array}
\right)
$$

Remark 10.4. Adapted from Karlin and Taylor, *A First Course in Stochastic Processes*, second edition, page 73, problem 1(b).

5:

$$
P_{ij}
\begin{cases}
\left(\frac{i}{N}\right) q & j = i - 1, \quad i = 1, 2, \ldots, N \\
\left(\frac{i}{N}\right) p + \left(\frac{N-i}{N}\right) q & j = i, \quad i = 0, 1, 2, \ldots, N \\
\left(1 - \frac{i}{N}\right) p & j = i + 1, \quad i = 0, 1, 2, \ldots, N-1 \\
0 & \text{otherwise}
\end{cases}
$$

An example in the case $N = 4$

$$
\begin{pmatrix}
q & p & 0 & 0 & 0 \\
\frac{q}{4} & \frac{3q}{4} + \frac{p}{4} & \frac{3p}{4} & 0 & 0 \\
0 & \frac{q}{2} & \frac{q}{2} + \frac{p}{2} & \frac{p}{2} & 0 \\
0 & 0 & \frac{3q}{4} & \frac{q}{4} + \frac{3p}{4} & \frac{p}{4} \\
0 & 0 & 0 & q & p
\end{pmatrix}.
$$

The unnormalized stationary distribution is $[1, 4p/q, 6p^2/q^2, 4p^3/q^3, p^4/q^4]$, for $p = 1/2 = q$, normalized, this is $[1/16, 1/4, 3/8, 1/4, 1/16]$, for $p = 3/4$, $q = 1/4$, normalized, this is approximately $[0.004, 0.047, 0.211, 0.422, 0.316]$.

There are no absorbing or transient states.

Remark 10.5. Adapted from Karlin and Taylor, *A First Course in Stochastic Processes*, second edition, page 74, problem 2(a).

6:

$$
P_{ij} =
\begin{cases}
\left(1 - \frac{i}{N}\right) p & j = i - 1, \quad i = 1, 2, \ldots, N \\
\left(\frac{i}{N}\right) p + \left(\frac{N-i}{N}\right) q & j = i, \quad i = 0, 1, 2, \ldots, N \\
\left(\frac{i}{N}\right) q & j = i + 1, \quad i = 0, 1, 2, \ldots, N-1. \\
0 & \text{otherwise}
\end{cases}
$$

There are no absorbing or transient states.

Remark 10.6. Adapted from Karlin and Taylor, *A First Course in Stochastic Processes*, second edition, page 74, problem 2(b).

7: From numerical experiments with the transition probability matrix, slightly more than 50% of the time. This matrix converges fairly slowly to the limiting matrix, so experiments require about P^{50}.

Remark 10.7. Adapted from FiveThirtyEight Riddler who in turn got the problem from Steve Simon.

8: Keep track of the different "states" of Louie's umbrellas at home at the end of the work day. He could have 3 at home and 0 at work, 2 at home and 1 at work and so on. In this solution, there are five states: 0, 1, 2 or 3 umbrellas at home plus the state in which Louie has already gotten wet. The rows of the matrix are Louie's current state, and the columns are his destination. For example, he starts in the third row (2 umbrellas at home and 1 at work). There is no chance he will end the day with zero umbrellas at home since he always has an umbrella available, there is a $0.3 = 0.5 \cdot 0.6$ chance he will end the day with just 1 at home (from rain in the morning and no rain in the afternoon), a 0.5 chance with 2 at home, a 0.2 chance with 3 at home, and no chance he will get wet on this first day since he always has an umbrella available.

The movement through the random states of a day is described by the transition matrix, which contains the probabilities of umbrellas moving from 1 place to another on any given day. The transition probability matrix is:

$$\begin{pmatrix} 0.3 & 0.2 & 0.0 & 0.0 & 0.5 \\ 0.3 & 0.5 & 0.2 & 0.0 & 0.0 \\ 0.0 & 0.3 & 0.5 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.3 & 0.5 & 0.2 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{pmatrix}.$$

To get the probabilities for the entire work week, simply multiply this matrix by itself five times, representing the umbrella transitions due to meteorological randomness over the five days, which gives

$$\begin{pmatrix} 0.04791 & 0.07634 & 0.04976 & 0.01776 & 0.80823 \\ 0.11451 & 0.19889 & 0.15274 & 0.06752 & 0.46634 \\ 0.11196 & 0.22911 & 0.22553 & 0.12610 & 0.3073 \\ 0.05994 & 0.15192 & 0.18915 & 0.12425 & 0.47474 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Last, look at the third row (the state in which he began) and the final column: That's the chance he gets wet during the week. Subtract from 1 to find the chance that he didn't get wet and get 69.27 percent.

Remark 10.8. Adapted from FiveThirtEight Riddler who in turn got the problem from Josh Vandenham.)

Remark 10.9. This problem and the solution as a Markov chain appeared in FiveThirtyEight Riddler column of December 14, 2018.

This is a version of Ross, page 181, problem number 22.

9:

(a) For this question, the rounds don't matter at all. The game will end once there is at least 1 correct ball in each cup, and the pruning phase at the end of each round only removes incorrect balls so effectively neglects pruning.

   This is a classical coupon collector problem. Define Y_i, for i from 0 to $N-1$, to be the number of additional tosses that need to be obtained after i distinct cups have been filled with corresponding numbered balls. Define Y as the total number of tosses required to fill the cups with correspondingly numbered balls so

$$Y = Y_0 + Y_1 + Y_2 + \cdots + Y_{N-1}.$$

When i distinct cups have been filled with corresponding numbered balls, the probability that a ball with a number of an unfilled cup is drawn is $\frac{N-i}{N}$. Each time a ball is thrown, there is a $1/N$ chance it will land in the correct cup. It follows that the probability of success in filling a new cup with the correctly numbered ball is $\frac{N-i}{N} \cdot \frac{1}{N}$. Thus, Y_i is a geometric random variable with success parameter $\frac{N-i}{N^2}$. Then $\mathbb{E}[Y_i] = \frac{N^2}{N-i}$. Therefore

$$\mathbb{E}[Y] = \mathbb{E}[Y_0] + \mathbb{E}[Y_1] + \mathbb{E}[Y_2] + \cdots + Y_{N-1}$$
$$= N + \frac{N^2}{N-1} + \frac{N^2}{N-2} + \cdots + \frac{N^2}{1}$$
$$= N^2 \sum_{\nu=1}^{N} \frac{1}{\nu} \approx N^2 (\log N + \gamma)$$

where $\gamma \approx 0.577$ is the Euler-Mascheroni constant.

(b) In general the number of rounds to end the game is complicated, so to illustrate the approach solve it for the case $N = 3$. Begin by breaking each round into a number of stages. Each stage tracks the progress of cups as they are filled. In the diagram below, each cup is either empty (no ball), is red (incorrect ball), or is green (correct ball). The transition probabilities are on the arrows.
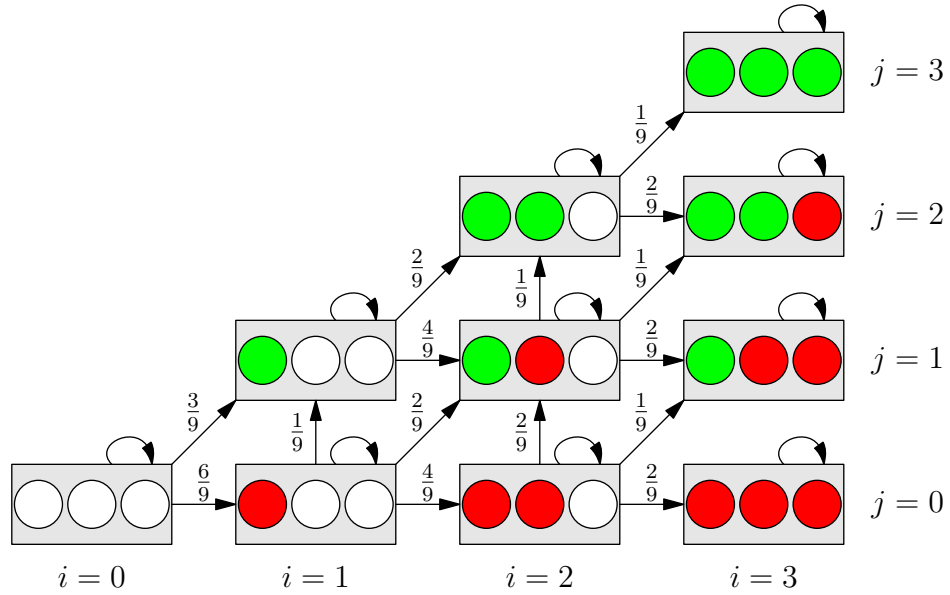
The integer i indicates the number of balls thrown. The integer j indicates the number of cups filled holding a correspondingly numbered ball. A colored cup may contain more than 1 ball. The cup position left to right is not associated with the cup or ball numbers, the position merely indicates the number of correctly or incorrectly filled cups. For example, if 2 balls have been thrown into 2 cups, 1 correctly numbered, 1 incorrectly numbered, the center diagram at $i = 2, j = 1$ illustrates this case. Then on the next ball thrown, that ball can incorrectly go into the empty cup with probability $\frac{N-1}{N} \cdot \frac{1}{N} = \frac{2}{9}$. The ball can go into the correct cup with probability $\frac{1}{N}$. In this diagram, self-loops have whatever probability is required so that the outgoing arrows sum to 1. This is a Markov chain.

Start on the initial state at the lower left (all cups empty, $i = 0$, $j = 0$), and advance to an absorbing state at the right, all cups full with $i = 3$. Label the states by columns from top to bottom from right to left. This unusual ordering puts the transition matrix in canonical form. The transition matrix for $N = 3$ is:

$$
P = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{9} & \frac{2}{9} & 0 & 0 & \frac{6}{9} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{9} & \frac{2}{9} & 0 & \frac{1}{9} & \frac{5}{9} & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{9} & \frac{2}{9} & 0 & \frac{2}{9} & \frac{4}{9} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{2}{9} & \frac{4}{9} & 0 & \frac{3}{9} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{4}{9} & \frac{4}{9} & \frac{1}{9} & \frac{2}{9} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{9} & \frac{6}{9} & 0
\end{pmatrix} .
$$

The matrix $N = (I - T)^{-1}$ is:

$$N = \begin{pmatrix} \frac{3}{1} & 0 & 0 & 0 & 0 & 0 \\ \frac{3}{2} & \frac{9}{4} & 0 & 0 & 0 & 0 \\ \frac{3}{5} & \frac{9}{10} & \frac{9}{5} & 0 & 0 & 0 \\ \frac{2}{1} & \frac{3}{2} & \frac{3}{2} & \frac{3}{2} & 0 & 0 \\ \frac{37}{35} & \frac{48}{35} & \frac{36}{35} & \frac{3}{14} & \frac{9}{7} & 0 \\ \frac{48}{35} & \frac{99}{70} & \frac{24}{35} & \frac{9}{14} & \frac{6}{7} & 1 \end{pmatrix}$$

and

$$B = NA = \begin{pmatrix} \frac{1}{3} & \frac{2}{3} & 0 & 0 \\ \frac{1}{12} & \frac{5}{12} & \frac{1}{2} & 0 \\ \frac{1}{30} & \frac{1}{6} & \frac{2}{5} & \frac{2}{5} \\ \frac{1}{6} & \frac{2}{7} & \frac{1}{3} & 0 \\ \frac{1}{15} & \frac{2}{7} & \frac{44}{105} & \frac{8}{35} \\ \frac{1}{10} & \frac{5}{14} & \frac{41}{105} & \frac{16}{105} \end{pmatrix}.$$

The ij entry in B gives the probability of going from state $i + 4$ to absorbing state $j = 1, 2, 3, 4$.

Being in absorbing state j means the throwing phase of the round ends and the pruning phase of the round begins. Absorbing state 1 requires no pruning, the round is finished. Absorbing state 2 transitions to state 5, state 3 transitions to state 8 and state 4 transitions to state 10 and the throwing part of the next round resumes.

Now let E_5, E_8, E_{10} be the expected number of rounds played to end the game starting from states $5, 8, 10$ respectively. The main interest is E_{10} which answers the question. Using first-step analysis, these expected values satisfy the equations

$$E_5 = 1 + \frac{2}{3} E_5$$

$$E_8 + 1 + \frac{1}{2} E_5 + \frac{1}{3} E_8$$

$$E_{10} = 1 + \frac{5}{14} E_5 + \frac{41}{105} E_8 + \frac{16}{105} E_{10}.$$

Solving numerically, $E_5 = 3$, $E_8 = 3.75$, $E_{10} = 4.171$.

Remark 10.10. This problem and the solution as a Markov chain appeared in FiveThirtyEight Riddler column of November 16, 2018. The solution and solution figures are adapted from Five Thirty Eight Riddler.

1:

(a)

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

(b)

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \text{ or } \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}$$

(c)

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

(d)

$$\begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 0 & 1 \end{pmatrix}.$$

State 2 communicates with states 1 and 3, but that is all.

(e)

$$\begin{pmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1/2 & 1/2 \end{pmatrix}.$$

2: The 4 states are the number of balls in urn A, from 0 to 3.

$$P = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix}.$$

In this Markov chain all states communicate, are positive recurrent, are aperiodic, and there are no absorbing or transient states, the Markov chain is irreducible. The stationary distribution is $\pi = [1/4, 1/4, 1/4, 1/4]$ so in the long run urn A spends $1/4$ of the time empty. This does not depend on the initial distribution of balls.

3: Label the $2(n+1)$ states as $(H, k, n-k)$ or $(W, k, n-k)$ indicating (redundantly) as starting the trip from Home with k umbrellas at home, and $n-k$ umbrellas at Work. Enumerate the states of the Markov chain successively as

1:: $(H, 0, n)$ is the state of starting from home with 0 umbrellas at home and n at work.

2:: $(W, 0, n)$ is the state of starting from work with 0 umbrellas at home and n at work.

: ...

$2n-1$:: $(H, n-1, 1)$ the state of starting from home with $n-1$ umbrellas and 1 at work.

$2n$:: $(W, n-1, 1)$ is the state of starting from work with $n-1$ umbrellas at home and 1 at work.

$2n+1$:: $(H, n, 0)$ is the state of starting from home with n umbrellas at home and 0 at work.

$2n+2$:: $(W, n, 0)$ is the state of starting from work with n umbrellas at home and 0 at work.

Then the probability transition matrix is

$$\begin{pmatrix} 0 & 1 & 0 & \ldots & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & \ldots & 0 & 0 & 0 \\ \ldots & & & & & & \\ 0 & 0 & 0 & \ldots & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & \ldots & 0 & 1 & 0 \end{pmatrix}$$

It is easiest to examine a small case, say $n = 3$ and extrapolate from that concrete example. For $n = 3$, the probability transition matrix is

$$\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\
0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\
0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 \\
0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 \\
0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}$$

Then solving $\pi = \pi \mathbf{P}$ for this small case $(n = 3)$ suggests that the solution is $\pi = (1/(2(2n + 1)), 1/(2n + 1), \ldots, 1/(2n + 1), 1/(2(2n + 1)))$. Then it is easy to verify that indeed this is a stationary probability distribution. The professor gets wet when he starts from home with all the umbrellas at work, or starts from work with all the umbrellas at home, (the first and last states) and it is raining. The probability of this, in the long run or asymptotically is

$$\frac{1}{2} \cdot \frac{1}{2(2n + 1)} + \frac{1}{2} \cdot \frac{1}{2(2n + 1)} = \frac{1}{2(2n + 1)}.$$

All states are accessible from each other and therefore the Markov chain is irreducible. It is easy to multiply the transition probability matrix against itself, and see that $P_{ii}^2 = 1/2$ except for $i = 2$ and $i = 2n + 1$. Therefore, each state is periodic of period 2.

4:   The transition probability matrix is

$$P = \begin{array}{c} \\ OCO \\ OFO \\ OOO \\ OOV \\ VO \\ VV \end{array} \begin{array}{c} OCO \\ \end{array} \begin{pmatrix}
1/2 & 0 & 0 & 0 & 0 & 1/2 \\
0 & 1/2 & 0 & 0 & 1/4 & 1/4 \\
0 & 0 & 1/2 & 0 & 1/2 & 0 \\
0 & 0 & 0 & 1/2 & 1/2 & 0 \\
r/2 & q/2 & p/2 & 0 & 1/2 & 0 \\
r/6 & q/6 & 0 & p/6 & 0 & 1/2
\end{pmatrix}.$$

The stationary distribution is

$$\left(\frac{r}{2}, \frac{q}{2}, \frac{p(q + 2p)}{4}, \frac{p(2r + q)}{4}, \frac{q + 2p}{4}, \frac{2r + q}{4} \right).$$

5:   See the next solution with $v = 1/3$.

6:   Using the assumption that the bathroom is occupied $1 - v$ of the time, and vacant v of the time, all day, every day, the transition from state to state each minute is a Markov chain with transition probability

$$\begin{array}{c} \\ O \\ V \end{array} \begin{array}{cc} O & V \end{array} \begin{pmatrix}
1 - v & v \\
1 - v & v
\end{pmatrix}.$$

It is easy to see that this chain is irreducible, positive recurrent and aperiodic. The stationary distribution is $(1 - v, v)$, consistent with the assumption that the bathroom is occupied $1 - v$ of the time, all day, every day.

The possible transitions at each minute are:

(1) A Conscientious person is in the bathroom, and it is Occupied and the sign says Occupied. This state is OCO. Either the person stays and the state remains Occupied with sign Occupied with probability $1 - v$, or the conscientious person leaves and the state is VV with probability $r \cdot v$.

(2) A Forgetful person is in the bathroom, and it is Occupied and the sign says Occupied. This state is OFO. Either the person stays and the state remains Occupied with sign Occupied with probability $1 - v$, or the forgetful person leaves and the state is VO with probability $(1/2) \cdot v = v/2$ or the forgetful person leaves, changing the sign and the state is VV with probability $1/2 \cdot v = v/2$.

(3) An Oblivious person is in the bathroom, and it is Occupied and the sign says Occupied. Name this state as OOO. Either the person stays and the state remains Occupied with sign Occupied with probability $1 - v$, or the oblivious person leaves and the state is VO with probability v.

(4) A Oblivious person is in the bathroom, and it is Occupied but the sign says Vacant. Name this state as OOV. Either the person stays and the state remains Occupied with sign Vacant with probability $1 - v$, or the oblivious person leaves and the state is VV with probability v.

(5) The bathroom is Vacant, but the sign says Occupied. With probability $1 - v$ some person will approach, nevertheless try the door and find the true state of vacancy and enter, dealing with the sign according to their type. Then the transition is to OCO with probability $r(1-v)$, OFO with probability $q(1-v)$ and OOO with probability $p(1-v)$. With probability v the state remains VO.

(6) The bathroom is Vacant, and the sign says Vacant. With probability $1-v$ some person will approach, nevertheless try the door and easily find the true state of vacancy and enter, dealing with the sign according to their type. Then the transition is to OCO with probability $r(1-v)$, OFO with probability $q(1-v)$ and OOV with probability $p(1-v)$. With probability v the state remains VV.

The transition probability matrix is

$$
P = \begin{array}{c} \\ OCO \\ OFO \\ OOO \\ OOV \\ VO \\ VV \end{array}
\begin{array}{c} OCO \quad\quad OFO \quad\quad OOO \quad\quad OOV \quad\quad VO \quad VV \end{array}
\left(
\begin{array}{cccccc}
1-v & 0 & 0 & 0 & 0 & v \\
0 & 1-v & 0 & 0 & v/2 & v/2 \\
0 & 0 & 1-v & 0 & v & 0 \\
0 & 0 & 0 & 1-v & v & 0 \\
r(1-v) & q(1-v) & p(1-v) & 0 & v & 0 \\
r(1-v) & q(1-v) & 0 & p(1-v) & 0 & v
\end{array}
\right).
$$

The stationary distribution is:

$$
\left(r(1-v), q(1-v), \frac{p(q+2p)(1-v)}{2}, \frac{p(2r+q)(1-v)}{2}, \right.
$$

$$
\left. \frac{(q+2p)v}{2}, \frac{(2r+q)v}{2} \right).
$$

As $v \to 1$, the stationary distribution tends to $\left(0, 0, 0, 0, \frac{(q+2p)}{2}, \frac{(2r+q)}{2}\right)$. As $v \to 0$, the stationary distribution tends to $\left(r, q, \frac{p(q+2p)}{2}, \frac{p(2r+q)}{2}, 0, 0\right)$.

7:   Suppose the probability of an oblivious person entering the bathroom is w. Then proportion of time that the bathroom is occupied by an oblivious person is $p(1-w)$. Likewise, and as in the previous problem, the proportion of time the bathroom is occupied by forgetful and conscientious people is $q(1-v)$ and $r(1-v)$ respectively. If the probabilities of occupancy by forgetful and conscientious people are both equal to $1-v$ then the assumption that oblivious people spend twice as much time in the bathroom is $p(1-w) = 2(q+r)(1-v)$ or $w = 1 - \frac{2(q+r)}{p}(1-v) = 1 - \frac{2(1-p)}{p}(1-v)$. Since $0 < w < 1$, this puts some restrictions depending on p and v.

The transition probability matrix is

$$P = \begin{array}{c} \\ OCO \\ OFO \\ OOO \\ OOV \\ VO \\ VV \end{array} \begin{array}{c} OCO \\ \left(\begin{array}{c} 1-v \\ 0 \\ 0 \\ 0 \\ r(1-v) \\ r(1-v) \end{array} \right. \end{array} \begin{array}{c} OFO \\ 0 \\ 1-v \\ 0 \\ 0 \\ q(1-v) \\ q(1-v) \end{array} \begin{array}{c} OOO \\ 0 \\ 0 \\ 1-w \\ 0 \\ p(1-v) \\ 0 \end{array} \begin{array}{c} OOV \\ 0 \\ 0 \\ 0 \\ 1-w \\ 0 \\ p(1-v) \end{array} \begin{array}{c} VO \\ 0 \\ v/2 \\ w \\ w \\ v \\ 0 \end{array} \begin{array}{c} VV \\ v \\ v/2 \\ 0 \\ 0 \\ 0 \\ v \end{array} \left. \begin{array}{c} \\ \\ \\ \\ \\ \end{array} \right).$$

8:

(1) First consider the statement: If i and j communicate, and i is recurrent, then so is j.
   (a) If i and j communicate, then there is an $r > 0$ such that $(P^r)_{ji} > 0$ and an $s > 0$ such that $(P^s)_{ij} > 0$.
   (b) If i is recurrent, then $\sum_{n=1}^{\infty}(P^n)_{ii} = \infty$ so also $\sum_{n=r+s+1}^{\infty}(P^n)_{ii} = \infty$
   (c) $(P^n)_{jj} > (P^r)_{ji}(P^{n-r-s})_{ii}(P^s)_{ij}$, that is the probability of going from j to j is greater than the probability of an excursion from j to i, then from i to i, and from i to j.
   (d) Then

$$\sum_{n=1}^{\infty}(P^n)_{jj} > \sum_{n=r+s1}^{\infty}(P^r)_{ji}(P^{n-r-s})_{ii}(P^s)_{ij} =$$

$$(P^r)_{ji}\left(\sum_{n=r+s+1}^{\infty}(P^{n-r-s})_{ii}\right)(P^s)_{ij} = \infty$$

   Therefore, j is recurrent.
(2) Suppose i and j communicate, and i is transient, but j is recurrent. Since j communicates with i and j is recurrent, by the previous proof then i must be recurrent, a contradiction. Therefore, j must be transient.
(3) The statement follows from the definition of irreducible and the previous proofs.

9:   The recursive equation can be interpreted by considering paths for which $X_0 = i$ and $X_n = j$ and the first return to state j occurs at step k. The probability

that the first return is at step k is f_{ij}^k and the probability of then going from state j to state j in $n-k$ steps, independent of the past, is $(P^{n-k})_{jj}$. Adding over all the disjoint possible first arrivals from i to j gives the recursive equation.

10: Let $\epsilon > 0$ be given. Choose N_1 so large that $|a_n - a| < \epsilon/2$ for $n > N_1$. Choose $N_2 > N_1$ so large that $\frac{1}{n}\sum_{\nu=1}^{N_1} |a_\nu - a| < \epsilon/2$ for $n > N_2$. Then for $n > N_2$,

$$\left| \frac{1}{n} \sum_{\nu=1}^{n} a_n - a \right| \leq \frac{1}{n} \sum_{\nu=1}^{N_1} |a_n - a| + \frac{1}{n} \sum_{\nu=N_1+1}^{n} |a_n - a|$$

$$\leq \frac{1}{n} \sum_{\nu=1}^{N_1} |a_n - a| + \frac{1}{n} \sum_{\nu=1}^{n} |a_n - a|$$

$$< \frac{\epsilon}{2} + \frac{1}{n} \sum_{\nu=N_1+1}^{n} \frac{\epsilon}{2}$$

$$< \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.$$

Therefore

$$\lim_{n \to \infty} \frac{1}{n} \sum_{\nu=1}^{n} a_\nu = a,$$

11: It is immediate that $\pi_1 = \frac{1}{p}\pi_0$. By induction $\pi_i = \frac{(1-p)^{i-1}}{p^i}\pi_0$. The normalization condition is

$$\pi_0 \left(1 + \frac{1}{p} + \frac{(1-p)}{p^2} + \frac{(1-p)^2}{p^3} + \cdots \right) = 1.$$

Now is apparent why $p > 1/2$ is necessary for stationarity. The normalization simplifies to $\pi_0 \left(1 + \frac{1}{2p-1} \right) = 1$. Then $\pi_\nu = \frac{2p-1}{2p} \left(\frac{(1-p)^{\nu-1}}{p^\nu} \right)$ for $\nu \geq 1$.

1: This leads to the single-step transition matrix P partitioned into one absorbing state for "leave firm" and three transient states: junior lawyer, senior lawyer, partner

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.05 & 0.80 & 0.15 & 0 \\ 0.10 & 0 & 0.70 & 0.20 \\ 0.05 & 0 & 0 & 0.95 \end{pmatrix}.$$

The transient transition matrix is

$$T = \begin{pmatrix} 0.80 & 0.15 & 0 \\ 0 & 0.70 & 0.20 \\ 0 & 0 & 0.95 \end{pmatrix}.$$

The corresponding fundamental matrix is

$$N = (I_3 - T)^{-1} = \begin{pmatrix} 5 & 5/2 & 10 \\ 0 & 10/3 & 40/3 \\ 0 & 0 & 20 \end{pmatrix}.$$

The waiting times to absorption are then

$$(I - T)^{-1}\mathbf{1} = (35/2, 50/3, 20)^T.$$

Now compute the covariances for the first transient state, junior lawyer. Using

$$\text{diag } N_{(1,\cdot)} N + N^T \text{diag } N_{(1,\cdot)} - N_{(1,\cdot)}^T N_{(1,\cdot)} - \text{diag } N_{(1,\cdot)}$$

gives

$$\text{Cov}\,[Y_{1j}, Y_{1k}] = \begin{pmatrix} 20 & 0 & 0 \\ 0 & 95/12 & 25/3 \\ 0 & 25/3 & 290 \end{pmatrix} = \begin{pmatrix} 20 & 0 & 0 \\ 0 & 7.91667 & 8.33333 \\ 0 & 8.33333 & 290 \end{pmatrix}.$$

Since $w_i = \sum_{j=a+1}^{t} Y_{ij}$, using $\text{Var}\,[w_i] = \sum_{j=a+1}^{t} \sum_{l=a+1}^{t} \text{Cov}\,[Y_{1j}, Y_{1l}]$, the variance is the sum of the entries in the covariance matrix. Hence $\text{Var}\,[w_1] = 334.58$ (in units of year^2). The standard deviation is 18.29 years. As a point of special interest, note that the standard deviation is larger than the mean so that knowing the mean is not enough to understand how long a junior lawyer will be with the firm.

    2:   Theorem 3.3 gives the variances of the waiting times as

$$\text{Var}\,[\mathbf{w}] = (2N - I)N\mathbf{1} - (N\mathbf{1})_{\text{sq}}.$$

Using the matrix N from the test example, this is

$$\begin{pmatrix} \frac{551592}{7225} \\ \frac{476496}{7225} \\ \frac{83352}{1445} \\ \frac{15264}{289} \\ \frac{330312}{7225} \\ \frac{247536}{7225} \\ \frac{5400}{289} \end{pmatrix}$$

or approximately

$$\begin{pmatrix} 76.345 \\ 65.951 \\ 57.683 \\ 52.817 \\ 45.718 \\ 34.261 \\ 18.685 \end{pmatrix}$$

The entry for 0 rounds to 52.81661, the same as the variance in the text example.

    3:   The game neither adds nor subtracts the number of balls in use at any turn of the game, so $a + b + c = n$ is fixed. Then it is sufficient to keep track of the number of balls in urns A and B in any turn. Let $D_{a,b} = \mathbb{E}\,[[]\,N_{a,b}]$ be the expected duration of the game when the urn A has a balls, urn B has b balls, and urn C has $c = n - a - b$ balls. Using first-step analysis:

- urn A has $a+1$ balls, urn B has $b-1$ balls and urn C has $n - a - b$ balls with probability $1/6$,

- urn A has $a+1$ balls, urn B has b balls and urn C has $n - a - b - 1$ balls with probability $1/6$,

- urn A has $a-1$ balls, urn B has $b+1$ balls and urn C has $n - a - b$ balls with probability $1/6$,

- urn A has a balls, urn B has $b+1$ balls and urn C has $n-a-b-1$ balls with probability $1/6$,
- urn A has $a-1$ balls, urn B has b balls and urn C has $n-a-b+1$ balls with probability $1/6$,
- urn A has a balls, urn B has $b-1$ balls and urn C has $n-a-b+1$ balls with probability $1/6$.

Then

$$D_{a,b} = (1/6)[D_{a+1,b-1} + D_{a+1,b} + D_{a-1,b+1} + D_{a,b+1} + D_{a-1,b} + D_{a,b-1}] + 1$$

The boundary conditions are

(1) $D_{0,b} = 0$ for $0 \le b \le n$,

(2) $D_{a,0} = 0$ for $0 \le a \le n$,

(3) $D_{a,b} = 0$ for $a + b = n$.

The solution of this non-homogeneous two-variable difference equation, with these boundary conditions is

$$D_{a,b} = \frac{3}{n}ab(n - a - b).$$

This can be verified by substitution but that leaves open the question of how such a solution is obtained. To motivate the equation by guessing, note that the original problem is symmetric in the urns A, B and C, that is it does not matter which urn is labeled A. In turn, that means the solution must be symmetric in the variables a and b. Since the only symmetric polynomials in a and b are ab and $a + b$, it makes sense to search for a solution as a combination of these variables. Furthermore, the product $Kab(n-a-b)$ (where K is a constant to be determined) satisfies the boundary conditions. Finally, this product is similar to the solution $a(n - a)$ in the two urn, single variable case. Substituting $Kab(n - a - b)$ and determining the constant K reveals the solution.

4:   The equality is trivial for the base case $m = 1$. Assume the equality holds for $m - 1$. Check the multiplication for the lower left block, the other three blocks are easily seen to hold. The multiplication for the lower left block is

$$\left((I_t + T + T^2 + \cdots + T^{m-2})A\right)I_a + T^{m-1}A = (I_t + T + T^2 + \cdots + T^{m-1})A.$$

Note $AI_a = A$. Using associative and distributive properties for matrix multiplication completes the induction step.

5:

(a) With R, for $p = 1/2$, `norm(T, type=c("2"))` $= 0.924$. With R, for $p = 2/3$, `norm(T, type=c("2"))` $= 0.933$.

(b) With R, `norm(T, type=c("2"))` $= 0.794$.

(c) With R, `norm(T, type=c("2"))` $= 0.866$.

(d) Proof lightly adapted from https://math.stackexchange.com/questions/36828/substochastic-matrix-spectral-radius#666603 For any state i and integer $n \ge 0$, let $r_i^n = \sum_\nu (T^n)_{i\nu}$ denote the ith row sum of T^n. For $n = 1$, for convenience write

r_i rather than r_i^1. Since T is a substochastic transition probability matrix, $0 \leq r_i^n \leq 1$.

Let l^* be an index with $r_{l^*} < 1$, and note that for $n \geq 1$

$$r_{l^*}^n = \sum_l T_{l^*l} r_l^{n-1} \leq \sum_l T_{l^*l} = r_{l^*} < 1.$$

By irreducibility, for any i, there is an m with $(T^m)_{il^*} > 0$. In fact, if T is an $N \times N$ matrix, and $i \neq l^*$ then take $m < N$. (Take the shortest path from i to l^* with positive transition probability). Since $(T^m)_{il}$ puts positive weight on the index $l = l^*$,

$$r_i^N = \sum_l T_{il}^m r_l^{N-m} < r_i^m \leq 1.$$

That is, every row sum of T^N is strictly less than 1. Now it is possible to show $T^{jN} \to 0$ as $j \to \infty$ and this shows T^N (and hence T) cannot have any eigenvalue with modulus 1.

(e) Note that if $\mathbf{x} = (0, 1, 0, 0)^T$, then $T\mathbf{x} = (0, 1, 1/2, 0)$ with $\|T(x)\| = \sqrt{5}/2 \approx 1.118 > 1$. In fact, With R, `norm(T, type=c("2")) = 1.162` but the eigenvalues are 0.833, 0.500, 0 and 0. The previous theorem is not violated because the transient probability transition matrix is not irreducible, it is not possible to reach state $1+1+1+1$. Note also the converse of the previous theorem is not true: This matrix has operator norm less than 1, but it is not irreducible.

6: First note $NT = TN$ because

$$T(I_t - T) = (I_t - T)T$$
$$T = (I_t - T)T(I_t - T)^{-1}$$
$$(I_t - T)^{-1}T = T(I_t - T)^{-1}$$
$$NT = TN$$

Start with $N = (I_t - T)^{-1}$. Then

$$I_t = N(I_t - T)$$
$$0 = N - NT - I_t$$
$$T = N - TN - I_t + T$$
$$T = (I_t - T)N - (I_t - T)$$
$$T = (I_t - T)(N - I_t)$$
$$(I_t - T)^{-1}T = N - I_t$$

7: Consider

$$(I - T)(I + T + T^2 + \cdots + T^m) = (I + T + T^2 + \cdots + T^m) -$$
$$(T + T^2 + \cdots + T^{m+1}) = I - T^{m+1}$$

Then

$$\left\| (I - T) \sum_{\nu=0}^m (T^\nu) - I \right\| = \|T^{m+1}\| \to 0$$

so

$$\sum_{\nu=0}^{\infty}(T^{\nu}) = (I - T)^{-1}.$$

8:    With 1 candle, and the assumption you blow out a random number of candles between one and the number that remain, w_1 must be $1 = H_1$ and the base case is established. Next assume

$$w_N = H_N = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots + \frac{1}{N} = \sum_{\nu=1}^{N}\frac{1}{\nu}.$$

The first-step equation is

$$w_{N+1} = 1 + \frac{1}{N+1}w_1 + \frac{1}{N+1}w_2 + \cdots + \frac{1}{N+1}w_N$$

$$= 1 + \frac{1}{N+1}\sum_{\nu=1}^{N}w_\nu$$

$$= 1 + \frac{1}{N+1}\sum_{\nu=1}^{N+1}\sum_{j=1}^{\nu}\frac{1}{j}$$

$$= 1 + \frac{1}{N+1}\sum_{j=1}^{N}\frac{N-(j-1)}{j}$$

$$= 1 + \frac{1}{N+1}\sum_{j=1}^{N}\left(\frac{N+1}{j} - 1\right)$$

$$= 1 + \sum_{j=1}^{N}\frac{1}{j} - \frac{1}{N+1}(N)$$

$$= \sum_{j=1}^{N}\frac{1}{j} + \frac{1}{N+1}$$

$$= w_{N+1}$$

1:

(a) For the first Ehrenfest model

$$P_{ij} = \begin{cases} i/k & j = i - 1 \\ (k-i)/k & j = i + 1 \\ 0 & \text{otherwise} \end{cases}.$$

(b) For the second Ehrenfest model

$$P_{ij} = \begin{cases} \frac{i}{k}q & j = i - 1 \\ \frac{i}{k}p + \frac{k-i}{k}q & j = i \\ \frac{k-i}{k}p & j = i + 1 \\ 0 & \text{otherwise} \end{cases}.$$

(c) For the third Ehrenfest model

$$
P_{ij} = \begin{cases}
\frac{k-i}{k}p & j = i - 1 \\
\frac{i}{k}p + \frac{k-i}{k}q & j = i \\
\frac{i}{k}q & j = i + 1 \\
0 & \text{otherwise}
\end{cases}.
$$

(d) For the fourth Ehrenfest model

$$
P_{ij} = \begin{cases}
\frac{i(k-i)}{k^2} & j = i - 1 \\
\frac{i^2}{k^2} + \frac{(k-i)^2}{k^2} & j = i \\
\frac{i(k-i)}{k^2} & j = i + 1 \\
0 & \text{otherwise}
\end{cases}.
$$

2: Entry i must satisfy

$$
\binom{k}{i-1}\left(\frac{1}{2}\right)^k \cdot \frac{k-i+1}{k} + \binom{k}{i+1}\left(\frac{1}{2}\right)^k \cdot \frac{i+1}{k} = \binom{k}{i}\left(\frac{1}{2}\right)^k
$$

or more simply

$$
\binom{k}{i-1} \cdot \frac{k-i+1}{k} + \binom{k}{i+1} \cdot \frac{i+1}{k} = \binom{k}{i}.
$$

This reduces to

$$
\binom{k-1}{i-1} + \binom{k-1}{i+1} = \binom{k}{i}.
$$

which is the basic binomial (Pascal triangle) identity.

3: Using the probability transition matrix from `ehrnefest3.R` and using the `markovchain` package in R:

```
R> absorptionProbabilities(ehrenfest3)
         0        7
1 0.984375 0.015625
2 0.890625 0.109375
3 0.656250 0.343750
4 0.343750 0.656250
5 0.109375 0.890625
6 0.015625 0.984375
R> meanAbsorptionTime(ehrenfest3)
        1        2         3         4        5        6
 3.033333  7.233333 10.733333 10.733333  7.233333  3.033333
```

Using the probability transition matrix from `ehrnefest4.R` and using the `markovchain` package in R:

```
R> absorptionProbabilities(ehrenfest4)
          0         7
1 0.8571429 0.1428571
2 0.7142857 0.2857143
3 0.5714286 0.4285714
4 0.4285714 0.5714286
```

5 0.2857143 0.7142857
6 0.1428571 0.8571429
R> meanAbsorptionTime(ehrenfest4)
      1   2   3   4   5   6
17.15000 26.13333 30.21667 30.21667 26.13333 17.15000

4:

$$P = \begin{pmatrix} 1-p & p & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

and

$$(\pi_{1,0}, \pi_{0,1}, \pi_{0,2}) = \left(\frac{1}{1+2p}, \frac{p}{1+2p}, \frac{p}{1+2p} \right).$$

5: Let the state be the number i of white balls in the first urn. This determines the number of black balls in the first urn and the number of black and white balls in the second urn. $p_{ii} = 2i(m-i)/m^2$, $p_{i,i-1} = i^2/m^2$, $p_{i,i+1} = (m-i)^2/m^2$. After a large number of turns $\pi_j = \binom{m}{k}\pi_0$, where $1/\pi_0 = \sum_{i=0}^{m} \left(\binom{m}{i}\right)^2 = \binom{2m}{m}$ and $\pi_j = = \left(\binom{m}{i}\right)^2 / \binom{2m}{m}$.

6: The state space is the set of nonnegative integers. At time period t, suppose there are i infected individuals not in quarantine. Suppose that j ($0 \leq j \leq i$) of these contagious individuals infect another j individuals and that $i-j$ are discovered and moved to quarantine. At the next time period there are a net number $i+j-(i-j) = 2j$ infectives in the population with probability $\binom{i}{j}p^j(1-p)^j$. This is a discrete time Markov chain on an infinite state space with transition probability $p_{i\ell} = \binom{i}{\ell/2}p^{\ell/2}(1-p)^{i-(\ell/2)}$ for $i \geq 0$ and $0 \leq \ell \leq 2i$ and ℓ even.

The probability that by period 1 the one carrier has not been discovered is p, by period 2 there are 2 carriers and the probability that each is not yet discovered is p^2, by period three there are 4 carries with no discoveries with probability p^4, and by period n there are 2^{n-1} carriers with no discoveries $p^{2^{n-1}}$.

1: Use the following script:

```
library("markovchain")

rotvec <- function(vec) vec[ c(length(vec), 1:(length(vec)-1)) ]

nStates <- 8
monopolyStates <- as.character( c(1:nStates) )

diceRoll <- matrix(0, nStates, nStates)
p1 <- c(0, 1/2, 1/2, 0, 0, 0, 0, 0)

for (i in 1:nStates) {
    diceRoll[i, ] <- p1
    p1 <- rotvec(p1)
}

Jail <- diag(nStates)
Jail[4, 4] <- 0
Jail[4, 2] <- 1                         #go from Police to Jail

Ch <- diag(nStates)
Ch[6, 7] <- 1/16                        #Nearest Railroad
Ch[6, 8] <- 1/16                        #Advance to Go
Ch[6, 1] <- 1/16                        #Reading Railroad
Ch[6, 2] <- 1/16                        #Go to Jail
```

```
25 Ch[6, 3] <- 1/16                           #Go back 3
26 Ch[6, 6] <- 11/16                          #All other cards
27
28 monopMat <- diceRoll %*% Jail %*% Ch
29
30 mcMonopoly8 <- new("markovchain", states = monopolyStates, byrow = TRUE,
31                 transitionMatrix = monopMat, name = "Monopoly")
32 ss <- steadyStates(mcMonopoly8)
33 cat(ss, "\n")
```

.

The stationary distribution is

1	0.07731016
2	0.383806
3	0.2342177
4	0.0
5	0.1171089
6	0.04025618
7	0.08234218
8	0.06495883

1: For $Y^{(1)} = y^{(1)}$, $Y^{(2)} = y^{(2)}$ with $y^{(1)} > y^{(2)}$, the individual die values *in some order* must be $y^{(1)}$, $y^{(2)}$ and $y^{(3)} \leq y^{(2)}$. There are 6 orders for each of the $y^{(2)}$ possibilities for $y^{(3)}$, each with probability $\frac{1}{6^3}$. However, if $y^{(3)} = y^{(2)}$, then the 6 orders are counted twice, so half of them, specifically 3, must be subtracted from the total.

2: Use the order statistic probability $\mathbb{P}\left[Y^{(1)} = y^{(1)}\right] = (1 - 3y^{(1)} + 3\left(y^{(1)}\right)^2)/216$. The possibilities for the defender to lose one army are the following:

(a) The largest die of the 3 rolled by the attacker is a 2, and the value rolled by the defender is 1; occurring with probability $\frac{7}{216} \cdot \frac{1}{6} = \frac{7}{1296}$.

(b) The largest die of the 3 rolled by the attacker is a 3, and the value rolled by the defender is 1 or 2; occurring with probability $\frac{19}{216} \cdot \frac{2}{6} = \frac{38}{1296}$.

(c) The largest die of the 3 rolled by the attacker is a 4, and the value rolled by the defender is 1 or 2 or 3; occurring with probability $\frac{37}{216} \cdot \frac{3}{6} = \frac{111}{1296}$.

(d) The largest die of the 3 rolled by the attacker is a 5, and the value rolled by the defender is 1, 2, 3, or 4; occurring with probability $\frac{61}{216} \cdot \frac{4}{6} = \frac{244}{1296}$.

(e) The largest die of the 3 rolled by the attacker is a 6, and the value rolled by the defender is 1, 2, 3, 4, or 5; occurring with probability $\frac{91}{216} \cdot \frac{5}{6} = \frac{455}{1296}$.

(f) The total probability is $(7 + 38 + 111 + 244 + 455)/1296 = 855/1296$ (which reduces to $95/144$).

3:

20	19	18	17	16
4.441	4.277	4.111	3.943	3.773
15	14	13	12	11
3.601	3.427	3.249	3.068	2.883
10	9	8	7	6
2.692	2.498	2.293	2.085	1.854
5	4	3	2	1
1.631	1.335	1.110	0.590	0.417

1: The transition probability matrix for *Tenzi* (rounded to three decimal places) is written in Maxima as

```
pi[i,j] := if (j >= i) then  binomial(11-i, j-i) * ((1/6)^(j-i)) *
((5/6)^(11-j)) else 0;
P : genmatrix(pi, 11, 11);
fpprintprec : 3;
float(P);
```

Transfer this matrix of values to R and evaluate with the following script:

```
x   <-  c(0.1615055828898457, 0.3230111657796915, 0.2907100492017223,
          0.1550453595742519, 0.05426587585098816, 0.01302381020423716,
          0.002170635034039527, 2.48072575318803*10^-4,
          1.860544314891023*10^-5, 8.2690858439601*10^-7,
          1.65381716879202*10^-8, 0.0, 0.1938066994678149,
          0.3488520590420667, 0.2790816472336534, 0.1302381020423716,
          0.03907143061271148, 0.007814286122542296,
          0.001041904816338973, 8.930612711476907*10^-5,
          4.465306355738455*10^-6, 9.922903012752121*10^-8, 0.0, 0.0,
          0.2325680393613778, 0.3721088629782046, 0.2604762040847432,
          0.1041904816338973, 0.02604762040847432, 0.004167619265355891,
          4.167619265355891*10^-4, 2.381496723060509*10^-5,
          5.953741807651273*10^-7,0.0, 0.0, 0.0, 0.2790816472336534,
          0.3907143061271148, 0.2344285836762689, 0.07814286122542295,
          0.01562857224508459, 0.001875428669410151,
          1.250285779606767*10^-4, 3.572245084590763*10^-6,0.0, 0.0,
          0.0, 0.0, 0.3348979766803841, 0.4018775720164609,
          0.2009387860082305, 0.05358367626886145, 0.008037551440329218,
          6.430041152263375*10^-4, 2.143347050754458*10^-5,0.0, 0.0,
          0.0, 0.0, 0.0, 0.4018775720164609, 0.4018775720164609,
          0.1607510288065844, 0.03215020576131687, 0.003215020576131687,
          1.286008230452675*10^-4,0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.4822530864197531, 0.3858024691358025, 0.1157407407407407,
          0.0154320987654321, 7.716049382716049*10^-4,0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.5787037037037037, 0.3472222222222222,
          0.06944444444444445, 0.004629629629629629,0.0, 0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.6944444444444444, 0.2777777777777778,
          0.02777777777777778,0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
          0.0, 0.8333333333333334, 0.1666666666666667,0.0, 0.0, 0.0,
          0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0)
P   <-  t( matrix(x, 11, 11) )

PPow   <- P
tenzicdf   <- c(PPow[1,11])
for (i in 2:50) {
  PPow   <- PPow %*% P;
  tenzicdf   <- c(tenzicdf, PPow[1,11])
}
```

```
library(ggplot2)
ggplot(data = as.data.frame(tenzicdf), aes(x=1:50, y=tenzicdf))
  + geom_point()
  + geom_line()

tenzipdf  <-  tenzipdf[2:50] - tenzipdf[1:49]
tenzipdf  <- c(tenzicdf[1], tenzipdf)
ggplot(data = as.data.frame(tenzipdf), aes(x=1:50, y=tenzipdf))
    + geom_point()
    + geom_line()

sum(tenzipdf)
tenzimean  <- (1:50) %*% tenzipdf

Q  <-  P[1:10, 1:10]
I10  <-  ident(10)
N  <- solve( I10 - Q, I10)
N %*% rep(1,10)

Varw  <- ((2 * N - I10) %*% N) %*% rep(1, 10) - ( N %*% rep(1, 10) )^2
sdw  <- sqrt(Varw)
```

   2:

(1) Denote as $P(n, j)$ the probability that the minimum value from n random samples of $p(j)$ is less than or equal to j. Then $P(n, j)$ is the complement of the probability that all n random samples (which are independent of each other) are greater than j, so $P(n, j) = 1 - (1 - P(j))^n$. The p.d.f. giving the probability that a game with n players is won in exactly j rolls is $p(n, j) = P(n, j) - P(n, j - 1)$.

(2) The expected number of turns in a game of Tenzi with 2 players is 11.897, with 3 players is 10.408, and with 4 players is 9.542.

   3:   A possible path is to roll a 1, to 38; roll a 6, to 44; roll a 6, to 50; roll a 1, to 67; roll a 6, to 73; roll a 6, to 79 and finally, roll a 1 to ascend from 80 to 100. Examination of the first 6 powers of the transition probability matrix show that the probability of 6 step paths from 38 is 0.007265947 but 0 for fewer than 6 steps.

   Another possible path is roll a 4, to 14; roll a 6, to 20; roll a 6, to 26; roll a 2, to 84; roll a 6, to 90; roll a 5, to 75 and finally, roll a 5 to ascend from 80 to 100. Examination of the first 6 powers of the transition probability matrix show that the probability of 6 step paths from 14 is 0.001800412 but 0 for fewer than 6 steps.

   4:

(a) The expected length of a game with no chutes or ladders is 31.1 with a standard deviation of 5.98.

(b) The expected length of the game drops from about 39.2 to about 29.8 with a standard deviation of about 18.8.

	AB	*S*	*D*	*T*	*HR*	*BB*	*AVE*	*SA*	*OB*	*OPS*
Ruth	8399	1517	506	136	714	2062	0.342	0.690	0.474	1.164
Williams	7706	1537	525	71	521	2021	0.344	0.634	0.481	1.114

Table 1. Statistics for Babe Ruth and Ted Williams.

(c) The game lengthens to about 45.8 moves with a standard deviation of about 33.4.

(d) Lengthens the expected playing time by more than two moves to about 41.9, since it increases the chances of missing the important ladder from state 80 to 100.

(e) Shortens the game by over a move to about 38.0, since it gives a second chance at the long ladder from 28 to 84.

5: The win probability if only square 2 is blue s 0.5288 The win probability if squares 2, 3 and 5 are blue is 0.7761. The remaining questions make a good project for student investigation.

1: In 2019, Strasburg was a pitcher for the Washington Nationals in the National League, as he had been since 2010. As a pitcher, his hitting skills and run production are not emphasized, it is his pitching that makes him a valuable player. Furthermore, as a pitcher, he appears in fewer games with even fewer plate appearances. That means he has a smaller base of hitting statistics for creating the matrix P so P may not fully reflect his hitting skills.

2: Using the R script with Trout's statistics, the expected time to absorption starting from the first state of no-outs is 5.22 at-bats per half-inning.

3: In the 25×25 matrix M_K, all entries are 0 except for the following:

- For each i, $i = 1, \ldots, 25$ and $j = 1, 2, 3$ with $j = i \mod 3$, $(M_K)_{ij} = q$,
- For each i, $i = 1, \ldots, 25$ with $i \neq 0 \mod 3$, and $j = i + 1$ $(M_K)_{ij} = 1 - q$
- For each i, $i = 1, \ldots, 25$ with $i = 0 \mod 3$ and $j = 25$ $(M_K)_{ij} = 1 - q$.

In the special case $q = 0.1$, the Markov runs statistic for a half-inning is 0.333 and the markov RPG is 3.

4: Depends on the player statistics.

5: The lifetime career statistics for Babe Ruth (1914–1935) and Ted Williams (1939–1942, 1946–1960) are in Table 1. (Values derived from Baseball-Reference.) Then

$$\rho_{BR} = (0.528, 0.197, 0.145, 0.048, 0.13, 0.068)$$

and

$$\rho_{TW} = (0.568, 0.212, 0.158, 0.054, 0.007, 0.054).$$

Using these probabilities in the script for Markov runs, the markov RPG for Babe Ruth is 12.65 and the value for Ted Williams is 12.55 The two values are close but Ruth has the slightly greater value.

No statistic can resolve the comparison between these two great players. Each played in different baseball eras, and each had particular strengths. Besides, what

is to be the basis for comparison, their lifetime statistics used here, or the best career year for each? If the latter, how to choose the best career year? Note that Williams lost 3 (possibly best) years of his career to military service in World War II. It is interesting to compute the Markov runs for each, but the statistic won't answer this opinion question.

1: Label the transient states in Figure 1 from 1 for 0:0 from to bottom, left to right and label the absorbing states last. Then 0:0 is state 1, state 40:30 is state 13, state 30:40 is state 15. The absorbing state "A wins" is 17 and "B wins" is 16. Use the script with this labeling.

For $p = 0.6$, the probability A wins is 0.736. For $p = 0.6$, the expected number of serves or points until absorption, that is the game ends, is 6.48 with a standard deviation of 2.59.

For $p = 0.51$, the probability A wins is 0.525. For $p = 0.51$, the expected number of serves or points until absorption, that is the game ends, is 6.75, only slightly longer. The standard deviation is 2.77, also slightly more.

The conclusion is that most games lasts between 4 and 9 serves, even if the players are nearly evenly matched. This makes sense, the game reaches the bottom row in 4 to 6 steps in any case and then is in the absorbing walk for a few steps.

2: Using the R script,

```
Mean A Scores:  89.52 Standard Deviation:  7.038825
Mean B Scores:  100.54 Standard Deviation:  7.428338
```

10.11. Algorithms and Scripts

Data: initial Susceptibles, Infectives, probability, time
Result: Plot of Susceptibles, Infectives versus time

Algorithm.

1 *Initialization*
2 Set initial Susceptibles, Infectives, probability, time
3 Loop over time
4 Get random binomial variate from number of Susceptible
5 Update the infectives
6 **return** *Plot of Susceptibles, Infectives versus time*
Algorithm 12: Simulation of epidemics with Greenwood and Reed Frost models.

Scripts.

R: R script for Greenwood model.

```
 1 S0 <- 999
 2 I0 <- 1
 3 p  <- 0.2
 4
 5 T <- 30
 6 S <- numeric(T + 1)
 7 I <- numeric(T + 1)
 8
 9 S[1]  <- S0
10 I[1]  <- I0
11 for (t in 2:(T + 1)) {
```

```
12      I[t]  <- rbinom(1, size=S[t-1], prob=p)
13      S[t] <-  S[t - 1] - I[t]
14 }
15
16 plot(1:(T + 1), S, col="blue",
17      xlim = c(0, (T + 1)), ylim=c(0, (S0 + I0)),
18      main="Greenwood Epidemic Model",
19      xlab="Time", ylab="S, I")
20 points(1:(T + 1), y=I, col="red")
```

.

R script for Reed-Frost model.

```
 1 S0 <- 9999
 2 I0 <- 1
 3 p   <- 0.00001
 4 q   <- 1 - p
 5
 6 T <- 25
 7 S <- numeric(T + 1)
 8 I <- numeric(T + 1)
 9
10 S[1]  <- S0
11 I[1]  <- I0
12 for (t in 2:(T + 1)) {
13      QT = q^I[t-1]
14      I[t]  <- rbinom(1, size=S[t-1], prob=1-QT)
15      S[t] <-  S[t - 1] - I[t]
16 }
17
18 plot(1:(T + 1), S, col="blue",
19      xlim = c(0, (T + 1)), ylim=c(0, (S0 + I0)),
20      main="Reed-Frost Epidemic Model",
21      xlab="Time", ylab="S, I")
22 points(1:(T + 1), y=I, col="red")
```

.

10.12. Problems to Work for Understanding

1: Write a script that simulates a place-based epidemic model. For this modification, assume that all the individuals are standing next to each other on a two-dimensional grid. Let one individual be an infective case. An infective can now probabilistically infect only his immediate neighbors on the grid. In the next time period, those people who got sick now have a chance to infect their neighbors and so on. Display on a two-dimensional grid that shows all the individuals what state they are in (i.e., Susceptible, Infective, or Recovered), with an animation if possible. Run the script many times with different input parameters. How does the pattern of the epidemic flow change with the contact probability?

Reversible Markov Chains

Assume X is an irreducible and positive recurrent chain, started at its unique invariant distribution π. This means X_0 has distribution π, and all successive X_n have distribution π as well. Now suppose that, for every n, the sequence of random variables $X_1, X_2, \ldots, X_n$ has the same joint distribution as the time-reversed sequence $X_n, X_{n-1}, \ldots, X_1$. Then the chain X is **reversible** and its invariant distribution π is also said to be reversible. This means a recorded simulation of a reversible chain looks the same if the "video" is run backwards.

Mathematically Mature: may contain mathematics beyond calculus with proofs.

11.1. Chapter Starter Question

Imagine a video is made of the Ehrenfest urn model started in its stationary distribution. Would it be possible to distinguish whether the video is played forward or in reverse simply by viewing the video and observing the movement of the balls between the urns?

11.2. Detailed Balance Equations

If the sequence of random variables $X_1, X_2, \ldots, X_n$ from a Markov chain has the same joint distribution as the time-reversed sequence $X_n, X_{n-1}, \ldots, X_1$, then the chain X is **reversible**. This means a recorded simulation of a reversible chain looks the same if the "video" is run backwards. The scripts in this chapter illustrate this property with the weather in Oz example. Reversing a long sample path from the weather Markov chain, then using the Maximum Likelihood Estimator

$$\hat{p}_{ij}^{\mathrm{MLE}} = \frac{n_{ij}}{\sum_\nu n_{i\nu}}$$

where n_{ij} is the number of transitions from $X_n = i$ to $X_{n+1} = j$ gives approximately the original transition probability matrix. This indicates, within the limits of the

simulation and estimation, that the reversed chain has the same joint distribution as the original weather chain. See [**Spe17**] for more details.

Proposition 11.1. *If a Markov chain is started in π, then the time-reversed chain has the Markov property.*

Proof.

$$\mathbb{P}\left[X_\ell = i \mid X_{\ell+1} = j, X_{\ell+2} = i_{\ell+2}, \ldots, X_n = i_n\right]$$
$$= \frac{\mathbb{P}\left[X_\ell = i, X_{\ell+1} = j, X_{\ell+2} = i_{\ell+2}, \ldots, X_n = i_n\right]}{\mathbb{P}\left[X_{\ell+1} = j, X_{\ell+2} = i_{\ell+2}, \ldots, X_n\right]}$$
$$= \frac{\pi_i P_{ij} P_{ji_{\ell+2}} \cdots P_{i_{n-1}i_n}}{\pi_j P_{ji_{\ell+2}} \cdots P_{i_{n-1}i_n}}$$
$$= \frac{\pi_i P_{ij}}{\pi_j}$$

which depends only on i and j, so the time-reversed chain has the Markov property.

$\square$

Remark 11.2. The definition is inconvenient because it requires the stationary distribution π in advance to check if the chain is time-reversible. The following theorem avoids having to know π in advance and can even help find π, see Theorem 11.8.

Theorem 11.3 (Reversibility Condition)**.** *A Markov chain is reversible if and only if the state space has a probability distribution π such that*

$$\pi_i P_{ij} = \pi_j P_{ji}$$

for all i and j and then π is the unique stationary distribution.

Remark 11.4. The condition

$$\pi_i P_{ij} = \pi_j P_{ji}$$

is the **detailed balance equation**.

Proof. ($\Rightarrow$)**:** From the proof of the Proposition, the time-reversed sequence of random variables satisfies

$$\mathbb{P}\left[X_\ell = i \mid X_{\ell+1} = j\right] = \frac{\pi_i P_{ij}}{\pi_j}.$$

If the Markov chain is reversible, the previous expression must be the same as the forward transition probability $\mathbb{P}\left[X_{\ell+1} = i \mid X_\ell = j\right] = P_{ji}$. Then the detailed balance equation holds.

($\Leftarrow$)**:** From the proposition, the time-reversed chain is Markov. If the original and the time-reversed chain both start at the same invariant distribution and satisfy the detailed balance, then the transition probabilities are the same forward or backward. Then by definition the chain is reversible.

$\square$

Remark 11.5. The condition says that starting from the stationary distribution

$$\mathbb{P}\left[X_n = i, X_{n+1} = j\right] = \mathbb{P}\left[X_n = j, X_{n+1} = i\right]$$

so the chain transitions from i to j as often as it transitions from j to i.

Example 11.6. If the $k \times k$ transition probability matrix is symmetric, then $\pi = (1/k, \ldots, 1/k)$ satisfies the detailed balance equation and the corresponding Markov chain is reversible. $\triangledown$

Remark 11.7. If there exist 2 states such that $P_{ij} > 0$ but $P_{ji} = 0$ then the detailed balance equation fails, since the limiting probabilities of all entries of an irreducible and positive recurrent chain are nonzero. Then the chain is not reversible.

Theorem 11.8. *Assume the Markov chain is irreducible and positive recurrent and probability distribution π satisfies the detailed balance equations*

$$\pi_i P_{ij} = \pi_j P_{ji}$$

(1) The distribution π is the stationary distribution of the chain.

(2) The chain is reversible.

Proof. Using the detailed balance equations

$$\sum_j \pi_j P_{ji} = \sum_j \pi_i P_{ij} = \pi_i \sum_j P_{ij} = \pi_i.$$

Since the chains is irreducible and positive recurrent, then π must be the unique stationary distribution. $\square$

11.3. Kolmogorov's Loop Criterion

A transition probability matrix is *reversible* if the corresponding Markov chain is reversible.

Definition 11.9. The equality condition

(11.1) $$P_{j_0 j_1} P_{j_1 j_2} \cdots P_{j_{\ell-1} j_\ell} P_{j_\ell j_0} = P_{j_0 j_\ell} P_{j_\ell j_{\ell-1}} \cdots P_{j_2 j_1} P_{j_1 j_0}$$

is called **Kolmogorov's loop criterion**. $\diamond$

Theorem 11.10 (Kolmogorov loop criterion)**.** *An irreducible and positive recurrent Markov chain is reversible if and only if*

$$P_{j_0 j_1} P_{j_1 j_2} \cdots P_{j_{\ell-1} j_\ell} P_{j_\ell j_0} = P_{j_0 j_\ell} P_{j_\ell j_{\ell-1}} \cdots P_{j_2 j_1} P_{j_1 j_0}$$

for every finite sequence of distinct states $j_0, j_1, \ldots, j_\ell$.

Proof. ($\Rightarrow$): An irreducible and positive recurrent Markov chain has a stationary distribution π. Since the matrix is reversible $P_{ij} = \frac{\pi_j P_{ji}}{\pi_i}$ for every i and j. Substituting this relation for every probability on the left side of the criterion

$$P_{j_0 j_1} P_{j_1, j_2} \cdots P_{j_{\ell-1} j_\ell} P_{j_\ell, j_0} =$$
$$\frac{\pi_{j_1} P_{j_1 j_0}}{\pi_{j_0}} \cdot \frac{\pi_{j_2} P_{j_2 j_1}}{\pi_{j_1}} \cdots \frac{\pi_{j_\ell} P_{j_\ell j_{\ell-1}}}{\pi_{j_{\ell-1}}} \cdots \frac{\pi_{j_0} P_{j_0 j_\ell}}{\pi_{j_\ell}}.$$

Canceling the π_i factors and reversing the order of multiplication gives the right side of the Kolmogorov loop criterion.

($\Leftarrow$): Assume the loop criterion holds. Fix states j_ℓ and j_0. Then

$$\mathbb{P}\left[X_n = j_\ell, X_{n-1} = j_{\ell_{n-1}}, \ldots, X_2 = j_2, X_1 = j_1 \mid X_0 = j_0\right] =$$

$$P_{j_0 j_1} P_{j_1 j_2} \cdots P_{j_{n-1} j_\ell} = \frac{P_{j_0 j_\ell}}{P_{j_\ell j_0}} P_{j_{n-2} j_{n-3}} \cdots P_{j_3 j_2} P_{j_2 j_1} P_{j_1 j_0} =$$

$$\frac{P_{j_0 j_\ell}}{P_{j_\ell j_0}} \mathbb{P}\left[X_n = j_0, X_{n-1} = j_1, \ldots, X_2 = j_{n-2}, X_1 = j_{n-1} \mid X_0 = j_\ell\right]$$

where the second equality is a rearrangment of the loop criterion. Now sum both sides over all choices of the intermediate states $j_1, j_2, \ldots, j_{\ell_{n-1}}$ to obtain the following relation among the n step transition probabilities

$$P_{j_0 j_\ell}^n = \frac{P_{j_0 j_\ell}}{P_{j_\ell j_0}} P_{j_\ell j_0}^n.$$

Multiply by $P_{j_\ell j_0}$ and take the limit in n on both sides and use that the limits of the multi-step transition probabilities are the stationary probabilities, guaranteed to exist since the Markov chain is irreducible and positive recurrent. The result is

$$\pi_{j_0} P_{j_0 j_\ell} = \pi_{j_\ell} P_{j_\ell j_0}$$

which is exactly the reversibility criterion.

$\square$

Remark 11.11. In words, Kolmogorov's loop criterion says a Markov transition matrix is reversible if and only for every loop of distinct states, the forward loop probability product equals the backward loop probability product.

For a two-state Markov chain Kolmogorov's loop criterion is always satisfied since $P_{12}P_{21} = P_{12}P_{21}$. If the transition matrix is symmetric, then $P_{ij} = P_{ji}$ for all i, j, so Kolmogorov's loop criterion is always satisfied and the chain is reversible.

Usually loop checking involves much computational work. The obvious reason is that the number of loops that need to be checked grows very quickly with n where n is the number of states. See Table 1. The following Proposition counts the number of equations that must be checked to apply Kolmogorov's loop criterion.

Proposition 11.12. *For an k state Markov chain, with $k \geq 3$, the number of equations that must be checked for reversibility by Kolmogorov's method is*

$$\sum_{\nu=3}^{n} \binom{n}{\nu} \frac{(\nu-1)!}{2}.$$

Proof.   (1) For a 3-state Markov chain, only 1 equation

$$P_{12}P_{23}P_{31} = P_{13}P_{32}P_{21}$$

needs checking, since all 2 step loops automatically satisfy the criterion and any other length 3 loop over the same states results in the same equation.

k	1	2	3	4	5	6	7	8	9	10
equations	0	0	1	7	37	197	1,172	8,018	62,814	556,014

Table 1. Number of equation to be checked for a Markov chain with k states.

(2) For $n = 4$, all 2 step loops automatically satisfy the criterion. Next check each loop of 3 states. For 3 state loops, choose any 3 of 4 states and there is one equation for each. Finally, for each loop of 4 states, fix the starting state. Then the other 3 states have 3! possible orders. However, the other side of the equation is just the reversed path, so there are only $\frac{3!}{2}$ paths involving 4 states with the first state fixed. In total, the number of loop equations is

$$\binom{4}{3} + \binom{4}{4}\frac{3!}{2}$$

which can be written as

$$\binom{4}{3}\frac{(3-1)!}{2} + \binom{4}{4}\frac{(4-1)!}{2}$$

which corresponds to the conclusion of the proposition.

(3) The previous argument easily generalizes to larger values of n.

$\square$

Next define operations to transform transition matrices in such a way as to preserve their reversibility status (either reversible or non-reversible). These transformations will be useful in creating new reversible Markov chains from existing ones, and for checking reversibility of Markov chains. A *row multiplication operation* on row i of a Markov transition matrix is the multiplication of row i by a positive constant leaving the sum of the non-diagonal elements at most 1, followed by an adjustment to p_{ii} to make the row sum exactly to 1. A *column multiplication operation* on column j of a Markov transition matrix is the multiplication of column j by a positive constant of allowable size (so no row sums exceed 1) followed by adjustments to all diagonal entries to make every row sum exactly 1.

Lemma 11.13. *A Markov chain matrix P maintains its reversibility status after a row multiplication operation or a column multiplication operation.*

Proof. (1) The Kolmogorov loop criterion states that P is reversible if and only if for all loops of distinct states, the forward loop probability product equals the backward loop probability product.

(2) Let the ith row of P correspond to state i. If a loop does not include state i, then a multiplication row operation on row i has no effect on the forward and backward loop products.

(3) Otherwise, state i appears in the first subscript of a forward loop probability if and only if it appears in the first subscript in the corresponding adjacent transition probability in the backward loop probability. See equation (11.1).

(4) So the row operation will have an identical effect on both sides of the loop product.

(5) A similar conclusion holds for column product operations.

□

Remark 11.14. Let $P^{(\star)}$ be the matrix resulting from a row (or column) multiplication operation on P, then the limiting probabilities for $P^{(\star)}$ in the lemma are generally different from the limiting probabilities for P.

If the matrix P is an $k \times k$ transition probability matrix, then at most $k-1$ row or column multiplication operations will determine if P is reversible.

Data: Transition probability matrix P
Result: Symmetric matrix $P^{(\star)}$
1 Pick two symmetric positions in P with non-zero entries, say $P_{i_1 i_2}$ and
      $P_{i_2 i_1}$.
2 $S \leftarrow \{i_1 i_2\}$.
3 $j \leftarrow i_1, \ell \leftarrow i_2$
4 repeat
5 | **if** $P_{j\ell} = P_{\ell j}$ **then**
6 | | Move to the next step.
7 | **else if** $P_{j\ell} < P_{\ell j}$. **then**
8 | | Multiply *row* ℓ by $P_{j\ell}/P_{\ell j}$
9 | | Adjust $P_{\ell\ell}$ to make the ℓ row sum to 1.
10 | **else if** $P_{j\ell} > P_{\ell j}$ **then**
11 | | Multiply *column* ℓ by $P_{\ell j}/P_{j\ell}$
12 | | Adjust $P_{\ell\ell}$ to make the ℓ *row* sum to 1.
       | `// The new matrix` $P^{(\star)}$ `will now have` $P_{j\ell}^{(\star)} = P_{\ell j}^{(\star)}$.
       | Choose another state $i_3 \notin S$ which has nonzero transition probabilities
       | to a state in S.
13 | $j \leftarrow i_3 \ \ell \notin S$ with $P_{i_3\ell} \neq 0$
14 | $S \leftarrow \{i_1, i_2, i_3\}$.
15 until *There are no more states left to add*
      `// After` $k-1$ `steps` $S = \{1, \dots, k\}$.
      Let $P^{(\star)}$ be the final matrix.

Theorem 11.15. *Let P be a $k \times k$ transition matrix to which the Algorithm is applied, resulting in $P^{(\star)}$. Then P is reversible if and only if $P^{(\star)}$ is symmetric.*

Proof. (1) ($\Rightarrow$)
      (a) Assume P is reversible, then by the lemma, $P^{(\star)}$ is reversible.
      (b) Let $\phi = (\phi_1, \dots, \phi_k)$ be the stationary distribution for $P^{(\star)}$.
      (c) Note that $P^{(\star)}$ is formed so that $P_{ij}^{(\star)} = P_{ji}^{(\star)}$ for *a particular subcollection*
          of (i, j) pairs which will include each of $1, 2, \dots, k$ *somewhere* among the
          subcollection of transitions (i, j).
      (d) Let (i, j) be one of the transition pairs in the collection. Since $P^{(\star)}$ is
          reversible, the detailed balance equations $\phi_i P_{ij}^{(\star)} = \phi_j P_{ji}^{(\star)}$ must hold for
          the pair (i, j).
      (e) Using the equality from the symmetry, $\phi_i = \phi_j$.

(f) But all the states $1, \ldots, k$ appear in the set of transitions somewhere in the collection, so $\phi_1 = \phi_2 = \cdots = \phi_k$.

(g) Now take an arbitrary pair (i, j). Since $P^{(\star)}$ is reversible, the detailed balance holds for all i, j. Hence $\phi_i P_{ij}^{(\star)} = \phi_j P_{ji}^{(\star)}$ for all i, j.

(h) Since $\phi_i = \phi_j$, for all pairs (i, j) then $P_{ij}^{(\star)} = P_{ji}^{(\star)}$, so $P^{(\star)}$ is symmetric.

(2) ($\Leftarrow$)

If $P^{(\star)}$ is symmetric, then $P^{(\star)}$ is reversible, so the by the lemma P is reversible.

$\square$

Remark 11.16. The Algorithm chooses the smaller of 2 matrix entries to change the matrix P. Also corrections are made to the diagonal elements in order to ensure the rows sum to 1. In fact, by examining the loop equation (11.1), correction of the diagonal elements is not necessary and the transformed matrix $P^{(\star)}$ no longer needs to be a transition matrix. The proof shows the important issue is whether $P^{(\star)}$ is symmetric or not.

Example 11.17. Let

$$P = \begin{pmatrix} 17/40 & 0 & 3/40 & 1/2 \\ 0 & 11/20 & 1/4 & 1/5 \\ 3/10 & 1/4 & 9/20 & 0 \\ 1/2 & 1/20 & 0 & 9/20 \end{pmatrix}.$$

To check for reversibility, transform P by column or row operations. First, the zeroes of P are symmetric, and $P_{14} = P_{41}$, so symmetry for states $\{1, 4\}$ already exists. Now make the $(1, 3)$ entry match the $(3, 1)$ entry without losing the $(1, 4)$ and $(4, 1)$ symmetry. It is possible to multiply column 3 by $(3/10)/(3/40) = 4$ or row 3 by $(3/40)/(3/10) = 1/4$. Make the latter choice to obtain a new matrix with row 3 equal to $(3/40, 1/16, 9/80, 0)$. This is not a transition probability matrix because the row no longer sums to 1, so change the diagonal element at $(3, 3)$ to $1 - 3/40 - 1/16 = 69/80$. Now the transformed matrix is

$$P^{(1)} = \begin{pmatrix} 17/40 & 0 & 3/40 & 1/2 \\ 0 & 11/20 & 1/4 & 1/5 \\ 3/40 & 1/16 & 69/80 & 0 \\ 1/2 & 1/20 & 0 & 9/20 \end{pmatrix}.$$

Now the set of included states is $S = \{1, 4, 3\}$, so state 2 needs to be included. To preserve the values in $(1, 4)$ and $(4, 1)$, and $(1, 3)$ and $(3, 1)$, only row 2 or column 2 can be changed. Choose to multiply column 2 by 4 to make entries $(4, 2)$ and $(2, 4)$ equal. The result is

$$P^{(2)} = \begin{pmatrix} 17/40 & 0 & 3/40 & 1/2 \\ 0 & 11/5 & 1/4 & 1/5 \\ 3/40 & 1/4 & 69/80 & 0 \\ 1/2 & 1/5 & 0 & 9/20 \end{pmatrix}.$$

Adjust the entries in the diagonal positions in rows 2, 3, 4 to make the matrix a transition probability matrix

$$P^{(\star)} = \begin{pmatrix} 17/40 & 0 & 3/40 & 1/2 \\ 0 & 11/20 & 1/4 & 1/5 \\ 3/40 & 1/4 & 27/40 & 0 \\ 1/2 & 1/5 & 0 & 3/10 \end{pmatrix}.$$

Now P is reversible if and only if $P^{(\star)}$ is reversible. But $P^{(\star)}$ is symmetric so it is automatically reversible. Using the detailed balance equations makes it easy to compute the stationary distribution of P, see the exercises.

$\triangledown$

11.4. Examples of Reversible Chains

The Weather in Oz. For the familiar example of the weather in Oz, the transition probability matrix is

$$P = \begin{array}{c} \\ R \\ N \\ S \end{array} \begin{array}{c} \begin{matrix} R & N & S \end{matrix} \\ \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/2 \end{pmatrix}. \end{array}$$

Using the stationary distribution $(\frac{2}{5}, \frac{1}{5}, \frac{2}{5})$, the detailed balance equations show that the Markov chain is reversible. See the exercises.

The single equation to check for the Kolmogorov loop criterion is

$$P_{RN} P_{NS} P_{SR} = P_{RS} P_{SN} P_{NR},$$

that is,

$$\frac{1}{4} \cdot \frac{1}{2} \cdot \frac{1}{4} = \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{2}$$

which is true, so the Markov chain is reversible.

The state N communicates with the other two states, so the criterion in the exercises also shows this Markov chain is reversible.

Negative Drift Random Walk on the Non-negative Integers. Consider a negative drift simple random walk, restricted to be non-negative by a reflecting boundary. That is, $P_{01} = 1$ and otherwise $P_{i,i+1} = p < 0.5$, $P_{i,i-1} = 1-p > 0.5$ for $i > 0$. The time reversibility equations are

$$\pi_0 = (1-p)\pi_1$$
$$p\pi_i = (1-p)\pi_{i+1}$$

so $\pi_1 = \pi_0/(1-p)$, $\pi_2 = p\pi_0/(1-p)^2$ and in general $\pi_n = p^{n-1}\pi_0/(1-p)^n$. Since $\sum_\nu \pi_\nu = 1$,

$$\pi_0 \left(1 + \frac{1}{1-p} \sum_{\nu=1}^{\infty} \left(\frac{p}{1-p} \right)^\nu \right) = 1.$$

Because $\frac{p}{1-p} < 1$ the geometric series converges and

$$\pi_0 = \frac{1}{2} \cdot \frac{1 - 2p}{1 - p}$$

$$\pi_n = \left(\frac{1}{2} - p\right) \left(\frac{p^{n-1}}{(1-p)^{n+1}}\right)^n, n \geq 1.$$

Random Walks on Weighted Graphs. A *random walk on a weighted graph* is a general and common example of a reversible Markov chain. Assume the undirected edge between vertices i and j in a connected graph has a weight $w_{ij} = w_{ji}$. The Markov chain is irreducible because the graph with edge weights greater than 0 is connected. When at i, the walker goes to j with probability proportional to w_{ij}, so

$$P_{ij} = \frac{w_{ij}}{\sum_\nu w_{i\nu}}.$$

Let

$$s = \sum_{i,j} w_{ij}$$

be the sum of all weights and let

$$\pi_i = \frac{\sum_\nu w_{i\nu}}{s}.$$

The transition probability matrix is reversible because

$$\pi_i P_{ij} = \frac{\sum_\nu w_{i\nu}}{s} \cdot \frac{w_{ij}}{\sum_\nu w_{i\nu}} = \frac{w_{ij}}{s} = \frac{w_{ji}}{s} = \frac{\sum_\nu w_{j\nu}}{s} \cdot \frac{w_{ji}}{\sum_\nu w_{j\nu}} = \pi_j P_{ji}.$$

It is not necessary to forbid self-edges, some edges w_{ii} may be nonzero. However, w_{ii} appears only once in the sum s, while the value w_{ij} appears twice, once each for i and j.

In the simple case with no self-edges and all nonzero weights equal to 1, the invariant distribution is

$$\pi_i = \frac{\text{degree}(i)}{2 \cdot (\text{number of edges})}.$$

Ehrenfest Urn Model. The physicist P. Ehrenfest proposed the following model for statistical mechanics and kinetic theory. The motivation is diffusion through a membrane. Two urns labeled A and B contain a total of N balls. In the Ehrenfest urn model a ball is selected at random with all selections equally likely, and moved from the urn it is in to the other urn. The state at each time is the number of balls in the urn A, from 0 to N. Then the transition probability matrix is

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ \frac{1}{N} & 0 & 1 - \frac{1}{N} & 0 & \cdots & 0 & 0 \\ 0 & \frac{2}{N} & 0 & 1 - \frac{2}{N} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & \frac{1}{N} \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}.$$

The balls fluctuate between the two containers with a drift from the one with the larger number of balls to the one with the smaller numbers.

A stationary distribution for this Markov chain has entry $\pi_i = \binom{N}{i}/2^N$. This is the binomial distribution on N, so that for large N, this can be approximated with the normal distribution. This conclusion is plausible given the physical origin of the Markov chain as a model for diffusion. All states are accessible and all states communicate so the chain has a stationary distribution. The chain is periodic with period 2 so it is not regular.

To show the Markov chain is reversible requires

$$\pi_0 P_{01} = \pi_1 P_{10},$$
$$\pi_i P_{i,i+1} = \pi_{i+1} P_{i+1,i},$$
$$\pi_i P_{i,i-1} = \pi_{i-1} P_{i-1,i},$$
$$\pi_N P_{N,N-1}1 = \pi_{N-1} P_{N-1,N}.$$

See the exercises.

11.5. Chapter Ending Answer

It should not be possible to distinguish whether the video is played forward or in reverse simply by viewing the video and observing the movement of the ball between the urns. The distribution of the balls is stationary, or in physical terms, in equilibrium. That is, a ball is as likely to move from urn A to urn B in forward time, or from urn B to urn A in reverse time. Observing the movement of balls between the urns provides no clue about the time direction of the video.

11.6. Chapter Summary

Key Concepts.

(1) A Markov chain with invariant distribution π is reversible if and only if the state space has a probability distribution π such that

$$\pi_i P_{ij} = \pi_j P_{ji}$$

for all i and j and π is the unique stationary distribution.

(2) The condition says that starting from the stationary distribution

$$\mathbb{P}[X_n = i, X_{n+1} = j] = \mathbb{P}[X_n = j, X_{n+1} = i]$$

so the chain transitions from i to j as often as it transitions from j to i.

(3) Assume the Markov chain is irreducible and positive recurrent and probability distribution π satisfies the detailed balance equation

$$\pi_i P_{ij} = \pi_j P_{ji}.$$

Then the distribution π is the stationary distribution of the chain and the chain is reversible.

(4) Let P be a $k \times k$ transition matrix to which an Algorithm of row and column operations is applied, resulting in $P^{(\star)}$. Then P is reversible if and only if $P^{(\star)}$ is symmetric.

Vocabulary.

(1) Assume X is an irreducible and positive recurrent chain, started at its unique invariant distribution π. This means X_0 has distribution π, and all other X_n have distribution π as well. Now suppose that for every n, the sequence of random variables $X_1, X_2, \ldots, X_n$ has the same joint distribution as the time-reversed sequence $X_n, X_{n-1}, \ldots, X_1$, then the chain X is **reversible**.

(2) The condition

$$\pi_i P_{ij} = \pi_j P_{ji}$$

is the **detailed balance equation**.

(3) **Kolmogorov's loop criterion** is

$$P_{j_0 j_1} P_{j_1 j_2} \cdots P_{j_{k-1} j_k} P_{j_k j_0} = P_{j_0 j_k} P_{j_k j_{k-1}} \cdots P_{j_2 j_1} P_{j_1 j_0}$$

for every finite sequence of distinct states $j_0, j_1, \ldots, j_k$.

Notation.

(1) X, X_0, X_n – Markov chain, starting state, general step of the chain.

(2) k – Number of states in the Markov chain.

(3) π, π_i – Stationary distribution and general element of the stationary distribution.

(4) i, j, i_ℓ etc. – arbitrary states of the Markov chain.

(5) P, P_{ij} – Transition probability matrix and general element of transition probability matrix.

(6) $P^{(\star)}$ – the matrix resulting from a row (or column) multiplication operation on P.

11.7. Sources

The definition of reversibility, the Proposition, and the Reversibility Condition Theorem are adapted from Gravner.

Remarks surrounding the Reversibility Condition Theorem are adapted from Chen and Sigman.

Identifying reversibility with the Kolmogorov loop condition is adapted from Brill et al. [**BhCHJ18**]. The proof that the Kolmogorov Loop Criterion is equivalent to reversibility is adapted from Kolmogorovs Criterion

The example of the negative drift random walk on the non-negative integers is adapted from Sigman.

The example of the random walk on a connected graph is adapted from Gravner.

The problem about the random walk of a king on a checkerboard is inspired by a problem in Gravner.

11.8. Reading Suggestion:

11.9. Outside Readings and Links:

(1)

(2)

(3)

(4)

11.10. Algorithms and Scripts

Data: Transition probability matrix
Result: Maximum Likelihood Estimator of reversed chain

Algorithm.

1 *Initialization of Matrices*

2 Initialize state names and transition probability matrix P; Create new Markov chain object; Run Markov chain for a long time and record sample path; Reverse the sample path; Create Maximum Likelihood Estimator of transition probability matrix;

Algorithm 13: Illustration of Reversible Markov Chains.

Scripts.

R: R script for reversiblechain.R.

```
library("markovchain")

weatherStates <- c("Rain", "Nice", "Snow")
byRow <- TRUE
weatherMatrix <- matrix(
   data=c(1/2, 1/4, 1/4, 1/2, 0, 1/2, 1/4, 1/4, 1/2),
   byrow = byRow, nrow = 3,
   dimnames = list(weatherStates, weatherStates))
mcWeather <- new("markovchain",
   states = weatherStates, byrow = byRow,
   transitionMatrix = weatherMatrix, name="Weather")
weatherDays <- rmarkovchain(n = 2000, object = mcWeather, t0 = "Nice")

reverseWeather <- rev(weatherDays)

weatherFittedMLE <- markovchainFit(data = reverseWeather, method = "mle")
weatherFittedMLE$estimate
```

.

11.11. Problems to Work for Understanding

Exercises.

1: Use the detailed balance equations to show the weather in Oz Markov chain is reversible.
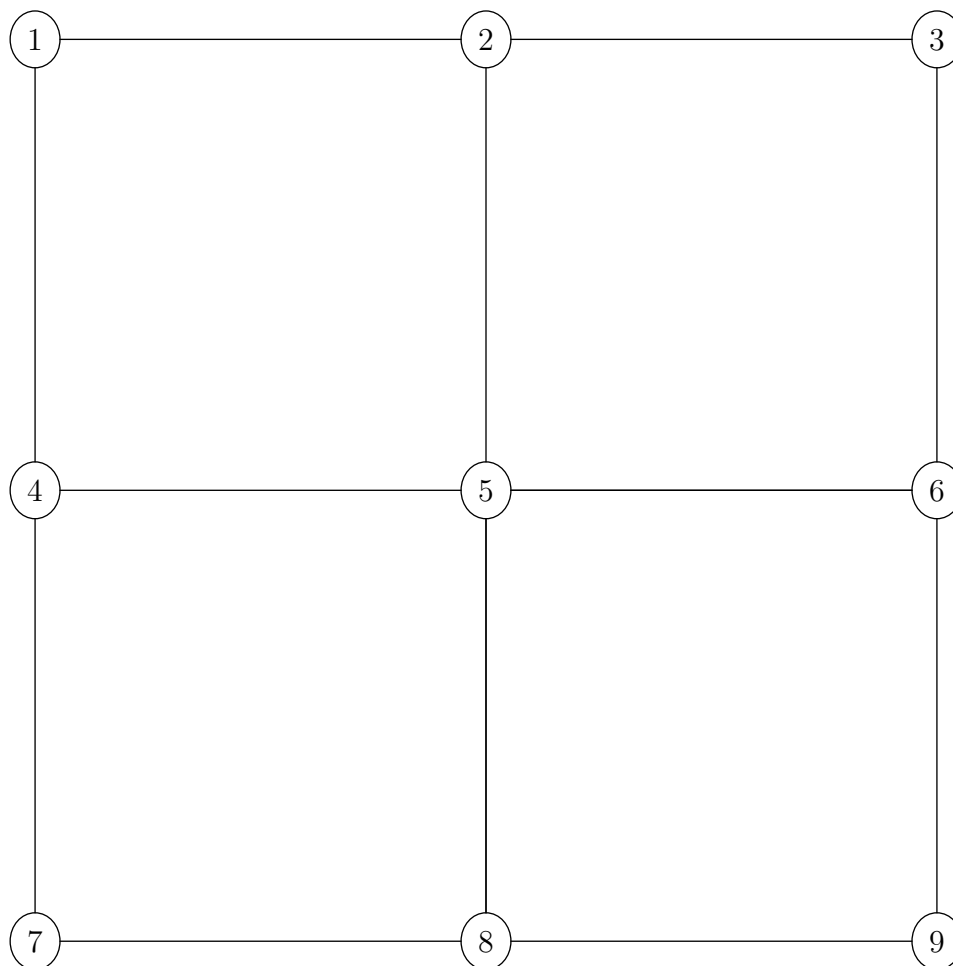
Figure 1. A 3×3 square lattice graph with uniform transition probabilities.

2: Compute the stationary distribution for

$$P = \begin{pmatrix} 17/40 & 0 & 3/40 & 1/2 \\ 0 & 11/20 & 1/4 & 1/5 \\ 3/10 & 1/4 & 9/20 & 0 \\ 1/2 & 1/20 & 0 & 9/20 \end{pmatrix}$$

using the detailed balance equations.

3: Consider the 3×3 square lattice graph in Figure 1. The graph has 9 vertices with 12 edges between nearest lattice neighbors and self-loops at each vertex. If a vertex has n neighbors then the probability of moving to a neighboring vertex or staying at the vertex is $\frac{1}{n+1}$ uniformly. Find the stationary distribution for the random walk on this graph.

4: In the game of Checkers, a "king" can move from a black square to any adjacent black square on an 8×8 square array colored alternately red and black.

(a) Assuming the king starts at one of the two corner squares of the chessboard, compute the expected number of steps before it returns to the starting position.

(b) Now assume two kings start at opposite corner squares, and move independently (and may occupy the same square). What is now the expected number of steps before they simultaneously occupy their starting positions again?

5:   Show that if the Markov chain has a state accesible to and from every other state in exactly 1 step (i.e. the j_0 row and j_0 column of the transition matrix have nonzero entries), then for the Kolmogorov loop criterion it is sufficient to check loops of only three states. (However, it is possible that no such state exists.)

Lumped Markov Chains

Partitioning the states of a Markov chain into equivalence classes or lumps and viewing the dynamics at a coarser level as the system of interest moves among the lumps is often convenient. Examples of such aggregation include a weather model lumping the states "rain" and "snow" into one state called "bad weather" or a physical model aggregating the microstates of a physical system into coarse grained "mesostates" representing many microstates. Other examples are reducing the size of the PageRank matrix used in web searches, and modeling clusters or "communities" in networks.

Although the dynamics moving among the lumps is not necessarily Markovian in general, there is a natural requirement for a Markov chain model on the set of lumped states. This requirement matches the time evolution of the original unlumped chain started at equilibrium. This provides bounds on the error of the dynamics predicted by the lumped chain considered as a model of the unlumped chain. The goal in this chapter is to analyze the accuracy of such coarse grained models compared to the exact microscopic behavior, i.e. to bound the error in a coarse grained description.

Mathematically Mature: may contain mathematics beyond calculus with proofs.

12.1. Chapter Starter Question

The weather in the land of Oz, where it is either rainy, nice or snowy, has probability transition matrix

$$
P = \begin{array}{c} \\ R \\ N \\ S \end{array} \begin{array}{ccc} R & N & S \\ \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \end{array}.
$$

Consider lumping the weather states into only "good" days with nice weather and "bad" days with rainy or snowy weather. What is the probability of going from a bad weather day to a bad weather day?

12.2. Lumpable Markov Chains

Definition 12.1. Let X_n be a Markov chain with state space $\mathcal{X} = \{x_1, x_2, \ldots, x_k\}$ and initial distribution X_0. Given a partition $\hat{math calS} = \{E_1, E_2, \ldots, E_v\}$ of the state space $\mathcal{X}$ into v equivalence classes, define a new stochastic process $\hat{X}_n$ as follows: At the jth step, the state of the new chain is the set E_ξ when E_ξ contains the state of the jth step of the original chain. Think of the notation $\hat{X}_n$ as "heaping up" the chain into the lumps. Assign the transition probabilities for $\hat{X}_n$ as follows: The initial distribution is

$$\mathbb{P}\left[\hat{X}_0 = E_\xi\right] = \mathbb{P}_{X_0}\left[x_0 \in E_\xi\right].$$

Given the initial state, the transition probability for the first step is

$$\mathbb{P}\left[\hat{X}_1 = E_\eta \mid \hat{X}_0 = E_\xi\right] = \mathbb{P}_{X_0}\left[X_1 \in E_\eta\right].$$

In general for the nth step

$$\mathbb{P}\left[\hat{X}_{n+1} = E_\eta \mid \hat{X}_n = E_{\eta_n}, \hat{X}_{n-1} = E_{\eta_{n-1}}, \ldots, \hat{X}_1 = E_{\eta_1}, \hat{X}_0 = E_\xi\right]$$
$$= \mathbb{P}_{X_0}\left[x_{n+1} \in E_j \mid x_n \in E_{t_n}, x_{n-1} \in E_{t_{n-1}}, \ldots, x_1 \in E_{t_1}, x_0 \in E_i\right].$$

Call this new process $\hat{X}_n$ a lumped process and if the lumped process is a Markov chain, it is a **lumped chain**. of the Markov chain X_n. Sometimes this is also called a *projection* of the Markov chain X_n. $\Diamond$

Figure 1 has a schematic diagram of a lumped Markov chain. A lumped process of a given Markov chain need not be a Markov chain in general, see the second example below.

Definition 12.2. A Markov chain with X_n with state space $\mathcal{X}$ is said to be **lumpable** with respect to a partition $\hat{math calS}$ of $\mathcal{X}$ if for every starting distribution X_0 the lumped chain $\hat{X}_n$ is a Markov chain with state space $\hat{\mathcal{S}}$ and the associated transition probabilities do not depend on the choice of X_0. $\Diamond$

The following theorem gives a necessary and sufficient condition for a Markov chain to be lumpable.

Theorem 12.3. *Denote the transition probability of the Markov chain X_n from state x_i to state x_j, for $i, j = 1, \ldots, k$, by P_{ij}. The Markov chain X_n is lumpable with respect to the partition $\hat{math calS}$ if and only if for every pair of sets E_ξ and E_η, $\sum_{x_\nu \in E_\eta} P_{i\nu}$ has the same value for every $x_i \in E_\xi$. These common values form the transition probabilities $\hat{P}_{\xi\eta}$ for the lumped chain.*

Proof. $(\Rightarrow)$ (a) If the chain is lumpable, then

$$\mathbb{P}\left[\hat{X}_1 = E_\eta \mid \hat{X}_0 = E_\xi\right] = \mathbb{P}_{X_0}\left[X_1 \in E_\eta\right]$$

is the same for every X_0 for which it is defined. Call this common value $\hat{P}_{\xi,\eta}$.
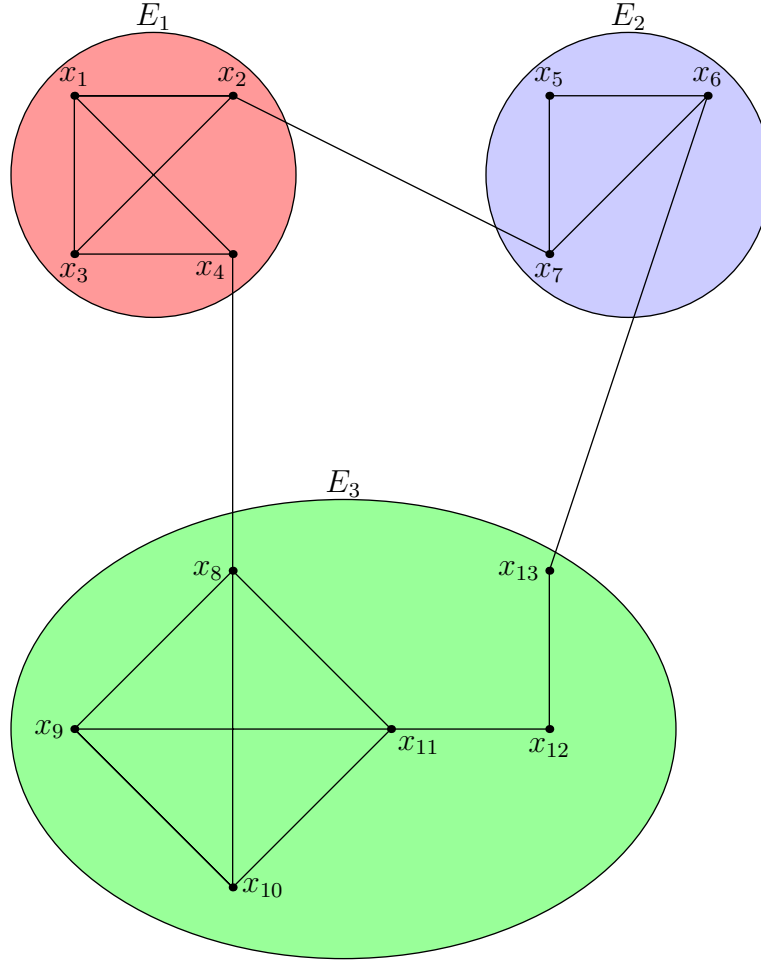
Figure 1. A schematic diagram of a lumped Markov chain.

(b) In particular this must be true for $X_0^{(\nu)}$ having a 1 in its νth component for state $x_\nu \in E_\xi$. Hence $\mathbb{P}_{X_0^{(\eta)}}[x_j \in E_j] = \sum_{x_\nu \in E_\eta} P_{i\nu}$ has the same value $\hat{P}_{ij}$ for every $x_i \in E_\xi$.

($\Leftarrow$) (a) The proof must show that if the lumpability condition is satisfied, the probability

$$\mathbb{P}\left[\hat{X}_{n+1} = E_\eta \mid \hat{X}_n = E_{\eta_n}, \hat{X}_{n-1} = E_{\eta_{n-1}}, \ldots, x_1 \in E_{\eta_1}, \hat{X}_0 = E_\xi\right]$$

depends only on $E_{\eta_{n+1}}$ and E_{η_n}.

(b) By definition, the probability in question is

$$\mathbb{P}_{X_0}\left[x_{n+1} \in E_{\eta_{n+1}} \mid x_n \in E_{\eta_n}, x_{n-1} \in E_{\eta_{n-1}}, \ldots, x_1 \in E_{\eta_1}\right].$$

(c) By the Markov memory-less property, this conditional probability may be written in the form $\mathbb{P}_{\xi'} \left[x_{n+1} \in E_{\eta_{n+1}} \right]$ where ξ' is a probability vector with nonzero components only in the states of E_{η_n}.
(d) This probability depends on the states in $E_{\eta_{n+1}}$ and on the states in E_{η_n}.
(e) By the hypothesis of the lumpability condition, $\mathbb{P}_{\xi'} \left[x_{n+1} \in E_{\eta_{n+1}} \right]$ where ξ' is a probability vector with nonzero components only in the states of E_{η_n} is $\hat{P}_{\eta_n \eta_{n+1}}$, independent of state $x_n \in E_{\eta_n}$.
(f) Thus the probability depends only on $E_{\eta_{n+1}}$ and E_{η_n}.

$\square$

Example 12.4. Recall the example of the weather in the land of Oz, where it is rainy, nice or snowy, with probability transition matrix

$$
P = \begin{array}{c} \\ R \\ N \\ S \end{array} \begin{pmatrix} R & N & S \\ 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/2 \end{pmatrix}.
$$

Now consider lumping the weather states into only "good" days with nice weather and "bad" days with rainy or snowy weather. That is, choose the partition $mat\hat{h}calS = \{\{N\}, \{R, S\}\} = \{G, B\}$. The probabilities of moving from R to N and from S to N are the same. The complementary probabilities of moving from R or S to $\{R, S\}$ are each the same. The probability of moving from N to $\{R, S\}$ or $\{N\}$ is trivially the same for all states in G. This weather Markov chain satisfies the lumpability condition. The lumped transition probability matrix is

$$
P = \begin{array}{c} \\ G \\ B \end{array} \begin{pmatrix} G & B \\ 0 & 1 \\ 1/4 & 3/4 \end{pmatrix}.
$$

By direct computation the stationary distribution on the lumped chain is $(1/5, 4/5)$, as expected from lumping the stationary distribution of the original weather chain.

On the other hand, the lumpability condition is not satisfied for the partition $\{\{R\}, \{N, S\}\}$ since $P_{N\{N,S\}} = 1/2$ but $P_{S\{N,S\}} = 3/4$. $\triangledown$

Example 12.5. An arbitrary lumping of states of a Markov chain does not necessarily lead to a Markov chain. Let X_n be a Markov chain with state space $\mathcal{X} = \{1, 2, 3\}$ with $X_0 = 2$ (or as a probability distribution, $X_0 = (0, 1, 0)$) and transition probability matrix

$$
P = \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.7 & 0.1 \\ 0 & 0.1 & 0.9 \end{pmatrix}.
$$

Then lump the states as $E_1 = \{1\}$ and $E_2 = \{2, 3\}$. First,

$$
\mathbb{P} \left[\hat{X}_1 = E_2 \mid \hat{X}_0 \right] = \mathbb{P}_{X_0} [X_1 = 2, 3] = 0.7 + 0.1 = 0.8
$$

and

$$\mathbb{P}\left[\hat{X}_2 = E_2, \hat{X}_1 = E_2 \mid \hat{X}_0\right] =$$
$$\mathbb{P}_{X_0}\left[X_2 \geq 2, X_1 = 2\right] + \mathbb{P}_{X_0}\left[X_2 \geq 2, X_1 = 3\right] =$$
$$(0.7 + 0.1) \cdot 0.7 + (0.9 + 0.1) \cdot 0.1 = 0.66.$$

Second,

$$\mathbb{P}\left[\hat{X}_2 = E_2, \hat{X}_1 = E_1\right] = \mathbb{P}_{X_0}\left[X_2 \geq 2, X_1 = 1\right] = 0.2 \cdot (0.1 + 0.1) = 0.04.$$

Combining probabilities for these two paths

$$\mathbb{P}\left[\hat{X}_2 = E_2\right] = \mathbb{P}\left[\hat{X}_2 = E_2, \hat{X}_1 = E_2\right] + \mathbb{P}\left[\hat{X}_2 = E_2, \hat{X}_1 = E_1\right] = 0.7.$$

Next consider the three step paths

$$\mathbb{P}\left[\hat{X}_3 = E_2, \hat{X}_2 = E_2, \hat{X}_1 = E_1\right] =$$
$$\mathbb{P}_{X_0}\left[X_1 = 1, X_2 = 2, X_3 \geq 2\right] + \mathbb{P}\left[X_1 = 1, X_2 = 3, X_3 \geq 2\right] =$$
$$0.2 \cdot 0.1 \cdot (0.7 + 0.1) + 0.2 \cdot 0.1 \cdot (0.1 + 0.9) = 0.036.$$

and

$$\mathbb{P}\left[\hat{X}_3 = E_2, \hat{X}_2 = E_2, \hat{X}_1 = E_2\right] =$$
$$\mathbb{P}_{X_0}\left[X_3 \geq 2, X_2 = 2, X_1 = 2\right] + \mathbb{P}_{X_0}\left[X_3 \geq 2, X_2 = 3, X_1 = 2\right] +$$
$$\mathbb{P}_{X_0}\left[X_3 \geq 2, X_2 = 2, X_1 = 3\right] + \mathbb{P}_{X_0}\left[X_3 \geq 2, X_2 = 3, X_1 = 3\right] =$$
$$0.7 \cdot 0.7 \cdot (0.7 + 0.1) + 0.7 \cdot 0.1 \cdot (0.1 + 0.9) +$$
$$0.1 \cdot 0.1 \cdot (0.7 + 0.1) + 0.1 \cdot 0.9 \cdot (0.1 + 0.9) = 0.56.$$

Combining the previous two probabilities

$$\mathbb{P}\left[\hat{X}_3 = E_2, \hat{X}_2 = E_2\right] = 0.56 + 0.036 = 0.596$$

so

$$\mathbb{P}\left[\hat{X}_3 = E_2 \mid \hat{X}_2 = E_2\right] = 0.596/0.7 \approx 0.8514$$

but

$$\mathbb{P}\left[\hat{X}_3 = E_2 \mid \hat{X}_2 = 2, \hat{X}_1 = E_2\right] = 0.56/0.66 \approx 0.8485$$

showing that $\hat{X}_k$ is not a Markov chain. $\triangledown$

A matrix formulation of this theorem for irreducible and aperiodic chains is useful. Define a partition $mat\hat{h}calS = \{E_1, E_2, \ldots, E_v\}$ of the finite state space $\mathcal{X} = \{x_1, x_2, \ldots, x_k\}$ of the underlying Markov chain X_n which is assumed to be irreducible and aperiodic. Then the Markov chain has a unique stationary distribution π. The **distributing matrix** U is the wide $v \times k$ matrix with entries

$$U_{ij} = \begin{cases} \pi_j / \sum_{x_\nu \in E_i} \pi_\nu & x_j \in E_i \\ 0 & \text{otherwise.} \end{cases}$$

The ith row of the distributing matrix is the stationary distribution restricted to E_i and renormalized so its entries add to 1.

Let the **collecting matrix** V be the tall $k \times v$ matrix where the jth column, $j = 1, 2, \ldots, v$, is a vector with 1 in the components corresponding to states in E_j, and 0 elsewhere. For any probability distribution p on $\mathcal{X}$ the collecting matrix specifies the lumped probability distribution $\hat{p} = pV$ on the partition $math\hat{c}alS$. Then $UV = I_v$ since V simply collects back exactly what U distributes. The lumped transition matrix is $\hat{P} = UPV$.

For any probability distribution q in the lumped space, the image qU is locally equilibrated, i.e. qU restricted to any element E_j of the partition equals the stationary distribution restricted to E_j and renormalized to add to q_j. In particular, with $\hat{\pi} = \pi V$

$$U_{ij} = V_{ji}\frac{\pi_j}{\hat{\pi}_i}, \quad i = 1, \ldots, k, j = 1, \ldots, v$$

whose ith row, $i = 1, 2, \ldots, v$, is the probability vector having equal values for states in E_ξ and 0 elsewhere. See the exercises.

Note that the rows of PV corresponding to the elements in the same set of the partition are the same. See the following example with the weather Markov chain. This will be true in general for a chain satisfying the condition for lumpability. The matrix U simply removes this duplication of rows. Again, see the following example with the weather Markov chain. The choice of U is not unique, all that is needed is that the ith row should be a probability vector with nonzero components only for states in E_i. (See the exercises for an example with an absorbing chain without a stationary distribution π.) Sometimes for simplicity and convenience, choose the vector with equal components.

Next consider the local equilibration operator VU. The name is justified by considering its action. Starting from any distribution p on $\mathcal{X}$, VU collects the probability in each E_j, then redistributes this much probability pV among the lumped states in the partition member E_j as $(pU)_j\hat{\pi}_i$. In particular,

$$\pi = \pi VU.$$

Example 12.6. For the land of Oz weather example with

$$\begin{pmatrix} 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/2 \end{pmatrix},$$

recall the stationary distribution is $\pi = (2/5, 1/5, 2/5)$. With the partition $math\hat{c}alS = \{\{N\}, \{R, S\}\} = \{G, B\}$ the distributing matrix is

$$U = \begin{pmatrix} 0 & 1 & 0 \\ 1/2 & 0 & 1/2 \end{pmatrix}.$$

For the probability distribution $q = (2/3, 1/3)$ on the lumped space, $qU = (1/6, 2/3, 1/6)$, so that the $1/3$ probability of being in the bad weather state is distributed proportionally according to the stationary distribution over the R and S states. This explains why U is called the distributing matrix.

The collecting matrix is

$$V = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

For the probability distribution $p = (1/6, 1/12, 3/4)$ on $\{R, N, S\}$, the lumped probability distribution $\{G, B\}$ is $\hat{p} = pV = (1/12, 11/12)$. In particular,

$$\left(\frac{1}{5}, \frac{4}{5}\right) = \hat{\pi} = \pi V = \left(\frac{2}{5}, \frac{1}{5}, \frac{2}{5}\right) \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

The equality $UV = I_2$ is easy to check.

The lumped weather transition probability matrix is

$$\hat{P} = UPV = \begin{pmatrix} 0 & 1 & 0 \\ 1/2 & 0 & 1/2 \end{pmatrix} \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1/4 & 3/4 \end{pmatrix}.$$

For the probability distribution $q = (5/6, 1/6)$ on the lumped weather, the image $qU = (1/12, 5/6, 1/12)$. Restricting qU to element B of the partition gives $(1/12, 1/12)$, the uniform distribution $(2/5, 2/5)$ of the stationary distribution on B renormalized to total $1/6$.

As an example of

$$U_{ij} = V_{ji}\frac{\pi_j}{\hat{\pi}_i}, \quad i = 1, \ldots, v, j = 1, \ldots, k$$

consider $U_{21} = 1/2$ while $V_{12}\frac{\pi_1}{\hat{\pi}_2} = 1 \cdot \frac{2/5}{4/5} = 1/2$.

The rows of

$$PV = \begin{pmatrix} 1/4 & 3/4 \\ 0 & 1 \\ 1/4 & 3/4 \end{pmatrix}$$

corresponding to the R and S states in B are the same. Then multiplying by U removes the duplication and illustrates why the rows of U only need to be a probability vector with nonzero components only for states in G or B. ∇

It is convenient to assume that the state are ordered so that those in E_1 come first, those in E_2 come next and so on with those in E_v last. From here on, assume that this ordering of states has been made.

Theorem 12.7. *If P is the transition probability matrix of the Markov chain X_n, then X_n is lumpable with respect to the partition $\hat{mathcalS}$ if and only if $VUPV = PV$.*

Proof.   (1) ($\Rightarrow$)

(a) The matrix VU has the block matrix form

$$VU = \begin{pmatrix} W_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & W_v \end{pmatrix}$$

where $W_1, W_2, \ldots, W_v$ are probability matrices.

(b) Since the chain is lumpable, the probability of moving from a state of E_i to the set E_j is the same for all states of E_i, hence the components of a column of PV corresponding to E_j are all the same.

(c) Say that common value of probability is ρ_j, so a column of PV is ρ_j in the entries corresponding to E_j and 0 elsewhere.

(d) Then multiplying by VU, with block probability matrix form, gives a column which is ρ_j in the entries corresponding to E_j and 0 elsewhere.

(e) Therefore the columns form a fixed vector for W_j, this proves $VUPV = PV$.

(2) ($\Leftarrow$)

(a) Assume $VUPV = PV$. Then the columns of PV are fixed vectors for VU.

(b) But each block W_j is the transition probability matrix, so its only fixed column vectors are of the form $c\mathbf{1}$.

(c) Hence all the components of a column of PV corresponding to a set E_i must be the same, so the chain is lumpable.

$\square$

Example 12.8. For the weather example with the ordering of states as N, R, S to correspond to $\{G, B\} = \{\{N\}, \{R, S\}\}$,

$$VU = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 1/2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 1/2 & 1/2 \end{pmatrix}.$$

Also

$$PV = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/4 & 1/2 & 1/4 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1/4 & 3/4 \\ 1/4 & 3/4 \end{pmatrix}$$

and it is quick to check that $VUPV = PV$. $\triangledown$

Corollary 12.9.

$$\hat{P}^n = UP^nV.$$

Proof. From the theorem,

$$\hat{P}^2 = UPVUPV = UP^2V$$

and the corollary follows by induction. $\square$

Assume the irreducible and aperiodic chain X with limiting matrix π is lumpable for some partition $\hat{math}calS$. The resulting lumped chain will be also be irreducible and aperiodic. Let $\hat{\pi}$ be the limiting matrix for the lumped chain. Then

$$\begin{aligned} \hat{\pi} &= \lim_{\nu \to \infty} \frac{\hat{P} + \hat{P}^2 + \cdots + \hat{P}^\nu}{\nu} \\ &= \lim_{\nu \to \infty} \frac{UPV + UP^2V + \cdots + UP^\nu V}{\nu} \\ &= U \left(\lim_{\nu \to \infty} \frac{P + P^2 + \cdots + P^\nu}{\nu} \right) V \\ &= U\pi V. \end{aligned}$$

This states that the components of $\hat{\pi}$ are obtained from π by simply adding the components of a given set. From the infinite series representation for the fundamental matrix $\hat{N}$

$$\hat{N} = UNV.$$

12.3. Lumping Absorbing Chains

Assume P is an absorbing chain. Restrict attention to the case of lumping only states of the same kind. That is, any subset of the partition will contain only absorbing states or only nonabsorbing states. Recall that the standard form for an absorbing chain is

$$P = \begin{pmatrix} I & 0 \\ R & Q \end{pmatrix}$$

and write U in the form

$$U = \begin{pmatrix} U_1 & 0 \\ 0 & U_2 \end{pmatrix}$$

where entries of U_1 refer to absorbing states and entries of U_2 to nonabsorbing states. Similarly write V in the form

$$V = \begin{pmatrix} V_1 & 0 \\ 0 & V_2 \end{pmatrix}.$$

In terms of this decomposition, the matrix condition for lumpability becomes

$$V_1 U_1 V_1 = V_1$$
$$V_2 U_2 R V_1 = R V_1$$
$$V_2 U_2 Q V_2 = Q V_2.$$

Since $U_1 V_1 = I$, the first condition is automatically satisfied. The standard form for the transition matrix $\hat{P}$ is

$$\hat{P} = UPV = \begin{pmatrix} U_1 & 0 \\ 0 & U_2 \end{pmatrix} \begin{pmatrix} I & 0 \\ R & Q \end{pmatrix} \begin{pmatrix} V_1 & 0 \\ 0 & V_2 \end{pmatrix} = \begin{pmatrix} I & 0 \\ U_2 R V_1 & U_2 Q V_2 \end{pmatrix}.$$

Then

$$\hat{R} = U_2 R V_1$$
$$\hat{Q} = U_2 Q V_2.$$

From $V_2 U_2 Q V_2 = Q V_2$, obtain $\hat{Q}^2 = U_2 Q V_2 U_2 Q V_2 = U_2 Q^2 V_2$ and so on inductively.

From the infinite series representation for the fundamental matrix N

$$\hat{N} = I + \hat{Q} + \hat{Q}^2 + \cdots$$
$$= U_2 I V_2 + U_2 Q V_2 + U_2 Q^2 V_2 + \cdots$$
$$= U_2 (I + Q + Q^2 + \cdots) V_2$$
$$= U_2 N V_2.$$

From this $\hat{\mathbf{w}} = \hat{N}\mathbf{1} = U_2 N V_2 \mathbf{1} = U_2 N \mathbf{1} = U_2 \mathbf{w}$ and

$$
\begin{aligned}
\hat{B} = \hat{N}\hat{R} &= U_2 N V_2 U_2 R V_1 \\
&= U_2 N R V_1 \\
&= U_2 B V_1.
\end{aligned}
$$

So the important absorption quantities are all easily obtained for the lumped chain from the original chain.

A consequence of $\hat{\mathbf{w}} = U_2 \mathbf{w}$ is the following. Let E_j be any nonabsorbing set, and let x_i be a state in E_j. Choose the ith row of U_2 to be a probability vector with 1 in the x_i component. This means for all x_ℓ in E_j, $\mathbf{w}_\ell = \mathbf{w}_i$. When a chain is lumpable, the mean time to absorption is the same for all starting states in the equivalence set E_j

12.4. Larger Examples

Example 12.10. Consider a random walk of a particle moving along a straight line in unit steps. Each step is 1 unit to the right with probability $\frac{1}{2}$ and to the left with probability $\frac{1}{2}$. The particle moves until it reaches one of two extreme points which are absorbing boundaries. In particular, consider the small case where the state space has 5 values with absorbing boundary states x_1 and x_5 and using the ordering for the standard form the probability transition matrix is

$$
P = \begin{array}{c} \\ s_1 \\ s_5 \\ s_2 \\ s_3 \\ s_4 \end{array}
\begin{array}{c}
\begin{array}{ccccc} s_1 & s_5 & s_2 & s_3 & s_4 \end{array} \\
\left(\begin{array}{ccccc}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
1/2 & 0 & 0 & 1/2 & 0 \\
0 & 0 & 1/2 & 0 & 1/2 \\
0 & 1/2 & 0 & 1/2 & 0
\end{array} \right).
\end{array}
$$

Take the partition $S = \{\{s_1, s_5\}, \{s_2, s_4\}, \{s_3\}\}$. For this partition the condition for lumpability is satisfied, see the exercises. Notice that this would not have been the case for unequal probabilities of moving to left or right.

It is easy to verify that (see the exercises)

$$
\hat{P} = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1 & 0 \end{pmatrix},
$$

$$
\hat{N} = \begin{pmatrix} 2 & 1 \\ 2 & 2 \end{pmatrix},
$$

$$
\hat{\mathbf{w}} = (3, 4)^T
$$

$$
\hat{B} = (1, 1)^T.
$$

∇

Example 12.11. Let Q^k be the k-dimensional hypercube graph, with vertices or node set

$$
V(Q^k) = \{x_0, x_1, \ldots x_{2^k}\} = \{0, 1\}^k
$$

and edge set

$$E(Q^k) = \{(x, y) : x, y \text{ differ in one coordinate}\}.$$

A lazy random walk on this graph is the sequence $X_0, X_1, \ldots, X_{n-1}, X_n, \ldots$ where, given X_n, first choose to remain at X_n with probability $\frac{1}{2}$. Alternatively, with probability $\frac{1}{2}$ choose to move to X_{n+1} selected uniformly at random from the nodes adjacent to x_n. A practical way to implement this random walk is to choose a coordinate $j \in \{1, 2, \ldots, k\}$ uniformly at random and replace the bit in that position with the value of a fair Bernoulli random value. For example, on the 3-dimensional cube a walk at 011 will move to 111 with probability $\frac{1}{2}$ or stay at 011 with probability $\frac{1}{2}$ if coordinate 1 is selected, likewise to 001 or stay at 011 if coordinate 2 is selected, and 010 or stay at 011 if coordinate 3 is selected. As $n \to \infty$, the distribution of X_n converges to the uniform distribution $\pi = \frac{1}{2^k}\mathbf{1}$ on $V(Q^k)$. See the exercises. The aperiodic lazy random walk avoids the annoying issue of the usual random walk being periodic with period 2.

Next consider two urns labeled A and B which contain a total of k balls. In the alternate Ehrenfest urn model a ball is selected at random with all selections equally likely, and moved from the urn it is in to the other urn. The state at each time is the number of balls in the urn A, from 0 to n. Then the transition probability matrix is

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ \frac{1}{2k} & \frac{1}{2} & \frac{1}{2} - \frac{1}{2k} & 0 & \cdots & 0 & 0 \\ 0 & \frac{2}{2k} & \frac{1}{2} & \frac{1}{2} - \frac{2}{k} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & \frac{1}{2k} \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}.$$

The balls fluctuate between the two containers with generally a drift from the one with the larger number of balls to the one with the smaller numbers.

The stationary distribution for this Markov chain has entry $\pi_i = \binom{k}{i}/2^k$. (See the exercises.) This is the binomial distribution on k with $p = 1/2$. For large k, the normal distribution approximates the stationary distribution.

The alternate Ehrenfest urn model is a lumping of the lazy random walk on the k-dimensional hypercube. A set E_ν in the partition, that is, the equivalence class, is defined as the set of vertices with the sum of the coordinates equal to $\nu \in \{0, 1, \ldots, k\}$. The position of the lazy random walk on the hypercube specifies the set of balls in urn A; changing a bit corresponds to moving that particular ball into or out of the urn. See the exercises to prove this using the lumping condition theorem.

$\triangledown$

The covariance matrix of the number of visits to a transient state of the lumped process follows easily from the covariance matrix of the original process. The elements of the covariance matrix $\text{Cov}\left[Y_{E_i E_j}, Y_{E_i E_l}\right]$ for the lumped process are

$$\text{Cov}\left[Y_{E_i E_j}, Y_{E_i E_l}\right] = \sum_{x_\mu \in E_j, x_\nu \in E_l} \text{Cov}\left[Y_{i\mu}, Y_{i\nu}\right]$$

12.5. Bounding Lumping Errors

Starting from a distribution x_0 on $\mathcal{X}$, the time evolution of the distribution among the lumped states E_j is $\hat{x}_t = x_0 P^t V$, while the time evolution of the distribution of the lumped Markov chain is $y_t = x_0 V \hat{P}^t$. How different are these two dynamics?

For the following definition, π may be any properly sized vector, but later it will be chosen to be the stationary distribution vector. Define the norm $\|\cdot\|_\pi$ by

$$\|v\|_\pi = \|v D_{\sqrt{\pi}}\|_2 = \sqrt{\sum_\nu v_\nu^2 / \pi_\nu}$$

where $D_{\sqrt{\pi}} = \mathrm{diag}(1/\sqrt{\pi_1}, \ldots, 1/\sqrt{\pi_k})$ and $\|\cdot\|_2$ is the standard Euclidean 2-norm. The corresponding operator norm is

$$\|A\|_\pi = \max_{\|x\|_\pi=1} \|xA\|_\pi = \max_{\|x D_{\sqrt{\pi}}\|_2=1} \|x D_{\sqrt{\pi}} D_{\sqrt{\pi}}^{-1} A D_{\sqrt{\pi}}\|_2 =$$
$$\max_{\|w\|_2=1} \|w D_{\sqrt{\pi}}^{-1} A D_{\sqrt{\pi}}\|_2 = \|D_{\sqrt{\pi}}^{-1} A D_{\sqrt{\pi}}\|_2.$$

Now using the v-dimensional stationary distribution vector $\hat{\pi}$ on the lumped chain, the one step difference is

$$\|x_0 V \hat{P} - x_0 P V\|_{\hat{\pi}} = \|x_0 V U P V - x_0 P V\|_{\hat{\pi}} = \|x_0 (V U P - P) V\|_{\hat{\pi}}.$$

The n step difference is

$$\|x_0 V \hat{P}^n - x_0 P^n V\|_{\hat{\pi}} = \|x_0 V (U P V)^n - x_0 P^n V\|_{\hat{\pi}}$$
$$= \|x_0 V\, U P V\, U P V \cdots U P V\, U P V - x_0 P^n V\|_{\hat{\pi}} =$$
$$\|x_0 (V U P)^n V - x_0 P^n V\|_{\hat{\pi}} = \|x_0 ((V U P)^n - P^n) V\|_{\hat{\pi}}.$$

The important operator for bounding the difference is VUP. For convenience, let $VUP = H$.

The goal is to bound the n step difference $\|H^n - P^n\|_\pi$ in terms of the 1-step difference $\|H - P\|_\pi$ now using the right sized π. The general behavior of the difference is that it can at most grow for a while, but must eventually decline to 0 since by construction the lumped and unlumped chain converge to the same equilibrium distribution. Bounding how large the difference can get in terms of the 1 step difference is the next theorem.

Theorem 12.12. *Assume P is the transition probability matrix for a regular and reversible Markov chain. Let $\delta = \|VUP - P\|_\pi$. Then*

$$\|(VUP)^n - P^n\|_\pi \le K(n)\delta \le \hat{K}\delta$$

with $K(n) = n\,|\lambda_2|^{n-1}$, where λ_2 is the second largest eigenvalue of P and $\hat{K} = -1/(\lambda_2 \cdot \mathrm{e} \cdot \log(\lambda_2))$.

Remark 12.13. For further understanding, the matrices and projections used in the following proof are illustrated in the small case of the lumped weather example after the proof.

Proof. (1) First note that

$$\|H^n - P^n\|_\pi = \|(H - P)H^{n-1} + P(H^{n-1} - P^{n-1})\|_\pi.$$

(2) Iterating,

$$\|H^n - P^n\|_\pi = \|\sum_{\nu=0}^{n-1} P^\nu (H - P) H^{n-\nu-1}\|_\pi$$

$$\leq \sum_{\nu=0}^{n-1} \|P^\nu (H - P) H^{n-\nu-1}\|_\pi.$$

(3) Recall that H and P have common stationary distribution π. Define the projection $P_\pi = \mathbf{1}\pi$ where $\mathbf{1}$ is an $k \times 1$ column vector and π is a $1 \times k$ row vector. P_π projects a column vector into the 1-dimensional subspace spanned by $\mathbf{1}$.

(4) The complementary projection is $P_\Sigma = I - P_\pi$.

(5) The decomposition into the subspaces gives the following representations of P and H:

$$P = (P_\pi + P_\Sigma)P = P_\pi + P_\Sigma P$$
$$H = (P_\pi + P_\Sigma)H = P_\pi + P_\Sigma H$$

because $P_\pi P = P_\pi$ and $P_\pi H = P_\pi$ since the rows of P_π are the stationary vector π.

(6) Then $H - P = P_\Sigma H - P_\Sigma P$ and $0 = P_\pi P_\Sigma P = P_\Sigma P P_\pi = P_\pi P_\Sigma H = P_\Sigma H P_\pi$.

(7) Recall $\delta = \|VUP - P\|_\pi$ and $VUP = H$, so $\|VUP - P\|_\pi = \|H - P\|_\pi = \|P_\Sigma H - P_\Sigma P\|_\pi = \delta$.

(8) For integers ν, n

$$\|P^\nu (H - P) H^{n-\nu-1}\|_\pi$$
$$= \|(P_\pi + P_\Sigma P)^\nu (P_\Sigma H - P_\Sigma P)(P_\pi + P_\Sigma H)^{n-\nu-1}\|_\pi$$
$$= \|(P_\Sigma P)^\nu (P_\Sigma H - P_\Sigma P)(P_\Sigma H)^{n-\nu-1}\|_\pi$$
$$= \|P_\Sigma P\|_\pi^\nu \cdot \delta \cdot \|P_\Sigma H\|_\pi^{n-\nu-1}.$$

(9) Use the special choice of norm, in particular, use the fact that for a matrix $\|A\|_\pi = \|D_{\sqrt{\pi}}^{-1} A D_{\sqrt{\pi}}\|_2$. By the assumption of reversibility, $D_{\sqrt{\pi}}^{-1} P D_{\sqrt{\pi}}$ is a symmetric matrix. Therefore, the 2-norm is the dominant eigenvalue of the matrix. In fact, as shown below the diagonal matrix $D_{\sqrt{\pi}}$ also symmetrizes P_π (see step 10), $P_\pi P$, (see step 11), $P_\Sigma P$ (see step 12), and VU (see steps 13 and 14). Therefore each has norm equaling the dominant eigenvalue.

(10) Direct calculation shows $\pi D_{\sqrt{\pi}} = \mathbf{1}^T D_{\sqrt{\pi}}^{-1}$. Starting with the definition of P_π

$$D_{\sqrt{\pi}}^{-1} P_\pi D_{\sqrt{\pi}} = D_{\sqrt{\pi}}^{-1} \mathbf{1}\pi D_{\sqrt{\pi}} = (\mathbf{1}^T D_{\sqrt{\pi}}^{-1})^T (\pi D_{\sqrt{\pi}}) =$$
$$(\pi D_{\sqrt{\pi}})^T (\pi D_{\sqrt{\pi}}) = ((\pi D_{\sqrt{\pi}})^T (\pi D_{\sqrt{\pi}}))^T = ((\mathbf{1}^T D_{\sqrt{\pi}}^{-1})^T (\pi D_{\sqrt{\pi}}))^T$$
$$= ((D_{\sqrt{\pi}}^{-1} \mathbf{1})(\pi D_{\sqrt{\pi}}))^T = (D_{\sqrt{\pi}}^{-1} (\mathbf{1}\pi) D_{\sqrt{\pi}})^T = (D_{\sqrt{\pi}}^{-1} P_\pi D_{\sqrt{\pi}})^T.$$

This shows P_π is symmetrized by $D_{\sqrt{\pi}}$.

(11) So $D_{\sqrt{\pi}}^{-1}PD_{\sqrt{\pi}}$ and $D_{\sqrt{\pi}}^{-1}P_\pi D_{\sqrt{\pi}}$ are symmetric and commute with each other since P and P_π commute. Thus,

$$D_{\sqrt{\pi}}^{-1}P_\pi PD_{\sqrt{\pi}} = D_{\sqrt{\pi}}^{-1}P_\pi D_{\sqrt{\pi}}D_{\sqrt{\pi}}^{-1}PD_{\sqrt{\pi}}$$

is symmetric.

(12) Then $D_{\sqrt{\pi}}^{-1}P_\Sigma PD_{\sqrt{\pi}} = D_{\sqrt{\pi}}^{-1}(I - P_\pi)PD_{\sqrt{\pi}}$ is symmetric as well.

(13) From $U_{ij} = V_{ji}\frac{\pi_j}{\sqrt{\hat{\pi}_i}}$, it follows that $U = D_{\sqrt{\hat{\pi}}^2}V^T D_{\sqrt{\pi}}^{-2}$ or equivalently $D_{\sqrt{\pi}}^{-1}UD_{\sqrt{\pi}} = D_{\sqrt{\hat{\pi}}}V^T D_{\sqrt{\pi}}^{-1}$.

(14) Thus,

$$\begin{aligned}
D_{\sqrt{\pi}}^{-1}VUD_{\sqrt{\pi}} &= (D_{\sqrt{\pi}}^{-1}VD_{\sqrt{\hat{\pi}}})(D_{\sqrt{\hat{\pi}}}^{-1}UD_{\sqrt{\pi}}) \\
&= (D_{\sqrt{\pi}}^{-1}VD_{\sqrt{\hat{\pi}}})(D_{\sqrt{\hat{\pi}}}^{-1}V^T D_{\sqrt{\pi}}^{-1}) \\
&= (D_{\sqrt{\pi}}^{-1}VD_{\sqrt{\hat{\pi}}})(D_{\sqrt{\pi}}^{-1}VD_{\sqrt{\hat{\pi}}})^T.
\end{aligned}$$

As the product of a matrix and its transpose, VU is symmetrized by $D_{\sqrt{\pi}}$.

(15) Since $D_{\sqrt{\pi}}^{-1}P_\Sigma PD_{\sqrt{\pi}}$ is symmetric and using the definition of the matrix norm, $\|P_\Sigma P\|_\pi = |\lambda_2|$, where λ_2 is the second-largest eigenvalue of P. To bound $\|P_\pi H\|_\pi$:

$$\begin{aligned}
\|P_\Sigma H\|_\pi = \|VUP - P_\pi\|_\pi &= \|VUP - VUP_\pi\|_\pi \\
&= \|VU(P - P_\pi)\|_\pi \le \|P_\Sigma P\|_\pi \|VU\|_\pi = |\lambda_2|\,\|VU\|_\pi.
\end{aligned}$$

(16) Since VU is a stochastic matrix and $D_{\sqrt{\pi}}$ symmetrizes VU, it follows that $\|VU\|_\pi = 1$.

(17) Thus the bound becomes $\|H^n - P^n\|_\pi \le n\,|\lambda_2|^{n-1}\,\delta = K(n)\delta$ where $K(n) = n\,|\lambda_2|^{n-1}$.

(18) Maximizing $K(n)$ over n gives $K(n) \le \hat{K} = \frac{-1}{\lambda_2\cdot e\cdot\log(\lambda_2)}$.

$\square$

Corollary 12.14.

$$\|x_0 V\hat{P}^n - x_0 P^n V\|_{\hat{\pi}} \le K_2 \cdot n\,|\lambda_2|^{n-1}\cdot\delta$$

where $\delta = \|VUP - P\|_\pi$, λ_2 is the second largest eigenvalue of P and K_2 is a constant related to x_0 and V.

Proof. See the exercises. $\square$

Example 12.15. For the weather example, the stationary distribution for P is $(2/5, 1/5, 2/5)$ so the vector norm is

$$\|v\| = \sqrt{\frac{5v_1^2}{2} + 5v_2^2 + \frac{5v_3^2}{2}}$$

and the diagonalizing matrix is

$$D_{\sqrt{\pi}} = \begin{pmatrix} 1/\sqrt{2/5} & 0 & 0 \\ 0 & 1/\sqrt{1/5} & 0 \\ 0 & 0 & 1/\sqrt{2/5} \end{pmatrix}.$$

The transition probability matrix and the collecting and distributing matrices are

$$P = \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/2 \end{pmatrix}, \quad V = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 0 & 1 & 0 \\ 1/2 & 0 & 1/2 \end{pmatrix}$$

so

$$H = VUP = \begin{pmatrix} 3/8 & 1/4 & 3/8 \\ 1/2 & 0 & 1/2 \\ 3/8 & 1/4 & 3/8 \end{pmatrix}.$$

The eigenvalues of P are 1, 1/4, and $-1/4$. The stationary distribution for H is also $(2/5, 1/5, 2/5)$ and the eigenvalues of H are 1, 0, and $-1/4$.

The projections are

$$P_\pi = \mathbf{1}\pi = \begin{pmatrix} 2/5 & 1/5 & 2/5 \\ 2/5 & 1/5 & 2/5 \\ 2/5 & 1/5 & 2/5 \end{pmatrix} \quad P_\Sigma = I - P_\pi \begin{pmatrix} 3/5 & -1/5 & -2/5 \\ -2/5 & 4/5 & -2/5 \\ -2/5 & -1/5 & 3/5 \end{pmatrix}.$$

Although it is obvious by construction, directly verifying

$$P = (P_\pi + P_\Sigma)P = P_\pi + P_\Sigma P$$
$$H = (P_\pi + P_\Sigma)H = P_\pi + P_\Sigma H$$

and $H - P = P_\Sigma H - P_\Sigma P$ and $0 = P_\pi P_\Sigma P = P_\Sigma P P_\pi = P_\pi P_\Sigma H = P_\Sigma H P_\pi$ is easy. Although it is obvious by construction, it is also easy to verify that $D_{\sqrt{\pi}}^{-1} P D_{\sqrt{\pi}}$ is symmetric directly by computation. The diagonal matrix $D_{\sqrt{\pi}}$ also symmetrizes P_π, $P_\pi P$, $P_\Sigma P$, and VU.

Figure **??** illustrates the numerical norm difference of the lumped chain with red dots from lumping the original chain with the blue points. The upper bound from the eigenvalue is illustrated with the red points, showing that the inequality is satisfied.

$$\nabla$$

12.6. Chapter Ending Answer

The probability of going from R to either R or S is $\frac{1}{4} + \frac{1}{2} = \frac{3}{4}$. The probability of going from S to either R or S is $\frac{1}{2} + \frac{1}{4} = \frac{3}{4}$. Any probability combination of being in the states R and S leads to a probability $\frac{3}{4}$ of being in the states R or S, so the probability of going from bad weather to bad weather is $\frac{3}{4}$.

12.7. Chapter Summary

Key Concepts.

(1) Denote the transition probability of the Markov chain X_n from state x_i to state x_j, for $i, j = 1, \ldots, k$, by P_{ij}. A necessary and sufficient condition for the Markov chain X_n to be lumpable with respect to the partition S is that for every pair of sets E_ξ and E_η, $\sum_{x_\nu \in E_\eta} P_{i,\nu}$ has the same value for every $x_i \in E_\xi$. These common values form the transition probabilities $P_{\xi,\eta}$ for the lumped chain.
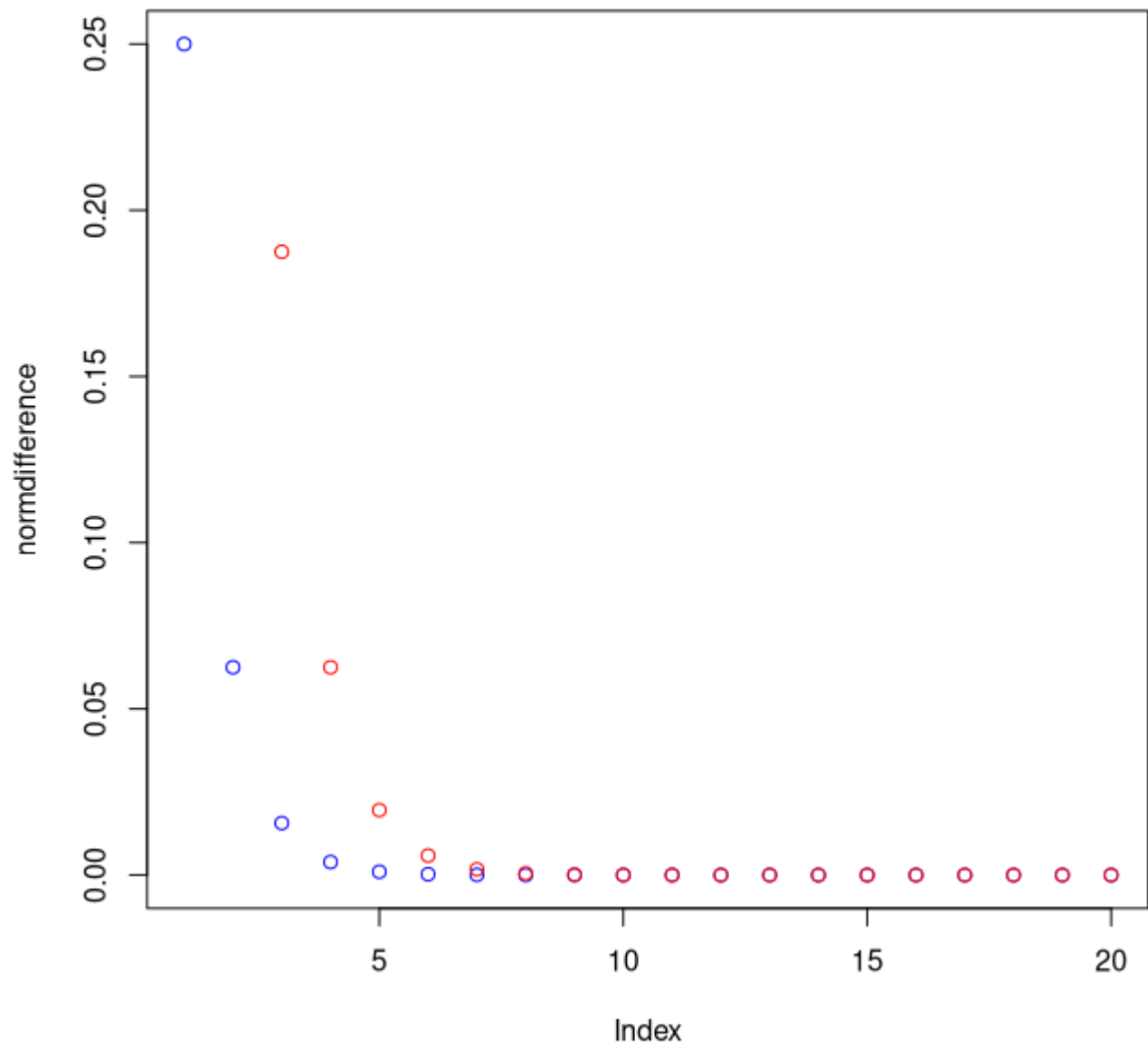
Figure 2. The numerical norm difference of the lumped chain from lumping the original chain in blue with the theoretical upper bound in red.

(2) The **distributing matrix** U is the wide $v \times k$ matrix with entries

$$U_{ij} = \begin{cases} \pi_j / \sum_{\nu \in E_j} \pi_\nu & x_j \in E_j \\ 0 & \text{otherwise.} \end{cases}$$

The rows of the distributing matrix are the stationary distribution restricted to E_j and renormalized so its entries add to 1. Let the **collecting matrix** V be the tall $k \times v$ matrix with the jth column, $j = 1, 2, \ldots, v$, is a vector with 1s in the components corresponding to states in E_j, and 0 elsewhere. The collecting matrix specifies the lumped probability distribution $\hat{P} = PV$ on the partition E.

(3) If P is the transition probability matrix of the Markov chain X_n, then X_n is lumpable with respect to the partition $\mathcal{S}$, if and only if $VUPV = PV$.

(4) Let $\delta = \|VUP - P\|_\pi$. Then

$$\|(VUP)^n - P^n\|_\pi < K(n)\delta < K\delta$$

with $K(n) = n |\lambda_2|^{n-1}$, where λ_2 is the second largest eigenvalue of P and $\hat{K} = -1/(\lambda_2 \cdot \mathrm{e} \cdot \log(\lambda_2))$.

Vocabulary.

(1) Let X_n be a Markov chain with state space $\mathcal{X} = \{x_1, x_2, \ldots, x_k\}$ and initial distribution X_0. Given a partition $math\hat{c}alS = \{E_1, E_2, \ldots, E_v\}$ of the state space $\mathcal{S}$, define a new stochastic process $\hat{X}_n$ as follows: At the jth step, the state of the new chain is the set E_ℓ when E_ℓ contains the state of the jth step of the original chain. Assign the transition probabilities for $\hat{X}_n$ as follows: The initial distribution is

$$\mathbb{P}\left[\hat{X}_0 = E_\xi\right] = \mathbb{P}_{X_0}\left[x_0 \in E_\xi\right].$$

Given the initial state, the transition probability for step 1 is

$$\mathbb{P}\left[\hat{X}_1 = E_\eta \mid \hat{X}_0 = E_\xi\right] = \mathbb{P}_{X_0}\left[X_1 \in E_\eta\right].$$

In general for the nth step

$$\mathbb{P}\left[\hat{X}_{n+1} = E_\eta \mid \hat{X}_n = E_{\eta_n}, \hat{X}_{n-1} = E_{\eta_{n-1}}, \ldots, \hat{X}_1 = E_{\eta_1}, \hat{X}_0 = E_\xi\right]$$
$$= \mathbb{P}_{X_0}\left[X_{n+1} \in E_\eta \mid X_n \in E_{\eta_n}, X_{n-1} \in E_{\eta_{n-1}}, \ldots, X_1 \in E_{\eta_1}, X_0 \in E_\xi\right].$$

Call this new stochastic process, $\hat{X}_n$, a **lumped chain** of the Markov chain X_n. Sometimes this is also called a *projection* of the Markov chain X_n.

(2) A Markov chain with X_n with state space $\mathcal{X}$ is said to be **lumpable** with respect to a partition $math\hat{c}alS$ of $\mathcal{X}$ if for every starting distribution X_0 the lumped process $\hat{X}_n$ is a Markov chain with state space $\hat{\mathcal{S}}$ and the associated transition probabilities do not depend on the choice of X_0.

(3) The **distributing matrix** U is the $v \times k$ matrix with entries

$$U_{ij} = \begin{cases} \pi_j / \sum_{\nu \in E_i} \pi_\nu & x_j \in E_i \\ 0 & \text{otherwise.} \end{cases}$$

The ith row of the distributing matrix is the stationary distribution restricted to E_i and renormalized so its entries add to 1.

(4) Let the **collecting matrix** V be the $k \times v$ matrix with the jth column, $j = 1, 2, \ldots, v$, is a vector with 1 in the components corresponding to states in E_j, and 0 elsewhere. The collecting matrix specifies the lumped probability distribution $\hat{p} = pV$ on the partition E.

Notation.

(1) X_n – discrete time, discrete space Markov chain

(2) $\mathcal{X} = \{x_1, x_2, \ldots, x_k\}$ – state space

(3) k – number of states in the Markov chain

(4) x_i, x_j – generic states of Markov chain

(5) P – $k \times k$ transition probability matrix

(6) p_{ij} – entry in the transition matrix

(7) X_0 – initial distribution or state of the Markov chain, with abuse of notation for the same notation to represent both an initial probability distribution over the states and the single state corresponding to a delta-distribution (certainty) on that state

(8) – π – stationary probability distribution

(9) $math\hat{c}alS = \{E_1, E_2, \ldots, E_v\}$ – a partition of the state space $\mathcal{X}$ into equivalence classes

(10) v – number of states in the lumped chain

(11) E_ξ, E_η – generic state of the lumped chain with E_ξ the source and E_η the destination.

(12) $\hat{X}_n$ – a lumped process

(13) $\hat{\pi}$ – lumped chain stationary distribution

(14) U and V – distributing matrix and collecting matrix respectively

(15) p, q – probability distributions on $\mathcal{X}$

(16) $\hat{p}$ – corresponding probability distribution on $mathcalS$

(17) I_v – $v \times v$ identity matrix

(18) W_i – submatrices of VU

(19) I, R, Q – submatrices of absorbing chain transition probability matrix P

(20) U_1, U_2, V_1, V_2 – submatrices of U, V

(21) $\hat{N}, \hat{\mathbf{w}}, \hat{B}$ – normal matrix and absorption quantities for lumped chain

(22) $Q^k, E(Q^k), V(Q^k)$ – k-dimensional hypercube as a graph, with the edge and vertex sets of the graph

(23) $\|v\|_\pi$ – vector norm related to π

(24) $D_{\sqrt{\pi}}$ – diagonal matrix related to π

(25) $H = VUP$ – matrix used for bounding norm difference

(26) P_π, P_Σ – projection matrices

(27) Σ – subspace complementary to span of π

(28) δ – initial norm difference

(29) λ_2 – second largest eigenvalue

(30) $K(n), \hat{K}$ – upper bounds on norm difference

(31) ν – A dummy variable of summation, no meaning, meant to mimic the counting variable n

12.8. Sources

The example of reducing the PageRank matrix is from Barreto and Fragoso.

The example of lumped chains from clustering or "communities" and the diagram are adapted from Piccardi.

The example of a lumped chain which is not Markov is adapted from class notes from Eric Slud, Statistics 650.

The accuracy bounds on the lumped chain are adapted from [**HS09**]

The example of lumping a random walk on the hypercube to the Ehrenfest urn model is based on Sections 2.3 of [**LPW09**].

12.9. Reading Suggestion:

12.10. Outside Readings and Links:

(1)

(2)

(3)

(4)

12.11. Algorithms and Scripts

Algorithm.

Data: Transition probability matrix, collecting, distributing, diagonal matrices

Result: Plot of norm difference between lumped and original chain

1 *Initialization of Matrices*

2 Initialize P, V, U, D, $Dinv$

3 Initialize λ_2

4 Initialize number of iterations N

5 $H \leftarrow VUP$

6 $H_n \leftarrow H$

7 $P_n \leftarrow P$

8 $normDifference \leftarrow [\|Dinv(H_n - P_n)D\|_2]$

9 **for** $n \leftarrow 1$ **to** $N - 1$ **do**

10 $\quad$ $H_n \leftarrow HH_n$

11 $\quad$ $P_n \leftarrow PP_n$

12 $\quad$ Append $\|Dinv(H_n - P_n)D\|_2$ to $normDifference$

13 **end**

14 Generate comparison list $n\lambda_2^{n-1}$ for $n = 1 : N$

15 Plot $normDifference$ and comparison list on same axes

Algorithm 14: Illustration of Theorem 12.12.

Scripts.

R: R script for lumpedchains.R.

```
1  # Enter matrices by columns, the default in R
2  P <- matrix(c(1/2, 1/2, 1/4, 1/4, 0, 1/4, 1/4, 1/2, 1/2), nrow=3, ncol=3)
3  V <- matrix(c(0, 1, 0, 1, 0, 1), nrow=3, ncol = 2)
4  U <- matrix(c(0, 1/2, 1, 0, 0, 1/2), nrow=2, ncol=3)
5  D <- matrix(c(1/sqrt(2/5), 0, 0, 0, 1/sqrt(1/5), 0, 0, 0, 1/sqrt(2/5)),
        nrow=3, ncol=3)
6  Dinv <- matrix(c(sqrt(2/5), 0, 0, 0, sqrt(1/5), 0, 0, 0, sqrt(2/5)), nrow
        =3, ncol=3)
7
8  lambda2 <- 1/4
9
10 N <- 20
11 H <- V %*% U %*% P
12 Hn <- H
13 Pn <- P
14
15 normdifference <- c( norm( Dinv %*% (Hn - Pn) %*% D, type="2") )
16 for (n in 1:(N-1) ) {
17     Hn <- H %*% Hn
18     Pn <- P %*% Pn
19     normdifference <-
20         c( normdifference, norm( Dinv %*% (Hn - Pn) %*% D, type="2") )
21 }
22 plot(normdifference, col="blue")
23
24 K <- function(n, l2) { n * l2^{n-1} }
25 points(K(1:N, lambda2), col="red")
```

.

12.12. Problems to Work for Understanding

1:   For the 5-state absorbing symmetric random walk with transition probability matrix

$$P = \begin{matrix} & \begin{matrix} s_1 & s_5 & s_2 & s_3 & s_4 \end{matrix} \\ \begin{matrix} s_1 \\ s_5 \\ s_2 \\ s_3 \\ s_4 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 1/2 & 1/2 & 0 & 0 \end{pmatrix} \end{matrix}$$

and partition $S = \{\{s_1, s_5\}, \{s_2, s_4\}, \{s_3\}\}$ show that the condition for lumpability is satisfied. Verify that

$$\hat{P} = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\hat{N} = \begin{pmatrix} 2 & 1 \\ 2 & 2 \end{pmatrix}$$

$$\hat{\mathbf{w}} = (3, 4)^T$$

$$\hat{B} = (1, 1)^T.$$

2:   Show that for irreducible and aperiodic chains with $\hat{\pi} = \pi V$

$$U_{ij} = V_{ji} \frac{\pi_j}{\hat{\pi}_i}, \quad i = 1, \ldots, k, j = 1, \ldots, v$$

whose ith row, $i = 1, 2, \ldots, v$, is the probability vector having equal values for states in E_ξ and 0 elsewhere.

3:   For the lazy random walk X_n on the hypercube Q^k show that as $n \to \infty$, the distribution of X_n converges to the uniform distribution π on $V(Q^k)$.

4:   Prove that the stationary distribution for the alternate Ehrenfest urn model has entry $\pi_i = \binom{k}{i}/2^k$.

5:   Show the alternate Ehrenfest urn model is a lumping of the lazy random walk on the n-dimensional hypercube using the correspondence between $\{0, 1\}^n$ and subsets of $\{1, \ldots, n\}$, under which a vertex corresponds to the vector or bit string with 1s in the positions of its elements. The position of the random walk on the hypercube specifies the set of balls in urn A; changing a bit corresponds to moving that numbered ball into or out of the urn.

6:   For the lumping of the lazy random walk on the 3-dimensional cube Q^3 starting at the origin into the alternate Ehrenfest urn model with 3 balls, plot the norm-difference of the distribution of the Ehrenfest urn model from the distribution of the lumped random walk and compare to the theoretical upper bound.

7:

(1) For the weather in Oz example with the probability distribution $p = (1/6, 1/12, 3/4)$ on $\{R, N, S\}$, the lumped probability distribution $\{G, B\}$ is $\hat{p} = pV = (1/12, 11/12)$. Calculate $\|pV\|_{\hat{\pi}}$ and $\|p\|_{\pi}$.

(2) Show in general that for any irreducible and aperiodic Markov chain and any
distribution $\|pV\|_{\hat{\pi}} \leq \|p\|_{\pi}$.

8: Show

$$\|x_0 V \hat{P}^n - x_0 P^n V\|_{\hat{\pi}} \leq K_2 \cdot n \, |\lambda_2|^{n-1} \cdot \delta$$

where $\delta = \|VUP - P\|_{\pi}$, λ_2 is the second largest eigenvalue of P and K_2 is a
constant related to x_0.

Part 2

Advanced Examples and Theory

Motivational Examples for the Metropolis Algorithm

13.1. Rating

The purpose of this chapter is to provide motivational examples of the Metropolis algorithm for creating Markov chains which converge to a given distribution. These examples are from group theory, and statistical mechanics. The examples and results are presented without proofs. These examples range from simple to sophisticated Markov chains.

Mathematicians Only: prolonged scenes of intense rigor.

13.2. Chapter Starter Question

(1) What is the "cycle and transposition" notation for elements of the group of permutations on n elements?

(2) Explain an algorithm for selecting a random permutation from the group of permutations on n elements.

13.3. Symmetric Function Theory

Let $\mathcal{X} = S_n$, the **symmetric group** of n letters, that is, the group of permutations on n letters.

Remark 13.1. This section uses the following notation for specific elements of S_n. For compact but complete representation, a permutation will usually be a row vector or ordered list with square brackets. For example in S_6 the permutation $[3, 4, 5, 6, 1, 2]$ is the function which carries the letter in position 1 to position 3, the letter in position 2 to position 4 and so on to the letter in position 6 carried to position 2. This is an "active permutation" in the sense used by Hankin [**?**] and

takes less space than the matrix format

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 1 & 2 \end{pmatrix}.$$

Hankin refers to this format as a permutation in "word" format. The classic representation of permutations in terms of cycles is also necessary in this section. The example permutation would be written as $(1, 3, 5)(2, 4, 6)$ with parentheses around the cycles. Hankin refers to this as the cycle representation. Transpositions are the short cycles (i, j) and cycles leaving an element fixed are (i).

The group operation is composition of permutations, thinking of them as functions, written with a small circle. The symmetric group element $\sigma \circ \tau$ is the function applying σ, and then applying function τ. For example, the first entry in the word representation of $\sigma \circ \tau$ is $\tau(\sigma(1))$. Postfix notation for the functions would denote the action of the permutation on letter or position x as $x\sigma\tau$.

Let $d(\sigma, \sigma_0)$ be a metric on the symmetric group by

$d(\sigma, \sigma_0) = $ minimum number of transpositions to bring σ to σ_0

$\quad\quad = $ minimum number of transpositions in a representation of $\sigma^{-1} \circ \sigma_0$.

This metric is called **Cayley's distance** because a result of A. Cayley implies that $d(\sigma, \sigma_0) = n - \text{cycles}(\sigma^{-1} \circ \sigma_0)$ where $\text{cycles}(\tau)$ is the number of cycles, including fixed cycles, in $\tau \in S_n$. This metric is bi-invariant: $d(\sigma, \sigma_0) = d(\tau \circ \sigma, \tau circ \sigma_0) = d(\sigma \circ \tau, \sigma_0 \circ \tau)$. Later σ_0 will be a distinguished or central element, usually the identity, so the metric measures distances from this "center".

Example 13.2. In S_6, let $\sigma_0 = [3, 4, 5, 6, 1, 2] = (1, 3, 5)(2, 4, 6)$. Consider $\sigma = [2, 6, 4, 3, 1, 5] = (1, 2, 6, 5)(3, 4)$. The permutation that carries σ to σ_0 is $[1, 3, 6, 5, 2, 4] = (1)(2, 3, 6, 4, 5)$. Writing

$$(2, 3, 6, 4, 5) = (2, 3)(3, 6)(6, 4)(4, 5) = (2, 5)(2, 4)(2, 6)(2, 3),$$

the minimum number of transpositions that brings σ to σ_0 is 4 so $d(\sigma, \sigma_0) = 4$. Furthermore, $\sigma^{-1} = [5, 1, 4, 3, 6, 2]$ so (remembering how permutations are functions) $\sigma^{-1} \circ \sigma_0 = [1, 3, 6, 5, 2, 4] = (1)(2, 3, 6, 4, 5)$ with 2 cycles. This is an example of Cayley's theorem that $d(\sigma, \sigma_0) = n - \text{cycles}(\sigma^{-1} \circ \sigma_0) = 6 - 2 = 4$. $\quad\quad\quad \triangledown$

Example 13.3. As another larger example, consider S_{11} with

$$\sigma_0 = [4, 2, 9, 10, 6, 5, 11, 7, 8, 1, 3] = (1, 4, 10)(3, 9, 8, 7, 11)(5, 6).$$

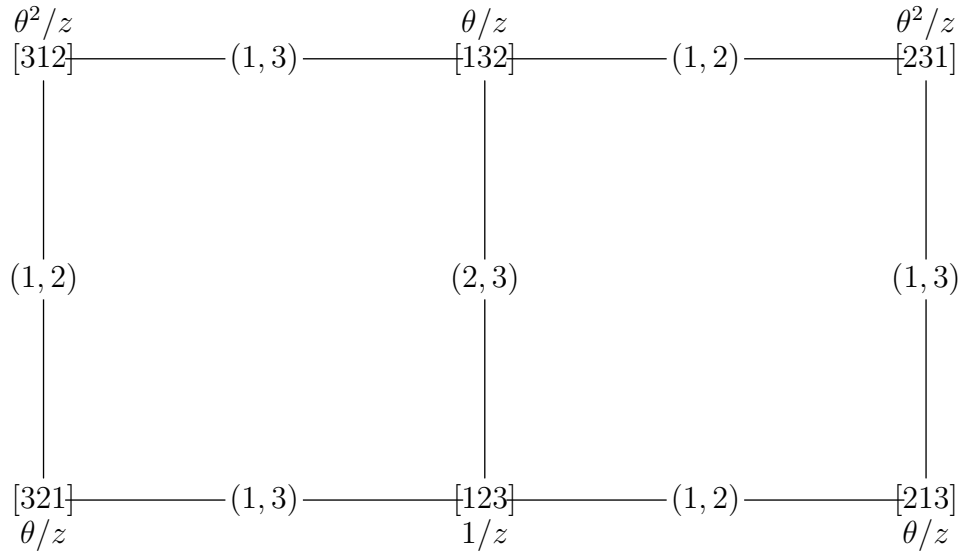Take $\sigma = [7, 4, 5, 3, 1, 2, 11, 9, 8, 6, 10]$. The permutation that carries σ to σ_0 is

$$[6, 5, 10, 2, 9, 1, 4, 8, 7, 3, 11] = (1, 6)(2, 5, 9, 7, 4)(3, 10)(8)(11)$$
$$= (1, 6)(2, 5)(5, 9)(9, 7)(7, 4)(3, 10).$$

The minimum number of transpositions that brings σ to σ_0 is 6 so $d(\sigma, \sigma_0) = 6$. Furthermore,

$$\sigma^{-1} = [5, 6, 4, 2, 3, 10, 1, 9, 8, 11, 7] = (1, 5, 3, 4, 2, 6, 10, 11, 7)(8, 9)$$

so (remembering that permutations are functions)

$$\sigma^{-1} \circ \sigma_0 = [6, 5, 10, 2, 9, 1, 4, 8, 7, 3, 11] = (1, 6)(2, 5, 9, 7, 4)(3, 10)(7)(9)$$

θ^2/z
$[312]$ ——————— $(1,3)$ ——————— $[132]$ ——————— $(1,2)$ ——————— $[231]$
θ/z ... θ^2/z

$(1,2)$ ... $(2,3)$ ... $(1,3)$

$[321]$ ——————— $(1,3)$ ——————— $[123]$ ——————— $(1,2)$ ——————— $[213]$
θ/z ... $1/z$ ... θ/z

Figure 1. The graph of S_3 with transpositions indicated on the edges, giving a diagram of the metric space with the metric being the number of edges between any two permutations and the probability distribution above and below each permutation.

with 5 cycles. This is an example of Cayley's theorem that $d(\sigma, \sigma_0) = n-\text{cycles}(\sigma^{-1}\circ \sigma_0) = 11 - 5 = 6$.
                                                                            ∇

Fix σ_0, usually the identity permutation $\sigma_0 = \text{id}$. Define a probability distribution on S_n by

$$\mathbb{P}\left[(|\,\sigma\right] = z^{-1}\theta^{d(\sigma,\sigma_0)}, \sigma, \sigma_0 \in S_n, 0 < \theta \leq 1.$$

The normalizing constant is known in this example:

$$z = \sum_\sigma \theta^{d(\sigma,\sigma_0)} = \prod_{\nu=1}^{n}(1 + (\nu - 1)\theta).$$

If $\theta = 1$, then $p(\sigma)$ is the uniform distribution on S_n. For $0 < \theta < 1$, $p(\sigma)$ is largest at σ_0 and falls off from its maximum as σ moves away from σ_0. This becomes a nonuniform distribution on S_n, with the distribution having a peak at σ_0. An example with the small permutation group S_3 is illustrated in Figure 1 with the metric being the number of edges between any two permutations. The normalizing constant is $z = 1 + 3\theta + 2\theta^2 = (1)(1 + \theta)(1 + 2\theta)$.

The main problem here is "How can samples be drawn from this distribution $\mathbb{P}\left[\cdot\right]$ defined on the symmetric group?" The answer here is to use a Markov chain defined by the Metropolis algorithm, based on random transpositions. To be specific, moving from σ, choose a transposition (i, j) uniformly at random and consider $(i, j)\sigma = \sigma^*$. If $d(\sigma^*, \sigma_0) \leq d(\sigma, \sigma_0)$ the Markov chain state moves to σ^*. If $d(\sigma^*, \sigma_0) > d(\sigma, \sigma_0)$, flip a coin with probability θ of coming up heads. If the coin comes up heads, move to σ^*, otherwise stay at σ. The Markov chain transition

probabilities are

$$
P(\sigma,\sigma^*) = \begin{cases} 1/\binom{n}{2} & \sigma^* = (i,j)\sigma, & d(\sigma^*,\sigma_0) \le d(\sigma,\sigma_0) \\ \theta/\binom{n}{2} & \sigma^* = (i,j)\sigma, & d(\sigma^*,\sigma_0) > d(\sigma,\sigma_0) \\ (1-m\theta)/\binom{n}{2} & \sigma^* = \sigma, & m = \mathrm{card}\left(\{(i,j) : d((i,j)\sigma,\sigma_0) > d(\sigma,\sigma_0)\}\right) \\ 0 & \text{otherwise} \end{cases}.
$$

This Markov chain usually moves toward σ_0, with occasional steps away from σ_0. This is consistent with the nonuniform distribution on S_n, with the distribution having a peak at σ_0. The Metropolis construction guarantees the stationary distribution π of this Markov chain is the distribution $\mathbb{P}[\cdot]$. Note that this discrete state space and discrete time Markov chain is a specific example of a random walk on the graph in Figure 1.

When $n = 3$, $\mathcal{X} = S_3$ and $\sigma_0 = \mathrm{id} = [123]$ the complete transition probability matrix is (see the exercises)

$$
\begin{array}{c c}
 & \begin{array}{cccccc} [123] & [213] & [321] & [132] & [231] & [312] \end{array} \\
\begin{array}{c} [123] \\ [213] \\ [321] \\ [132] \\ [231] \\ [312] \end{array} &
\left(\begin{array}{cccccc}
1-\theta & \frac{\theta}{3} & \frac{\theta}{3} & \frac{\theta}{3} & 0 & 0 \\
\frac{1}{3} & \frac{2}{3}(1-\theta) & 0 & 0 & \frac{\theta}{3} & \frac{\theta}{3} \\
\frac{1}{3} & 0 & \frac{2}{3}(1-\theta) & 0 & \frac{\theta}{3} & \frac{\theta}{3} \\
\frac{1}{3} & 0 & 0 & \frac{2}{3}(1-\theta) & \frac{\theta}{3} & \frac{\theta}{3} \\
0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\
0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0
\end{array} \right)
\end{array}.
$$

The stationary distribution is the left eigenvector proportional to $(1,\theta,\theta,\theta,\theta^2,\theta^2)$, precisely the distribution $\mathbb{P}[\cdot]$. So sampling from the distribution $\mathbb{P}[\cdot]$ can be done by running the Markov chain long enough and using the resulting distribution of states. The eigenvalues are 1, $\frac{2}{3}(1-\theta), -\theta, 0$ with multiplicities 1, 1, 3 and 1 respectively

A theorem from Diaconis [**DS81, Dia09, DS98**], shows that starting from the identity and using the identity as σ_0, on the order of $n \log n$ steps are necessary and sufficient to make the distance to stationarity small. If the chain starts far from the identity, for example at an n-cycle, it can be shown that order $n^2 \log n$ steps suffice. In the example for S_3 this means between 3 to 10 steps. As a larger example, if $n = 52$, the number of cards in a standard deck, then $n \log n \approx 205.465$, while $n^2 \log n \approx 10684.163$. So the running time for sampling from the stationary distribution should be from about several hundred to several tens of thousands of steps.

13.4. General Setting from Statistical Mechanics

The following is a general setting of Markov chains in statistical mechanics with some illustrative examples. A more specific model is described in detail in the next section.

A state of the multiple particles in the statistical mechanical model is described by a **configuration** x from the state space $\mathcal{X}$, called a **configuration space** in statistical mechanics. The configuration space can be infinite or finite, continuous or discrete. For example, with N interacting particles, each particle's position and

velocity in three-dimensional space specifies a configuration. In this case, $\mathcal{X}$ is an infinite, continuous subset of $\mathbb{R}^{6N}$. As another example, $\mathcal{X}$ might be taken as a bounded subset Λ of the integer lattice in the plane, attaching a value ± 1 to each site in Λ. This value might indicate the presence of particle there, or it might indicate an orientation (or spin) of a particle at the site. If $\text{card}\,(\Lambda) = N$, then the configuration space $\mathcal{X}$ consists of all 2^N possible assignments of values to sites in Λ.

The physics of a configuration space $\mathcal{X}$ is described by an energy function $E : \mathcal{X} \to \mathbb{R}^+$. The energy of a particular configuration is $E(x)$. For the continuous example, the energy could be the sum of potential energies, or the sum of all kinetic energies, or the sum of both. For the discrete example above, the energy could show the total influence that neighboring particles exert on each other. This is the **Ising** model. Note that the configuration space and the number of states is quite large compared with most of the standard examples of Markov chains.

A basic principle of statistical physics is that Nature prefers low-energy configurations with high probability. This principle governs the random organization of molecules in a room. Rarely observed configurations, say all the molecules gathering in a corner of the room have high energies and hence low probabilities. In fact, the probabilities are so low such configurations are in principle never observed. Common configurations, such as all molecules distributed uniformly at all locations throughout the room have low energies and hence higher probabilities, high enough that those configurations are essentially the only configurations ever observed.

For a system at equilibrium, the relative frequency of a configuration x is given by its **Boltzmann weight**

$$\mathrm{e}^{E(x)/\mathrm{k}T}$$

where T is the temperature and k is Boltzmann's constant. For any $x \in \mathcal{X}$, its Boltzmann probability $\text{Boltz}(x)$ is

$$\text{Boltz}(x) = \frac{\mathrm{e}^{E(x)/\mathrm{k}T}}{z}$$

where the denominator

$$z = \sum_{x \in \mathcal{X}} \mathrm{e}^{E(x)/\mathrm{k}T}$$

is called the **partition function**. In any realistic setting, the partition function is analytically and computationally intractable. This intractability accounts for the scarcity of analytic, closed-form results in statistical mechanics.

The total energy of the system is the expected value of the energy function, defined by

$$\langle E \rangle = \sum_{x \in \mathcal{X}} E(x)\, \text{Boltz}(x) = \frac{\sum_{x \in \mathcal{X}} E(x)\mathrm{e}^{E(x)/\mathrm{k}T}}{z}.$$

Other physical properties are defined similarly. In each case, there is no avoiding the partition function z.

Expressions such as the total energy could be naively approximated by using simple sampling. Generate a sample $x_1, x_2, \ldots x_M$ *uniformly* from x and estimate

both the numerator and the denominator separately, giving

$$\langle E \rangle \approx \frac{\sum_{\nu=1}^{M} E(x_\nu) e^{E(x_\nu)/kT}}{\sum_{\nu=1}^{M} e^{E(x_\nu)/kT}}.$$

The limitation of sampling uniformly from the configuration space is not practical since a configuration where $e^{E(x)/kT}$ is very small gets chosen with high probability; that is a configuration with low weight. The key to the Metropolis algorithm is that instead of choosing configurations randomly, then weighting them with $e^{E(x)/kT}$, instead choose configurations with probability $e^{E(x)/kT}$ and weight them uniformly. In other words, it is better to sample from $\mathcal{X}$ so that x is selected with probability Boltz(x). If this can be done, then for any such sample $x_1, x_2, \ldots x_M$

$$\frac{1}{M} \sum_{\nu=1}^{M} E(x_\nu) \to \langle E \rangle$$

as $M \to \infty$ with rate of convergence $O(M^{-1/2})$. The challenge is to sample from the nonuniform distribution.

13.5. Phase Transitions and Hard Disks

The study of phase transitions in statistical mechanics is a classical problem. For many substances, like water, experiments produce phase diagrams such as that shown in the schematic Figure 2. The general picture has a finite length liquid-vapor phase transition line ending in a critical point and a triple point of temperature and pressure where all three forms of matter coexist. A solid-liquid phase line extends to infinity. This general form of the phase transition diagram seems universal for all kinds of matter. This suggests there must be a general explanation, independent of details such as molecular structure. As a model, the physicist John G. Kirkwood posed the problem of whether a gas of hard spheres would show phase transitions.

As a simplified two-dimensional hard spheres model consider placement of n discs of radius ϵ in the unit square. The discs must be non-overlapping and completely contained in the unit square. Typically, n is large on the order of 100 to 10^6 with ϵ correspondingly small. The centers of the discs give a configuration point in the configuration space, $[0,1]^{2n}$. Consider the set of configurations $\mathcal{X}(n, \epsilon)$ for fixed n. One can think of the configuration as something like a "foam" in $[0,1]^{2n}$. For fixed n and ϵ very small this set is connected, one can move from one point in $\mathcal{X}(n, \epsilon)$ to another point by sliding one or more discs around in the square. Some points in $[0,1]^{2n}$ will be inaccessible because the corresponding discs overlap or extend outside the square. By its embedding in $[0,1]^{2n}$, $\mathcal{X}(n, \epsilon)$ inherits a natural uniform distribution, Lebesgue measure restricted to $\mathcal{X}(n, \epsilon)$. The problem is to pick points in $\mathcal{X}(n, \epsilon)$ uniformly. Choosing $X_1, X_2, \ldots X_M$ with uniform distribution for function $f : \mathcal{X}(n, \epsilon) \to \mathbb{R}$ gives the approximation

$$\int_{\mathcal{X}(n,\epsilon)} f(x) \, dx \approx \frac{1}{M} \sum_{\nu=1}^{M} f(X_\nu).$$

This hard disks problem is the original motivation for the Metropolis algorithm. The following is a version of the Metropolis algorithm for hard discs.
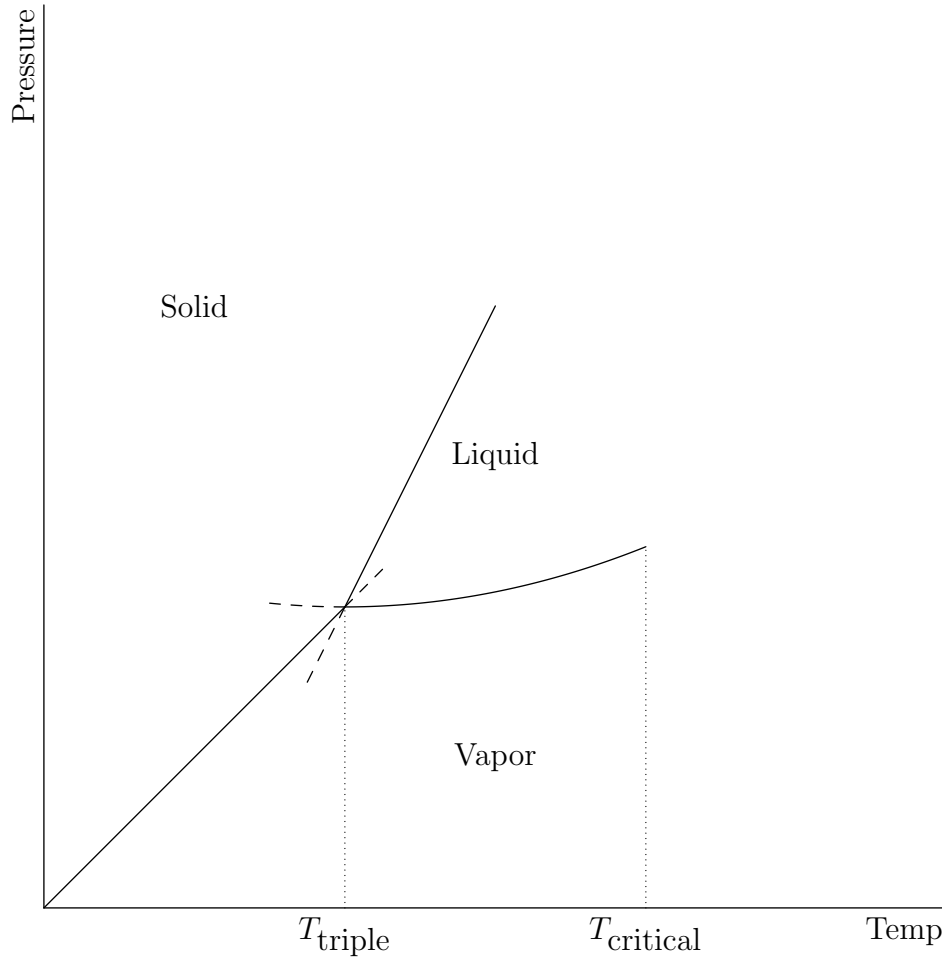
Figure 2. Schematic phase transition diagram.

(1) Start with a configuration $x \in \mathcal{X}(n, \epsilon)$.

(2) Pick a disc in that configuration at random, that is, with probability $1/n$.

(3) Pick a point at random in a disc of radius h, centered at the chosen disc center. At random means with Lebesgue measure restricted to the disc of radius h.

(4) Try to move the chosen disc center to the chosen point, if the resulting configuration is in $\mathcal{X}(n, \epsilon)$ accept the move; else, stay at x.

(5) The algorithm continues, randomly moving coordinates.

The algorithm provides a discrete time continuous space Markov chain. If $X_1, X_2, \ldots X_M$ are successive configurations, then theory and simulations show that X_M becomes uniformly distributed provided ϵ is small and M is large. For large M, the X_i can be used to approximate integrals as above.

This hard disks model has been studied from a number of perspectives. Simulations indicate a phase transition when the density of disks is about 0.71. This empirical value is below the close packing density which is $\frac{\pi\sqrt{3}}{6} \approx 0.9069$. Below the transition density, the disks look random, above the transition density, the disks look close to a lattice packing. The notions of randomness and packing are quantified by a variety of functions. For example,

$$f(x) = \left| \frac{1}{n} \sum_{\nu=1}^{n} \frac{1}{n_\nu} \sum_\ell e^{6i\theta_{\nu,\ell}} \right|$$

where the sum is over the n particles encoded by $x \in [0,1]^{2n}$, the sum in ℓ is over the n_ℓ neighbors of the νth particle and $\theta_{\nu,\ell}$ is the angle between the particles ν and ℓ in some fixed reference frame. If the configuration has a local hexagonal structure, the sum should be small. Different functions can be used to study long-range order.

13.6. A Generalization of Hard Disks

Let $\mathcal{X} \subset \mathbb{R}^d$ be a bounded connected open set. Let $\tilde{p}(x) > 0$, $z = \int_{\mathcal{X}} \tilde{p}(x)\, dx < \infty$, $\mathbb{P}[x] = z^{-1}\tilde{p}(x)$ be a probability distribution on $\mathcal{X}$. As necessary, extend $p(x) = 0$ outside $\bar{\mathcal{X}}$, the topological closure of $\mathcal{X}$.. Sampling problems can be stated as *Given $\tilde{p}$, choose points in $\mathcal{X}$ from $\mathbb{P}[\cdot]$*. Note that the normalizing constant z may not be given and is usually impossible to approximate. As an example, consider placing fifty hard discs of radius $\epsilon = 1/100$ randomly in the unit square. The set of allowable configurations is a complex "cuspy" set. Even though $\tilde{p} = 1$ on $\mathcal{X}$, it is not practical to compute z. Nevertheless, it is still possible to sample from $\mathbb{P}[\cdot]$.

(1) For $x \in \mathcal{X}$, fix a small positive h.
(2) Choose $y \in B_x(h)$, from normalized Lebesgue measure on the ball $B_x(h) \subset \mathbb{R}^d$.
(3) If $\mathbb{P}[y] \geq Probx$, move to y.
(4) If $\mathbb{P}[y] < \mathbb{P}[x]$, move to y with probability $\mathbb{P}[y]/\mathbb{P}[x]$.
(5) If $\mathbb{P}[y] < \mathbb{P}[(|x)$, stay at x with probability $1 - \mathbb{P}[y]/\mathbb{P}[x]$.

Note that this algorithm does not require knowing z because it only relies on the ratio $\mathbb{P}[y]/\mathbb{P}[x] = \tilde{p}(y)/\tilde{p}(x)$. The transition from x to y has a transition kernel (not matrix, since the state space is continuous)

$$P(x,\, dy) = m(x)\delta_x + \frac{h^{-d}}{\mathrm{Vol}(B_1)}\delta_{B_1}\left(\frac{x-y}{h}\right)\min\left(\frac{p(x)}{p(y)}, 1\right)\, dy$$

where

$$m(x) = 1 - \int_{\mathbb{R}^d} \frac{h^{-d}}{\mathrm{Vol}(B_1)}\delta_{B_1}\left(\frac{x-y}{h}\right)\min\left(\frac{p(x)}{p(y)}, 1\right)\, dy.$$

This kernel operates on $L^2(p)$ by

$$P[f](x) = \int_{\mathbb{R}^d} f(y)P(x,\, dx).$$

$P(x,\, dy)$ is a bounded self-adjoint operator on $L^2(p)$. Describe this discrete time continuous space Markov chain as:

(1) Start at $X_0 = x \in \mathcal{X}$.

(2) Pick X_1 from $P(X_0, \mathrm{d}y)$.

(3) Pick X_2 from $P(X_1, \mathrm{d}y)$.

(4) Continue this process.

This means that

$$P(X_2 \in A) = P_x^2(A) = \int_{\mathbb{R}^d} P(z, A) P(x, \mathrm{d}z),$$

and more generally

$$P(X_M \in A) = P_x^M(A) = \int_{\mathbb{R}^d} P(z, A) P^{M-1}(x, \mathrm{d}z).$$

Under the assumptions that for $\mathcal{X}$ connected and h small, then for all $x \in \mathcal{X}$ and $A \subset \mathcal{X}$, the process converges to the stationary distribution,

$$P_x^M(A) \to \int_A p(y) \, \mathrm{d}y \,.$$

A natural question is to ask how fast this convergence occurs. How many steps should the algorithm be run to achieve an acceptable degree of convergence? The following theorem of Diaconis, Lebeau and Michel gives an estimate:

Theorem 13.4 (Diaconis, Lebeau, Michel)**.** *Let $\mathcal{X}$ be a connected Lipshitz domain in $\mathbb{R}^d$. For p measurable (with $0 < m \le p(x) \le M < \infty$ on $\mathcal{X}$) and h fixed and small, the Metropolis algorithm satisfies*

$$\left| P_x^M(A) - \int_A p(y) \, \mathrm{d}y \right| \le c_1 e^{-c_2 M h^2} \ \text{uniformly in } x \in \mathcal{X}, A \subset \mathcal{X}.$$

Here c_1, c_2 are positive constants that depend on $\tilde{p}$ and $\mathcal{X}$ but not on x, M or h. The inequality has a matching lower bound. Good estimates on c_2 are available.

Note that the Metropolis algorithm in this section is based on steps in the full-dimensional ball $B_{\epsilon(x)}$ while the Metropolis algorithm for discs in the previous section is based on changing just two coordinates at a time.

13.7. Chapter Ending Answer

(1) The cycle $(i_1, i_2, \ldots, i_r)$ is the permutation that sends element i_1 to i_2, i_2 to i_3 and so on, until i_r is sent to i_1, leaving all other elements fixed. A transposition is just a 2-cycle (i_1, i_2) exchanging 2 elements. Compose cycles by composing the permutations each represents. A standard result about permutations is that every permutation is the composition of its cycles, and that every permutation is uniquely expressed as the composition of its disjoint cycles. Furthermore, a simple examination shows that the cycle $(i_1, i_2, \ldots, i_r)$ can be written as $(i_1, i_2)(i_1, i_3) \ldots (i_1, i_r)$ so that every cycle can be written as the product of transpositions.

(2) A simple algorithm to generate a permutation of n items uniformly at random without retries, known as the Knuth shuffle or the modern Fisher-Yates algorithm, is to start with any permutation (for example, the identity permutation), and then go down through the positions n to 1. For each position i swap the element currently there with a randomly chosen element from positions 1 through i inclusive. It follows that this algorithm will produce any

permutation of n elements with probability exactly $1/n!$, thus yielding a uniform distribution over all such permutations.

13.8. Chapter Summary

Key Concepts.

(1) Let $X = S_n$, the **symmetric group** of n letters, that is, the set of permutations on n letters with the Cayley distance as metric. The metric defines a probability distribution on S_n by

$$\pi(\sigma) = z^{-1}\theta^{d(\sigma,\sigma_0)}, \sigma, \sigma_0 \in S_n, 0 \le \theta \le 1.$$

Choose a transposition (i,j) uniformly at random and consider $(i,j)\sigma = \sigma^*$. Using the Metropolis algorithm with this transposition defines a Markov chain on the symmetric group. A theorem from Diaconis [**DS98, DS81**] shows that starting from the identity and using the identity as σ_0, on the order of $n \log n$ steps are necessary and sufficient to make the distance to the stationary distribution π small.

(2) A state of the particles in statistical mechanics is described by a **configuration** ω from the **configuration space** Ω. The physics of a configuration space Ω is described by an energy function $E : \Omega \to \mathbb{R}^+$. A basic principle of statistical physics is that Nature seeks low-energy configurations. The total energy of the system is the expected value of the energy function. Defining a probability transition with the Metropolis algorithm, choose configurations with probability $e^{E(\omega)/kT}$ and weight them uniformly to find the expected value.

(3) As a model, the physicist John Kirkwood posed the problem of whether a gas of hard disks would show a phase transitions. Consider placement of n discs of radius ϵ in the unit square. This hard disks problems is the original motivation for the Metropolis algorithm defining a Markov chain in the configuration space.

Vocabulary.

(1) Let $X = S_n$, the **symmetric group** of n letters, that is, the set of permutations on n letters. Let $d(\sigma, \sigma_0)$ be a metric on the symmetric group by

$$d(\sigma, \sigma_0) = \text{ minimum number of transpositions to bring } \sigma \text{ to } \sigma_0 .$$

This metric is called **Cayley's distance**.

(2) In statistical mechanics, a state of the particles is described by a **configuration** x from the **configuration space** $\mathcal{X}$. The configuration can be infinite or finite, continuous or discrete.

(3) For Ω a bounded subset Λ of the integer lattice in the plane, attach a value ± 1 to each site in Λ. This value might indicate the presence of particle there, or it might indicate an orientation (or spin) of a particle at the site. If $\text{card}(\Lambda) = N$, then the configuration space Ω consists of all 2^N possible assignments of values to sites in Λ. The physics of a configuration space Ω is described by an

energy function $E : \Omega \to \mathbb{R}^+$. The energy could show the total influence that neighboring particles exert on each other. This is the **Ising** model.

(4) For a system at equilibrium, the relative frequency of a configuration ω is given by its **Boltzmann weight**

$$\mathrm{e}^{E(\omega)/\mathrm{k}T}$$

where T is the temperature and k is Boltzmann's constant. For any $\omega \in \Omega$, its Boltzmann probability $\mathrm{Boltz}(\omega)$ is

$$\mathrm{Boltz}(\omega) = \frac{\mathrm{e}^{E(\omega)/\mathrm{k}T}}{z}$$

where the denominator

$$z = \sum_{\omega \in \Omega} \mathrm{e}^{E(\omega)/\mathrm{k}T}$$

is called the **partition function**.

Notation.

(1) $\mathcal{X} = S_n$ – the **symmetric group** of n letters, that is, the group of permutations on n letters

(2) σ, σ_0, τ – elements of symmetric group

(3) $d(\sigma, \sigma_0)$ – a metric on the symmetric group

(4) $c(\tau)$ – number of cycles in permutation τ

(5) $p(\sigma)$ – probability distribution on S_n

(6) z – normalizing constant for the probability distribution,

(7) θ – factor defining the probability distribution

(8) x – **configuration** from the state space, called a **configuration space** $\mathcal{X}$ in statistical mechanics.

(9) Λ – the integer lattice in the plane

(10) $E : \mathcal{X} \to \mathbb{R}^+$ – energy function

(11) T – temperature and k is Boltzmann's constant.

(12) $\langle E \rangle$ – expected value of the energy function

(13) M – number of samples for expected energy approximation

(14) ϵ – radius of small disc

(15) $\mathcal{X} \subset \mathbb{R}^d$ – a bounded connected open set

(16) $\tilde{p}(x) > 0$ – unnomralized distribution function

(17) $+ z$ – normalization factor for $\tilde{p}(x)$

(18) δ_x – Dirac delta function

13.9. Sources

This section is adapted from: The Markov Chain Monte Carlo Revolution by Persi Diaconis [**Dia09**]. The Chapter Ending Answer on random permutations is taken from the Wikipedia article Random Permutations

13.10. Reading Suggestion:

13.11. Outside Readings and Links:

(1)

(2)

(3)

(4)

13.12. Algorithms and Scripts

Data: Distribution factor θ

Result: Total variation distance of empirical from theoretical

Algorithm.

1 Initialize "burn-in" threshold and iterations; Initialize center σ_0

2 Initialize vector to hold empirical visit count

3 Initialize σ as *id* **for** $1 : iterations$ **do**

4   Generate a random transposition and apply to σ giving $\sigma^\star$

5   If $\sigma^\star$ is closer to σ_0 than σ Update σ as $\sigma^\star$ **else**

6   Flip a θ-coin

7   **end**

8   **if** *Win* **then**

9   Update σ as $\sigma^\star$

10   **end**

11   **else**

12   Keep current σ

13   **end**

14   After "burn-in" threshold record iteration results in vector corresponding to Cayley distances in S_3.

15 **end**

16 Report Total variation distance of empirical from theoretical

Scripts.

R: library("permutations")

```
1  n <- 3 # Same as text example
2  theta <- 1/2
3  threshold <- round(3 * n * log(n)) # reasonable guess, based on theory
4  iterations <- threshold + 100 * factorial(n)
5  sigma0 <- id
6
7  visited <- numeric( factorial(n)  )
8
9  numcycles <- function( x ) { # number of cycles, including length 1 cycles
10     if (is.id( x ) ) {
11         return( n )
12     } else {
13         numrealcycles <- length( get1( x ) ) # get element from each cycle
14         numfixed <- sum( fixed( x ) )
15         return(numrealcycles + numfixed)
16     }
17 }
18
```

```
19  cayleydist <- function(u, v) { # Cayley distance of perm u to perm  v
20      ## u, v should be same symmetric group.
21      max( size(u), size(v) ) - numcycles( inverse(u) * v )
22      ## size(id) = 0, so if u = id, use size(v) instead
23  }
24
25  sigma <- id
26  for (i in 1:iterations) {
27      sigmastar <- rcyc(1, 2, n) * sigma # move by random transposition
28      if (cayleydist( sigmastar, sigma0) <= cayleydist(sigma, sigma0) ) {
29          sigma <- sigmastar # if closer, update sigma
30      } else {
31          if ( runif(1) <= theta ) { # toss theta-coin
32              sigma <- sigmastar     # update if win toss
33          }
34      }
35
36      if ( i > threshold) {
37          if ( sigma == id ) {
38              visited[1] <- visited[1] + 1
39          } else if ( sigma == as.cycle( "(12)" ) ) {
40              visited[2] <- visited[2] + 1
41          } else if ( sigma == as.cycle( "(13)" ) ) {
42              visited[3] <- visited[3] + 1
43          } else if ( sigma == as.cycle( "(23)" ) ) {
44              visited[4] <- visited[4] + 1
45          } else if ( sigma == as.cycle( "(123)" ) ) {
46              visited[5] <- visited[5] + 1
47          } else { # as.cycle( "(132)" )
48              visited[6] <- visited[6] + 1
49          }
50      }
51  }
52
53  stationary <-
54      c(1, theta, theta, theta, theta^2, theta^2)/(1 + 3*theta + 2*theta^2)
55
56  totalvariation = (1/2)* sum( abs(visited/iterations - stationary) )
57
58  cat("Total Variation Distance of Empirical to Theoretical:"
59  totalvariation)
```

.

13.13. Problems to Work for Understanding

1: Show that the transition probability matrix for the Metropolis algorithm on the symmetric group S_3 is

$$
\begin{array}{c}
 \\
[123] \\
[213] \\
[321] \\
[132] \\
[231] \\
[312]
\end{array}
\begin{array}{cccccc}
[123] & [213] & [321] & [132] & [231] & [312] \\
\end{array}
\left(
\begin{array}{cccccc}
1-\theta & \frac{\theta}{3} & \frac{\theta}{3} & \frac{\theta}{3} & 0 & 0 \\
\frac{1}{3} & \frac{2}{3}(1-\theta) & 0 & 0 & \frac{\theta}{3} & \frac{\theta}{3} \\
\frac{1}{3} & 0 & \frac{2}{3}(1-\theta) & 0 & 0 & \frac{\theta}{3} & \frac{\theta}{3} \\
\frac{1}{3} & 0 & 0 & \frac{2}{3}(1-\theta) & \frac{\theta}{3} & \frac{\theta}{3} \\
0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\
0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\
\end{array}
\right).
$$

2: Choose a value of θ ($0 < \theta < 1$) and run the Markov chain on S_3 for 10 iterations. How far is each row from the stationary distribution?

3:   Prove that any permutation of n elements will be produced by the Knuth shuffle algorithm with probability exactly $1/n!$, thus yielding a uniform distribution over all such permutations.

Markov Chain Monte Carlo

The **Markov chain Monte Carlo** (MCMC) method depends on the observation:

> If π is the stationary distribution for an irreducible, aperiodic Markov chain, then the Markov chain can be used to sample from π.

To get a sample, select $s_1 \in \mathcal{X}$ arbitrarily. For any $n > 1$, if $s_{n-1} = i$, select $s_n = j$ with probability p_{ij}. By the Fundamental Theorem for Markov Chains and the Strong Law of Large Numbers the resulting sequence $s_1, s_2, \ldots$ has, as $M \to \infty$,

$$\frac{\text{card}\left(\{n : n \leq M, s_n = j\}\right)}{M} \to \pi_j$$

with probability 1.

Any large but finite part of the sequence approximates a sample from π. Often, one discards the first m terms of the sequence and uses the "tail" of the sequence, $s_{m+1}, s_{m+2}, \ldots, s_M$ as the sample.

Samples from π, however obtained, provide a way to approximate properties of π. Suppose f is a real-valued function on the state space $\mathcal{X}$. To approximate the expected value

$$\mathbb{E}\left[f\right] = \sum_{\nu=1}^{N} f(x_\nu)\pi_\nu$$

select a sample $s_1, s_2, \ldots s_M$ from π and the ergodic theorem guarantees

$$\frac{1}{M} \sum_{\nu=1}^{M} f(s_\nu) \to \mathbb{E}\left[f\right]$$

as $M \to \infty$ with convergence rate $O(M^{-1/2})$.

Given a reasonably sized transition probability matrix for an irreducible, aperiodic Markov chain, it is a standard exercise to find its stationary distribution. The heart of the MCMC method is the inverse problem:

Given a distribution π on a finite state space, find an irreducible, aperiodic Markov chain stationary on π.

Two solutions to this inverse problem are the Metropolis algorithm and the Metropolis-Hastings algorithm. The Metropolis algorithm was motivated by the desire to understand properties of the *Boltzmann distribution* from statistical mechanics, the branch of physics concerned with the average behavior of large systems of interacting particles.

Mathematicians Only: prolonged scenes of intense rigor.

14.1. Chapter Starter Question

In statistical mechanics from physics, what is the Boltzmann distribution and what does it represent?

14.2. A Motivational Example from Statistical Mechanics

Consider a very large but conceptually finite state space $\mathcal{X}$, also called a configuration space $\mathcal{X}$, an energy function $E(\cdot)$ on $\mathcal{X}$ and a fixed temperature T. Let $\tilde{\mathcal{X}}$ be any sample of configurations selected *with replacement* from $\mathcal{X}$. It is desirable to let $\tilde{\mathcal{X}}$ be larger than $\mathcal{X}$. The goal is to change $\tilde{\mathcal{X}}$ so that it becomes approximately a sample from the Boltzmann distribution. Suppose $\mathrm{card}\left(\tilde{\mathcal{X}}\right) = \tilde{N}$ and let N_x denote the number of occurrences of x in $\tilde{\mathcal{X}}$. To say the sample approximates the Boltzmann distribution means

$$\frac{N_x}{N} = \frac{\mathrm{e}^{-E(x)/kT}}{z}$$

for the proportionality or normalizing constant z. The denominator

$$z = \sum_{x' \in \mathcal{X}} \mathrm{e}^{-E(x')/kT}$$

is called the partition function. In any realistic setting, the partition function is intractable, usually due to the size of $\mathcal{X}$. However, for any two configurations x and x'

$$(14.1) \qquad \frac{N_{x'}}{N_x} = \frac{\mathrm{e}^{-E(x')/kT}}{\mathrm{e}^{-E(x)/kT}} = \mathrm{e}^{-(E(x')-E(x))/kT} = \mathrm{e}^{-\Delta E/kT}.$$

This ratio does not depend on the partition function.

Now start from an irreducible, aperiodic Markov chain on $\mathcal{X}$ with the transition probability $P_{x,x'}$ from configuration x to x'. Also, let the transition probability matrix be symmetric, $P_{x,x'} = P_{x',x}$ so the proposed Markov chain is reversible. Allow transitions from configurations with high energy $E(x')$ to lower energy $E(x)$ when they are proposed. Any downhill transition is allowed, the choice is based on convenience and naturalness. The number of times this occurs is $P_{x',x}N_{x'}$.

To reach equilibrium with a complete distribution of energies, not just the low energies achieved by flowing downhill, occasionally allow uphill transitions from configurations with low energy $E(x)$ to high energy $E(x')$ with some probability $\mathbb{P}\left[x \to x'\right]$. The number of times the uphill transition occurs is $P_{x,x'}N_x\mathbb{P}\left[x \to x'\right]$.

Since the Markov chain is reversible, the transition probability is symmetric and $P_{x',x} = P_{x,x'}$. The net flux from configurations with high energy $E(x')$ to low energy $E(x)$ is

$$(14.2) \qquad P_{x',x}N_{x'} - P_{x,x'}N_x \mathbb{P}\left[x \to x'\right] = P_{x,x'}\left(N_{x'} - N_x \mathbb{P}\left[x \to x'\right]\right).$$

In equilibrium, so that the distribution of energies in $\tilde{\mathcal{X}}$ reflects the Boltzmann distribution very well, the net energy flux should be zero. Using equations (14.1) and (14.2) implies that the uphill probability must be

$$\mathbb{P}\left[x \to x'\right] = \frac{N_{x'}}{N_x} = e^{-\Delta E/kT}.$$

This modified Markov chain process will drive an arbitrary distribution of energies toward the Boltzmann distribution for the following reasons. Suppose the distribution has too many configurations with high energy $E(x')$ compared with configurations with the low energy $E(x)$, that is,

$$\frac{N_{x'}}{N_x} > e^{-\Delta E/kT}.$$

In this case, the net flux in equation 14.2 is positive and there will be more transitions from configurations with high energy to those with low energy than in the other direction. The distribution of energies in $\tilde{\mathcal{X}}$ will move properly towards the Boltzmann distribution. Repeating the process will produce a set of configurations whose distribution of energies approximates the Boltzmann distribution. Understanding the rate of convergence is left for further investigation.

The following is a formal statement of the Metropolis algorithm for a suitable proposed transition. For an arbitrary $x \in \mathcal{X}$ define the Metropolis algorithm transition to a configuration $x^\star$ by the following

(1) Define a symmetric transition probability matrix, $P_{x,x'} = P_{x',x}$.

(2) Select y according to the proposed transition probability matrix.

(3) If $E(y) \leq E(x)$ or equivalently $\text{Boltz}(y) \geq \text{Boltz}(y)$, let $x^\star = y$. In other words, always move to a lower or equal energy (equivalently, a greater or equal probability) configuration.

(4) If $E(y) > E(x)$ or equivalently $\text{Boltz}(y) < \text{Boltz}(x')$, let $x^\star = y$ with probability

$$\frac{\text{Boltz}(y)}{\text{Boltz}(x)} = e^{-\Delta E/kT}.$$

In other words, move to a higher energy (lesser probability) configuration with probability $e^{-\Delta E/kT}$.

(5) Otherwise, $x^\star = y$.

This process defines an irreducible, aperiodic Markov chain on the configuration space $\mathcal{X}$. The ratio is crucial to the computational utility of the Metropolis algorithm because it avoids the intractable partition function. The steps of the chain are easily computable, or at least as easy as the proposal transition. In many settings, $\Delta E = E(x') - E(x)$ is simple to compute, often it is independent of card $(\mathcal{X})$. The Markov chain defined by the Metropolis algorithm can be implemented without knowing the entire transition matrix.

14.3. Abstraction of the Metropolis Algorithm

The Metropolis algorithm creates an easily computed Markov chain that is stationary on the desired distribution π. The algorithm requires only the relative weights of the limiting distribution, not the probabilities, thereby avoiding the possibly intractable normalizing constant z.

Let $\mathcal{X}$ be a potentially large but finite state space and $\pi(x)$ a probability mass vector on $\mathcal{X}$, perhaps specified only up to an unknown normalizing constant. Let P be a transition probability matrix on $\mathcal{X}$ with $P(x, x') > 0$ if and only if $P(x', x) > 0$. This is a weaker requirement than reversibility. Using parentheses instead of integer subscripts to index entries is a slight change of notation for matrices. The analogy to the previous example from statistical mechanics, along with a connection to spectral theory used later, motivates the change in notation. At the start P is unrelated to π. The **Metropolis algorithm** changes P to a new transition probability matrix $Q(x, x')$ with stationary distribution π. Define the *acceptance ratio* as

$$A(x, x') = \frac{\pi(x')P(x', x)}{\pi(x)P(x, x')}.$$

The potential lack of a normalizing constant for the distribution π does not matter since it cancels in the acceptance ratio. The acceptance is the ratio of probabilities at equilibrium of moving between x and x'. If $A(x, x') \geq 1$, the probability at equilibrium of moving from x' to x is greater (or equal) probability than probability at equilibrium of moving from x to x', so accept the change from x to x'. That is, accept a change to a place of greater probability. If $A(x, x') < 1$, the probability at equilibrium of moving from x' to x is the lesser probability, so only accept that change with some probability, see the next paragraph for the precise amount. That is, occasionally accept a change to a place of lesser probability.

The Metropolis algorithm changes P to

$$Q(x, x') = \begin{cases} P(x, x') & x \neq x', A(x, x') \geq 1 \\ P(x, x')A(x, x') & x \neq x', A(x, x') < 1 \\ P(x, x') + \sum_{z:A(x,z)<1} P(x, z)(1 - A(x, z)) & x = x'. \end{cases}$$

The Metropolis algorithm has a simple interpretation: from x, choose x' with probability $P(x, x')$ from the original Markov chain. Then if the acceptance ratio $A(x, x') \geq 1$, move to x'; if the acceptance ratio $A(x, x') < 1$, flip a coin with success probability $A(x, x')$ and move to x' if success occurs; in other cases stay at x.

The new chain satisfies

$$\pi(x)Q(x, x') = \pi(x')Q(x', x)$$

(see the exercise) and thus

$$\sum_{x \in \mathcal{X}} \pi(x)Q(x, x') = \sum_{x \in \mathcal{X}} \pi(x')Q(x', x) = \pi(x') \sum_{x \in \mathcal{X}} Q(x', x) = \pi(x').$$

Therefore, π is a left eigenvector with eigenvalue 1. If the Markov chain is connected, then the Fundamental Theorem of Markov Chains is in effect. After many steps

of the chain, it should have converged to a stationary distribution. That is, the chance of being at x is approximately $\pi(x)$, no matter what the starting state.

14.4. Metropolis-Hastings Algorithm

In a 1970 paper, the statistician W. K. Hastings cast the Metropolis algorithm as a general purpose sampling algorithm, distilling the Metropolis algorithm to mathematical essentials. The following is Hastings's generalization of the Metropolis algorithm, known as the **Metropolis-Hastings algorithm**. Given a distribution π to sample, select any transition probability matrix $P = (p_{ij})$ on the state space. Here the notation for matrix entries reverts again to subscripts since the application is typically for matrices on finite dimensional spaces. Unlike the Metropolis algorithm, P does not need the symmetric positivity requirement. Define the transition probability matrix $Q = (q_{ij})$ by

$$Q_{ij} = \begin{cases} p_{ij}\alpha_{ij} & \text{if } i \neq j \\ 1 - \sum_{\nu \neq i} p_{i\nu}\alpha_{i\nu} & \text{if } i = j \end{cases}$$

where α_{ij} is given by

$$\alpha_{ij} = \frac{s_{ij}}{1 + \frac{\pi_i}{\pi_j}\frac{p_{ij}}{p_{ji}}}.$$

The values s_{ij} can be quite general so long as $s_{ij} = s_{ji}$ for all i and j and $0 \leq \alpha_{ij} \leq 1$. With this choice of s_{ij} it is easy to verify that π is the unique stationary distribution for Q. For a symmetric P, a simple choice of s_{ij} recovers the original Metropolis algorithm.

For a given distribution π, different choices of s_{ij} give different Metropolis-Hastings algorithms, all of which result in a Markov chain with stationary distribution on π. P. H. Peskun showed that among all choices of the s_{ij}, the variance of the estimate

$$\frac{1}{M}\sum_{\nu=1}^{M} f(s_\nu) \to \mathbb{E}\left[f\right]$$

is asymptotically minimal for the choice that leads to the Metropolis algorithm.

14.5. Convergence

A basic problem of Markov chain theory but especially for Markov chain Monte Carlo theory is: What is the rate of convergence in $P^n(x, y) \to \pi(x)$? How long must the chain run to be suitably close to π? It is customary to measure distance between probabilities by the total variation distance

$$\|P_x^n - \pi\|_{TV} = \frac{1}{2}\sum_y |P_x^n - \pi| = \max_{A \subset \mathcal{X}} |P^n(x, A) - \pi(A)|.$$

Probability distributions P_x^n and π are far apart in total variation distance if there is a "bad event" A such that P_x^n and π measure A differently. The goal is to show that for large n the Markov chain and the stationary distribution do not share a "bad event". This yields the problem: Given P, π, x, $\epsilon > 0$, how large must n be so that $\|P_x^n - \pi\|_{TV} < \epsilon$? This question has a precise answer for only a few problems, see for example the chapter on Card Shuffling.

Suppose the Markov chain is reversible, so $\pi(x)P(x,y) = \pi(y)P(y,x)$. In words, reversibility means that at equilibrium the probability of going from state x to state y is the same as going from y to x. Here again the notation uses parentheses for matrix and vector entries instead of subscripts. The parenthesis notation emphasizes the following connection to L^2 theory. Let the space be $L^2(\pi)$, the set of mappings $\{\mathbf{g} : \mathcal{X} \to \mathbb{R}\}$ with inner product $\langle \mathbf{g}, \mathbf{h} \rangle = \sum_x g(x)h(x)\pi(x)$. Take P to operate on L^2 by usual matrix multiplication $P\mathbf{g}(x) = \sum_y P(x,y)g(y)$. The property of reversibility implies $\langle P\mathbf{g}, \mathbf{h} \rangle = \langle \mathbf{g}, P\mathbf{h} \rangle$, so P is self-adjoint, see the exercises. The Spectral Theorem says the space has an orthonormal basis (with respect to the inner product) of right (or column) eigenvectors ψ_i and real eigenvalues λ_i, so $P\psi_i = \lambda_i \psi_i$ for $1 \leq i \leq \operatorname{card}(\mathcal{X})$ ordered by $1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{\operatorname{card}(\mathcal{X})} \geq -1$. Here subscripts are indexing the sets of eigenvectors and eigenvalues, not entries of vectors or matrices. Each row of P sums to 1, so $\mathbf{1} = (1,1,\ldots,1)^T$ is a right eigenvector corresponding to eigenvalue 1. Additionally, with respect to the weighted inner product $\|\mathbf{1}\|^2 = \langle \mathbf{1}, \mathbf{1} \rangle = \sum_x 1 \cdot 1 \cdot \pi(x) = 1$, so without loss of generality use $\lambda_1 = 1$ and $\psi_1 = \mathbf{1}$ in the convergence estimate below.

By standard inner product space manipulations (see the exercises)

$$P(x,y) = \pi(y) \sum_{\nu=1}^{\operatorname{card}(\mathcal{X})} \lambda_\nu \psi_\nu(x)\psi_\nu(y),$$

$$P^n(x,y) = \pi(y) \sum_{\nu=1}^{\operatorname{card}(\mathcal{X})} \lambda_\nu^n \psi_\nu(x)\psi_\nu(y).$$

Using the total variation metric definition and the Cauchy-Schwartz inequality

$$2\|P_x^n - \pi\|_{TV} = \sum_y |P^n(x,y) - \pi(y)|$$

$$= \sum_y |P^n(x,y) - \pi(y)| \cdot \frac{\sqrt{\pi(y)}}{\sqrt{\pi(y)}}$$

$$\leq \sqrt{\sum_y \frac{|P^n(x,y) - \pi(y)|^2}{\left(\sqrt{\pi(y)}\right)^2}} \cdot \sqrt{\sum_y \left(\sqrt{\pi(y)}\right)^2}$$

$$= \sqrt{\sum_y \frac{|P^n(x,y) - \pi(y)|^2}{\pi(y)}}.$$

Then

$$4\|P_x^n - \pi\|_{\mathrm{TV}}^2 \leq \sum_y \frac{(P^n(x,y) - \pi(y))^2}{\pi(y)}$$

$$= \sum_y \frac{\left(\pi(y) \sum_{\nu=1}^{\mathrm{card}(\mathcal{X})} \lambda_\nu^n \psi_\nu(x)\psi_\nu(y) - \pi(y)\right)^2}{\pi(y)}$$

$$= \sum_y \pi(y) \left(\sum_{\nu=1}^{\mathrm{card}(\mathcal{X})} \lambda_\nu^n \psi_\nu(x)\psi_\nu(y) - 1\right)^2.$$

Expand the squared term, distribute the $\pi(y)$ terms and interchange the order of summation, factoring out terms not depending on y.

$$= \sum_y \pi(y) \left(\sum_{\nu=1}^{\mathrm{card}(\mathcal{X})} \lambda_\nu^{2n} \psi_\nu(x)^2 \psi_\nu(y)^2 + 2 \sum_{\nu=1}^{\mathrm{card}(\mathcal{X})} \sum_{\mu=\nu+1}^{\mathrm{card}(\mathcal{X})} \lambda_\nu \lambda_\mu \psi_\nu(x)\psi_\nu(y)\psi_\mu(x)\psi_\mu(y)\right.$$
$$\left. - 2 \sum_{\nu=1}^{\mathrm{card}(\mathcal{X})} \lambda_\nu \psi_\nu(x)\psi_\nu(y) + 1\right)$$

$$= \sum_y \left(\sum_{\nu=1}^{\mathrm{card}(\mathcal{X})} \lambda_\nu^{2n} \psi_\nu(x)^2 \psi_\nu(y)^2 \pi(y) + 2 \sum_{\nu=1}^{\mathrm{card}(\mathcal{X})} \sum_{\mu=\nu+1}^{\mathrm{card}(\mathcal{X})} \lambda_\nu \lambda_\mu \psi_\nu(x)\psi_\mu(y)\psi_\mu(x)\psi_\mu(y)\pi(y)\right)$$
$$- 2 \sum_{\nu=1}^{\mathrm{card}(\mathcal{X})} \lambda_\nu \psi_\nu(x)\psi_\nu(y)\pi(y) + \pi(y)$$

$$= \sum_{\nu=1}^{\mathrm{card}(\mathcal{X})} \lambda_\nu^{2n} \psi_\nu(x)^2 \sum_y \psi_\nu(y)^2 \pi(y)$$
$$+ 2 \sum_{\nu=1}^{\mathrm{card}(\mathcal{X})} \sum_{\mu=\nu+1}^{\mathrm{card}(\mathcal{X})} \lambda_\nu \lambda_\mu \psi_\nu(x)\psi_\mu(x) \sum_y \psi_\nu(y)\psi_\mu(y)\pi(y)$$
$$- 2 \sum_{i=1}^{\mathrm{card}(\mathcal{X})} \lambda_i \psi_i(x) \sum_y \psi_i(y)\pi(y) + \sum_y \pi(y)$$

By orthonormality, $\sum_y \psi_\nu(y)^2 \pi(y) = \langle \psi_\nu, \psi_\nu \rangle = 1$ for $\nu = 1, \ldots, \mathrm{card}(\mathcal{X})$. By orthogonality, $\sum_y \psi_\nu(y)\psi_\mu \pi(y) = \langle \psi_\nu, \psi_\mu \rangle = 0$ for $\nu = 1, \ldots, \mathrm{card}(\mathcal{X})$ and $\mu = \nu+1, \ldots, \mathrm{card}(\mathcal{X})$. Recalling $\lambda_1 = 1$ and $\psi_1 = \mathbf{1}$, by orthogonality $\sum_y \psi_i(y)\pi(y) = \sum_y 1 \cdot \psi_i(y)\pi(y) = \langle \psi_\mathbf{1}, \psi_\mathbf{i} \rangle = 0$ for $i = 2, \ldots \mathrm{card}(\mathcal{X})$ while $\sum_y \psi_1(y)\pi(y) = 1$.

Also, $\sum_y \pi(y) = 1$. This leaves

$$= \lambda_1^{2n} \psi_1(x)^2 + \sum_{i=2}^{\text{card}(\mathcal{X})} \lambda_i^{2n} \psi_i(x)^2 - 2\lambda_1^n \psi_1(x) + \sum_y \pi(y)$$

$$= 1 + \sum_{i=2}^{\text{card}(\mathcal{X})} \lambda_i^{2n} \psi_i(x)^2 - 2 + 1$$

$$= \sum_{i=2}^{\text{card}(\mathcal{X})} \lambda_i^{2n} \psi_i(x)^2.$$

Summarizing

(14.3) $$\|P_x^n - \pi\|_{\text{TV}}^2 \leq \frac{1}{4} \sum_{i=2}^{\text{card}(\mathcal{X})} \lambda_i^{2n} \psi_i(x)^2.$$

This bound is the basic eigenvalue bound used to get rates of convergence for the Markov chain Monte Carlo method. Recalling Theorem 11 from Stationary Distributions, there exist constants $\alpha \in (0,1)$ and $C > 0$ such that

$$\max_{x \in \mathcal{X}} \|(P^n)_{i\cdot} - \pi\|_{TV} \leq C\alpha^n.$$

The results here give more information, but to get good bounds on the right hand side requires knowledge of both eigenvalues and eigenvectors.

14.6. The Weather Example

Recall the magical land of Oz where the weather follows a pattern. If it is raining today (R), then tomorrow has a 50% chance of raining again, and a 25% chance of either having a nice day (N) or a snowy day (S) tomorrow. Similarly if it is snowing, the weather has a 50% chance of again having snow, and a 25% chance of either having a nice day, or a rainy day. Also the land of Oz never has two nice days in a row, and equally has rain or snow the day after a nice day. Further recall that for P^n, for large values of n, the row vectors approach the stationary distribution $\pi = (2/5, 1/5, 2/5)$ in the order (R, N, S).

Take this distribution as the desired distribution π. To illustrate the Markov chain Monte Carlo method, use the Metropolis-Hastings algorithm. For simplicity and in the absence of any other reasonable transition probability matrix, take the initial transition probability matrix to be uniform on the state space of weather days, $P = (p_{ij})$ with $p_{ij} \equiv \frac{1}{3}$. Then α_{ij} is given by

$$\alpha_{ij} = \frac{s_{ij}}{1 + \frac{\pi_i}{\pi_j} \frac{p_{ij}}{p_{ji}}}$$

with $s_{ij} \equiv 1$ again for simplicity and the lack of any other reasonable choice. With these choices,

$$\alpha = \begin{pmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{2} \\ \frac{2}{3} & \frac{1}{2} & \frac{2}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{2} \end{pmatrix}.$$

Following the construction of the Metropolis-Hastings algorithm

$$Q = \begin{pmatrix} \frac{13}{18} & \frac{1}{9} & \frac{1}{6} \\ \frac{2}{9} & \frac{5}{9} & \frac{2}{9} \\ \frac{1}{6} & \frac{1}{9} & \frac{13}{18} \end{pmatrix}.$$

Then $\pi = (2/5, 1/5, 2/5)$ is a stationary distribution of Q, just as it is for P. Comparing the rates of convergence is interesting too. The eigenvalues of P are $1, 1/4, -1/4$, so the rate of convergence of P^n is about like $(1/4)^n$. On the other hand, the eigenvalues of Q are $1, 5/9, 4/9$, so the rate of convergence of Q^n is about like $(5/9)^n$, much slower. This is not surprising, since sampling the stationary distribution with the Metropolis algorithm is not as direct.

The Markov chain with Q as transition probability matrix is reversible. After 5 steps the square of the total variation distance of the distribution from state 1, Rain, from the stationary distribution is approximately 7.95×10^{-4} while the bound from (14.3) is 8.94×10^{-4}. After 10 steps the square of the total variation distance of the distribution from state 1 from the stationary distribution is approximately 2.05×10^{-6} while the bound from (14.3) is 2.45×10^{-6}. Regression analysis gives

$$\|(Q^n)_{1,\cdot} - \pi\|_{TV} \approx 0.5295 \cdot (0.5538)^n,$$

about what is expected.

14.7. Example with Continuous State Space

Consider three particles each randomly located in the square $[0,1]^2 \subset \mathbb{R}^2$ with positions (x_{i1}, x_{i2}) for $i = 1, 2, 3$. The state space is $\mathcal{X} = [0,1]^6$. Suppose the positions of the particles are distributed according to the density

$$\pi(x) = \pi(x_1, x_2, x_3) = \frac{1}{z} \exp\left[-C \sum_{i=1}^{3} \|x_i\| - D \sum_{i<j} \frac{1}{\|x_i - x_j\|} \right]$$

where $\|\cdot\|$ is the usual Euclidean norm on $\mathbb{R}^2$, C and D are fixed positive constants and z is the normalizing constant or partition function. In this density the first sum pushes the particle towards the origin, the second sum pushes them away from each other.

To create a Markov chain with π as its stationary distribution, use the Metropolis algorithm. Given X_n, first choose $y \in [0,1]^6$ from the uniform distribution on $\mathcal{X}$. Then with probability $r = \min[1, \pi(y)/\pi(x)]$, accept $X_{n+1} = y$, otherwise reject y and $X_{n+1} = x$. Then this Markov chain has π as its stationary distribution. The later section on Convergence to the Stationary Distribution will show that with $C = D = \frac{1}{10}$ the rate of convergence is

$$\| \operatorname{dist}(X_n) - \pi\|_{TV} \leq (0.883)^n$$

so that after 38 steps, the total variation distance between the Markov chain and the stationary distribution is less than 0.01.

A plot of the stationary distribution is Figure 1.

Figure 1. Plot of points x_1, x_2, x_3 from step 100 to step 3000 in the Metropolis algorithm simulation of the stationary distribution π for the continuous state space Markov chain.
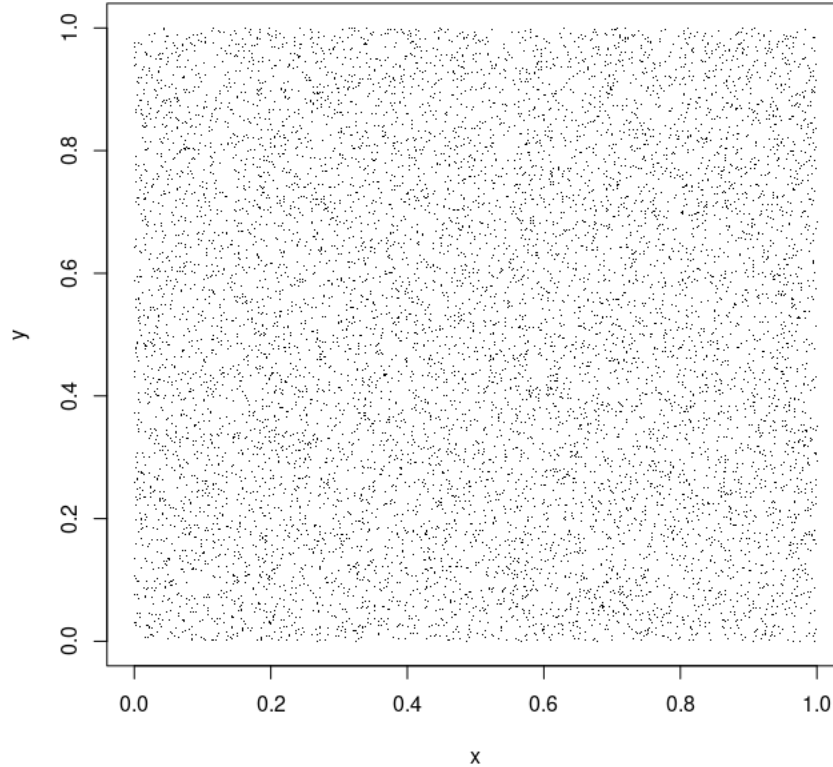
14.8. Example with Unbounded Continuous State Space

Let the state space be $\mathcal{X} = \mathbb{R}$ with target density the double exponential or Laplace distribution. $\pi(x) = \mathrm{e}^{-|x|}/2$. It is easy to simulate the double exponential density using the standard technique of choosing a uniform random deviate and then applying the inverse of the cumulative distribution function, so this is an illustrative example rather than a practical simulation method for the double exponential distribution.

Use another version of the Metropolis algorithm. Choose the initial state x_0 uniformly at random from the interval $[-2, +2]$. At succeeding steps, first propose to move from state x to y chosen uniformly at random from the interval $[x-2, x+2]$. With probability $\min[1, \pi(y)/\pi(x)]$, accept the proposal y which becomes the new state, otherwise reject it and remain at x. Again, this procedure creates a Markov chain with π as its stationary distribution.

The later section on Convergence to the Stationary Distribution will show that a theoretical upper bound on the rate of convergence is

$$\| \operatorname{dist}(X_n) - \pi \|_{TV} \leq (0.983)^{1+n/439.56} + (2 \cdot (20.04)^{n/439.56})(0.993)^n$$

so that after 120,000 steps, the total variation distance between the Markov chain and the stationary distribution is less than 0.01. The R script in the Algorithms section shows that after about 48,000 steps total variation distance between this Markov chain and the stationary distribution is less than 0.05. Either way, the convergence to a stationary distribution is much slower than other examples.

14.9. Chapter Ending Answer

In statistical mechanics and mathematics, a Boltzmann distribution is a probability distribution or probability measure that gives the probability that a system will be in a certain state or "configuration" as a function of that state's energy and the temperature of the system. The distribution is expressed in the form:

$$p_i \propto \mathrm{e}^{-\frac{\varepsilon_i}{kT}}$$

where p_i is the probability of the system being in state i, ε_i is the energy of that state, and a constant kT of the distribution is the product of Boltzmann's constant k and thermodynamic temperature T. With this distribution, states with lower energy have a higher probability of being occupied.

14.10. Chapter Summary

Key Concepts.

(1) With Markov chain Monte Carlo, to get a sample, select $s_1 \in \mathcal{X}$ arbitrarily. If $s_{k-1} = i$, select $s_k = j$ with probability p_{ij}. By the Fundamental Theorem for Markov Chains and the Strong Law of Large Numbers for any $k > 1$, the resulting sequence $s_1, s_2, \ldots$ has, as $M \to \infty$,

$$\frac{\operatorname{card}(\{k : k \leq M, s_k = j\})}{M} \to \pi_j$$

with probability 1.

(2) The heart of the MCMC method is the inverse problem: Given a distribution π on a finite state space, find an irreducible, aperiodic Markov chain that is stationary on π. The solution to this inverse problem is the **Metropolis algorithm**.

(3) The formal statement of the Metropolis algorithm for a suitable proposed transition is: For an arbitrary $x \in \mathcal{X}$ define the transition to a configuration $x^\star$.

  (a) Select x' according to the proposed transition.

  (b) If $E(x') \leq E(x)$ or equivalently $\operatorname{Boltz}(x') \geq \operatorname{Boltz}(x)$, let $x^\star = x'$. In other words, always move to a lower energy (higher probability) configuration.

(c) If $E(x') > E(x)$ or equivalently $\text{Boltz}(x') < \text{Boltz}(x')$, let $x^\star = x'$ with probability

$$\frac{\text{Boltz}(x')}{\text{Boltz}(x)} = e^{-\Delta E/kT}.$$

(d) Otherwise, $x^\star = x'$.

(4) For the Metropolis-Hastings algorithm, given a distribution π to sample, select any transition probability matrix $P = (p_{ij})$ on the state space. Unlike the Metropolis algorithm, P does not need to be symmetric. Define the transition probability matrix $Q = (q_{ij})$ by

$$Q_{ij} = \begin{cases} p_{ij}\alpha_{ij} \text{ if } i \neq j \\ 1 - \sum_{k \neq i} q_{ik} \text{ if } i = j \end{cases}$$

where α_{ij} is given by

$$\alpha_{ij} = \frac{s_{ij}}{1 + \frac{\pi_i}{\pi_j}\frac{p_{ij}}{p_{ji}}}.$$

The values s_{ij} can be quite general so long as $s_{ij} = s_{ji}$ for all i and j and $0 \leq \alpha_{ij} \leq 1$.

Vocabulary.

(1) The **Markov chain Monte Carlo** (MCMC) method depends on the observation that if π is the stationary distribution for an irreducible, aperiodic Markov chain, then the Markov chain can be used to sample from π.

(2) The heart of the MCMC method is the inverse problem: Given a distribution π on a finite state space, find an irreducible, aperiodic Markov chain that is stationary on π. Two solutions to this inverse problem are the **Metropolis algorithm** and the **Metropolis-Hastings algorithm**.

(3) In a 1970 paper, the statistician W. K. Hastings cast the Metropolis algorithm as a general purpose sampling algorithm, distilling the Metropolis algorithm to mathematical essentials. Hastings's generalization of the Metropolis algorithm is the **Metropolis-Hastings algorithm**.

(4) Suppose the Markov chain is reversible, so $\pi(x)P(x,y) = \pi(y)P(y,x)$. In words, reversibility means that at equilibrium the probability of going from state x to state y is the same as going from y to x.

Notation.

(1) π – the stationary distribution for an irreducible, aperiodic Markov chain

(2) $\mathcal{X}$ – state space for Markov chain

(3) s_1 – arbitrary element of the state space

(4) $n > 1$ – arbitrary step in Markov chain sample

(5) s_{n-1}, s_n – successive steps in Markov chain sample

(6) p_{ij} – transition probability from state i to state j

(7) M, m – large sample sizes in Markov chain sample

(8) f – arbitrary function on the state space

(9) $E(\cdot)$ – energy function

(10) T – temperature

(11) $\tilde{\mathcal{X}}$ be any sample of configurations selected *with replacement* from $\mathcal{X}$

(12) $\mathrm{card}\left(\tilde{\mathcal{X}}\right) = \tilde{N}$

(13) N_x – the number of occurrences of x in $\tilde{\mathcal{X}}$

(14) $z = \sum_{x' \in \mathcal{X}} \mathrm{e}^{-E(x')/kT}$ – the partition function.

(15) k – Boltzmann constant

(16) $\mathrm{Boltz}(x')$ – Boltzmann probability density

(17) $P(x, x')$ – transition probability matrix on $\mathcal{X}$

(18) $P = (p_{ij})$ – transition probability matrix on the state space

(19) Q – alternative transition probability matrix from Metropolis or Metropolis Hastings algorithm

(20) $L^2(\pi)$ – the set of mappings $\{\vec{g} : \mathcal{X} \to \mathbb{R}\}$ with inner product $\langle \vec{g}, \vec{h} \rangle = \sum_x g(x)h(x)\pi(x)$

(21) β_i – real eigenvalues

(22) $\vec{\psi_i}$ – eigenvectors

(23) (x_{i1}, x_{i2}) for $i = 1, 2, 3$ – positions for three particles each randomly located in the square $[0, 1]^2 \subset \mathbb{R}^2$

(24) C and D – fixed positive constants

14.11. Sources

This section is adapted from: "The Evolution of Markov Chain Monte Carlo Methods" by Matthew Richey [**Ric10**] and "The Markov chain Monte Carlo revolution" by Persi Diaconis [**Dia09**]. The examples of the continuous state space and the unbounded state space is adapted from [**JLL$^+$21**].

14.12. Reading Suggestion:

14.13. Outside Readings and Links:

(1)

(2)

(3)

(4)

14.14. Algorithms and Scripts

Algorithm.

Data: Desired stationary distribution
Result: Approximate stationary distribution

1 *Initialization of Matrices*
2 Load Markov chain library
3 Set number of states
4 Set simple uniform transition probability matrix
5 Set state names
6 Set desired stationary distribution
7 Set up matrix with s values
8 Set matrix with α values according to algorithm
9 Set new probability transition matrix
10 Set markovchain object
11 Set length of long sample path and start position for sample
12 Slice the long sample path from the transient time to the end
13 In the slice count the appearance of each state
14 Empirical stationary distribution

Algorithm 15: Metropolis-Hastings algorithm

Scripts.

R: R script for metHast.R

```r
library("markovchain")

nStates <- 3
P <- matrix(rep(1/nStates, nStates * nStates),
            nrow=nStates, byrow=TRUE)
stateNames <- c("Rain", "Nice", "Snow")
rownames(P) <- stateNames
colnames(P) <- stateNames

pi <-  c(2/5, 1/5, 2/5)
s <- matrix(rep(1, nStates * nStates),
            nrow=nStates, byrow=TRUE)

alpha <- matrix(0, nStates,nStates)
for (i in 1:nStates) {
    for (j in 1:nStates) {
        alpha[i, j] <- s[i, j] * (1 /( 1 + (pi[i]/pi[j])*(P[i,j]/P[j,i])))
    }
}

Q <- matrix(0, nStates,nStates)
for (i in 1:nStates) {
    for (j in 1:nStates) {
        if (i != j) {
            Q[i, j] <- P[i, j] * alpha[i, j]
        }
    }
    Q[i, i] <- 1 - sum( Q[i, ] )
}

hastMetWeather <- new("markovchain", states = stateNames,
                      transitionMatrix = Q,
                      name = "hastMetWeather")
cat( "Steady State Distribution: ", steadyStates(hastMetWeather), "\n" )
```

```
36  start <- 900
37  end <- 1000
38  hastMetWeatherSim <- rmarkovchain(n=end, object=hastMetWeather,
39                                    t0="Nice")
40  cat("Simulation Results\n")
41  cat( "Rain:", sum( hastMetWeatherSim[(start+1):end] == "Rain")/(end-start),
        "\n")
42  cat( "Nice:", sum( hastMetWeatherSim[(start+1):end] == "Nice")/(end-start),
        "\n")
43  cat( "Snow:", sum( hastMetWeatherSim[(start+1):end] == "Snow")/(end-start),
        "\n")
```

.

14.15. Problems to Work for Understanding

1: For the Metropolis algorithm, show that

$$\pi(x)Q(x, x') = \pi(x')Q(x', x).$$

$+$ 2: Consider the following Markov Chain: N black balls and N white balls are placed in two urns so that each contains N balls. At each step one ball is selected at random from each urn and the two balls interchange urns. Let $N = 3$ and the state of the system is the number of white balls in the first urn.

(a) What is the probability transition matrix P?

(b) What is the stationary distribution?

(c) What is the matrix α of entries α_{ij} in the Metropolis-Hastings algorithm?

(d) Choose a satisfactory transition matrix and then create the matrix Q in the Metropolis-Hastings algorithm.

(e) Find the eigenvalues of P and the approximate rate of convergence to the stationary distribution.

(f) Find the eigenvalues of Q and the approximate rate of convergence to the stationary distribution.

3: Consider the second Ehrenfest urn model with $N = 4$, $p = q = \frac{1}{2}$ and transition probability matrix

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{8} & \frac{1}{2} & \frac{3}{8} & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & \frac{3}{8} & \frac{1}{2} & \frac{1}{8} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

(a) Show that this Markov chain is irreducible, positive recurrent, aperiodic.

(b) Find the stationary distribution π.

(c) Show the Markov chain is reversible.

(d) Find the eigenvalues and corresponding eigenvectors, ordered by decreasing eigenvalue.

(e) Using the inner product weighted by π, find the normalized orthonormal eigenvector basis.

(f) Show that

$$P(x, y) = \pi(y) \sum_{i=1}^{5} \lambda_i \psi_i(x) \psi_i(y).$$

4:

(a) Using the results from the previous problem, find the upper bound (14.3) for the square of the total variation distance of the distribution starting from state 5, with all balls in the first urn, from the stationary distribution.

(b) With numerical computation, check the inequality for 5 and 10 steps of the Markov chain.

5:    Suppose the Markov chain is reversible, so $\pi(x)P(x, y) = \pi(y)P(y, x)$. Let the space be $L^2(\pi)$, the set of mappings $\{g : \mathcal{X} \to \mathbb{R}\}$ with inner product $\langle \mathbf{g}, \mathbf{h}\ rangle = \sum_x \mathbf{g}(x)\mathbf{h}(x)\pi(x)$. Then P operates on $L^2(\pi)$ by multiplication $P\mathbf{g}(x) = \sum_y P(x, y)\mathbf{g}(y)$. Show that the property of reversibility implies $\langle P\mathbf{g}, \mathbf{h}\rangle = \langle \mathbf{g}, P\mathbf{h}\rangle$, so P is self-adjoint.

6:    For self-adjoint matrix P and stationary distribution π, with an orthonormal basis (with respect to the inner product $\langle \mathbf{g}, \mathbf{h}\rangle = \sum_x \mathbf{g}(x)\mathbf{h}(x)\pi(x)$) with right eigenvectors ψ_i and real eigenvalues λ_i, so $P\psi_i = \lambda_i \psi_i$ for $1 \leq i \leq \mathrm{card}\,(\mathcal{X})$ ordered by $1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{\mathrm{card}(\mathcal{X})} \geq -1$, show that

$$P(x, y) = \pi(y) \sum_{i=1}^{\mathrm{card}(\mathcal{X})} \lambda_i \psi_i(x) \psi_i(y).$$

The Ising Model

Magnets are common objects but magnetism itself is enigmatic. Heating a ferromagnetic material such as iron causes it to lose magnetism. Slow cooling allows the material to magnetize again. From 1922 to 1925 Wilhelm Lenz and his student Ernst Ising developed what is now called the Ising model to explain this phase transition in magnetism caused by heating or cooling. The Ising model connects electromagnetic theory with statistical mechanics. Exploring the Ising model for the magnetic phase transition uses Markov chains. The goal of this chapter is to describe applications of the Metropolis algorithm to the Ising model for magnetism.

Mathematically Mature: may contain mathematics beyond calculus with proofs.

15.1. Chapter Starter Question

What is a phase transition? What causes phase transitions? What are some examples of phase transitions?

15.2. Magnetism and the Ising Model

In a magnetic material such as iron, the atoms each have a magnetic moment. A piece of iron has small regions or "domains" that are completely magnetized because the atoms align in the same direction. The overall energy of neighboring atoms in the domain is lower when the moments are parallel. If the energy of neighboring atoms and domains were the only consideration, a magnetic material would settle into a magnetized configuration, but heat tends to randomize the magnetic moment directions. On a large scale, the directions of the domains are not aligned so average magnetization is about zero.

In the physics of magnetism, the Curie temperature T_c is the temperature above which certain materials lose their permanent magnetic properties. The Curie temperature is named after Pierre Curie, who showed that magnetism disappears above

a critical temperature. Above T_c thermal energy disrupts the tendency toward magnetic dipole alignment, and materials become just weakly magnetic.

The **Ising model** is a simple model of a ferromagnet's domain alignment due to energy and heat. The model describes the tendency for neighboring atoms to align their magnetic fields with each other or with an external magnetic field. The Ising model exhibits a **phase transition** in the magnetization, a dramatic change between differently ordered magnetic states as a parameter passes through a critical value. Moreover, the 2-dimensional Ising model is one of the most important theoretical statistical mechanical models. As with all mathematical models, the aim is to find the simplest framework that exhibits important properties of the physical system. With just the simple elements of binary variables on a regular grid, the Ising model exhibits phase transitions.

The Ising model is a lattice model of atomic magnetic dipoles aligned up or down in some fixed direction. A 1-dimensional Ising model pictures dipoles regularly spaced along a line. A 2-dimensional Ising model is a square lattice, and a 3-dimensional Ising model is a cubic lattice. Each atom interacts with its nearest neighbors. The goal of the model is to find T_c. Interestingly, the 1-dimensional Ising model does not have a phase transition, first proved by Ernst Ising in 1925. A 1944 result of Lars Onsager gives a relation for the exact value for the critical temperature in the 2-dimensional Ising model:

$$\frac{\mathrm{k_{Boltz}}T_c}{J} = \frac{2}{\log(1 + \sqrt{2})} \approx 2.269$$

where $\mathrm{k_{Boltz}}$ is the Boltzmann constant from statistical mechanics and J is a proportionality constant for the energy of a dipole alignment.

For definiteness for this application of the Metropolis algorithm, assume a planar square lattice with N^2 sites. An $N \times N$ matrix represents the square lattice, the first row at the top, and the Nth row at the bottom, the first column at the left, the Nth column at the right. At each lattice site (i, j), there is a *spin* represented by $\omega_{i,j} = \pm 1$. A configuration is a matrix of spins, $\omega = (\omega_{1,1}, \omega_{1,2}, \ldots, \omega_{N,N})$. Define Ω as the set of all possible configurations so $\mathrm{card}(\Omega) = 2^{N^2}$. The magnetic energy (also called the Hamiltonian) of a configuration is

$$E(\omega) = -J \sum_{i,j=1}^{N} \sum_{k=1}^{4} \omega_{i,j} \omega_{\langle i,j \rangle [k]} - H \sum_{i=1}^{N} \omega_i$$

where $J > 0$ is the nearest-neighbor affinity (a proportionality constant with units of energy for the strength of the binding between neighbors), $H \geq 0$ represents the external magnetic field and $\langle i, j \rangle$ indicates the set of 4 sites that are periodic nearest neighbors, sites in the lattice with a horizontal or vertical bond. For the top row with $i = 1$, periodic means $\langle 1, j \rangle = \{(2, j), (1, j - 1), (N, j), (1, j + 1)\}$, interpreting periodically if $j = 1$ or $j = N$. Similarly for the bottom row, and the leftmost and rightmost columns. The periodic boundary conditions mean that boundary sites are not treated differently than interior sites and make the finite grid simulate an infinite grid. For purposes of the example, assume there is no external field, so $H = 0$ and that $J = 1$.

The probability distribution of spin configurations is proportional to the **Boltzmann weight** $\text{Boltz}(\omega) = \mathrm{e}^{-E(\omega)/(\mathrm{k_{Boltz}}T)}$ where $\mathrm{k_{Boltz}}$ is the Boltzmann constant and T is the absolute temperature, so $\mathrm{k_{Boltz}}T$ has units of energy. The energy units are chosen to make $\mathrm{k_{Boltz}}T_C = \frac{2}{\log(1+\sqrt{2})}$. This section uses the combination $\mathrm{k_{Boltz}}T$ consistently for the heat energy at temperature T to emphasize the connection with statistical mechanics, with $0 \leq \mathrm{k_{Boltz}}T \leq 10$ for illustration. Given a configuration, the Boltzmann weight is proportional to the probability that configuration will appear at temperature T. For completely ordered configurations with $\omega_{i,j} \equiv +1$ or $\omega_{i,j} \equiv -1$, $E(\omega) = -J \cdot 4N^2$ (low magnetization energy) and $\text{Boltz}(\omega) = \mathrm{e}^{J \cdot 4N^2/(\mathrm{k_{Boltz}}T)}$ is large (relatively higher probability). For highly disordered configurations with $\omega_{i,j} = \pm 1$ distributed randomly, $E(\omega) \approx 0$ (higher magnetization energy) and $\text{Boltz}(\omega) \approx \mathrm{e}^{-0/(\mathrm{k_{Boltz}}T)} \approx 1$ is small (relatively lower probability).

For a configuration ω the net spin is

$$M(\omega) = \sum_{i,j=1}^{N} \omega_{i,j}.$$

The **magnetization** at temperature T is $\mathbb{E}[M]_T$ is the expected value of $M(\omega)$ with respect to the Boltzmann weight:

$$\mathbb{E}[M]_T = \sum_{\omega \in \Omega} M(\omega) \, \text{Boltz}(\omega)$$

$$= \frac{1}{z} \sum_{\omega \in \Omega} M(\omega) \mathrm{e}^{-E(\omega)/(\mathrm{k_{Boltz}}T)}.$$

Recall that the denominator

$$z = \sum_{\omega \in \Omega} \mathrm{e}^{-E(\omega)/(\mathrm{k_{Boltz}}T)}$$

is called the partition function.

This model exhibits some general features. At high temperatures there is no correlation between sites, states are essentially uniformly distributed and hence $\mathbb{E}[M]_T$ is near zero. This highly disordered configuration is not magnetic. Lowering the temperature to a critical value called the Curie temperature T_c, spontaneous magnetization occurs. The model goes to one of the highly ordered state of all $+1$ or all -1 spins. See Figure 1 for a schematic diagram of the phase transition. The Curie temperature is a critical temperature below which sites influence each other at long distances. It is impractical to directly compute the expected magnetization since the number of configurations is too large. Even for $N = 5$, to directly compute the magnetization requires 2^{25} matrices with 25 entries, over 838 million values.

15.3. The Metropolis Algorithm and the Ising Model

Since direct calculation is unfeasible, the goal is to apply the Metropolis algorithm to the Ising model to find T_c. To do that, run the Markov chain Monte Carlo method at different temperatures to find the average magnetization as a function of temperature to find where the phase change occurs. Figure 2 and Figure 3 are
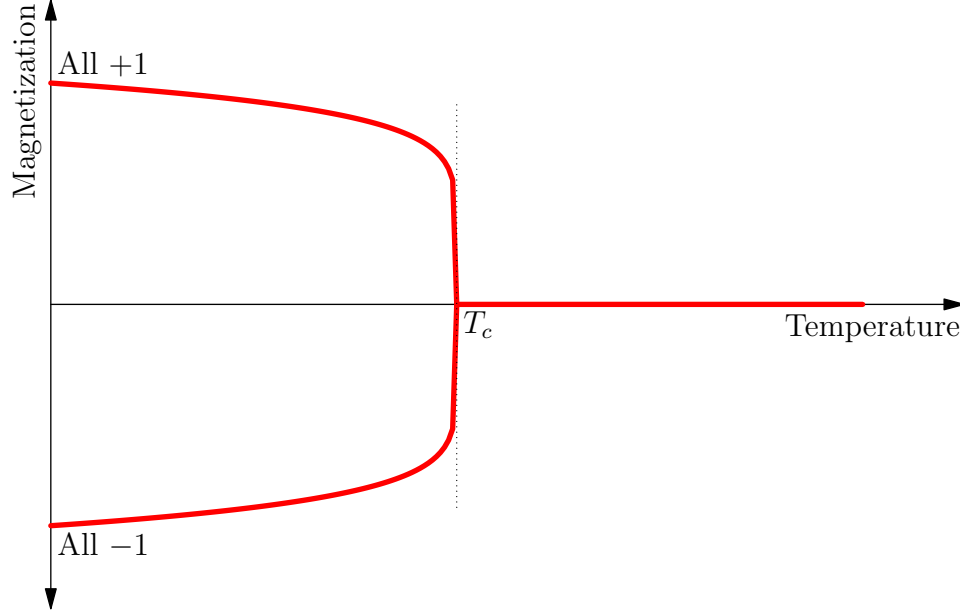
**Figure 1.** Schematic magnetic phase diagram for the Ising model.

snapshots of applying the Metropolis algorithm to the Ising model. Figure 2 shows large areas of consistent magnetization after 500,000 Markov chain steps started from a random configuration at a normalized temperature $2.2 < T_c$. Figure 2 appears random even after 50,000,000 Markov chain steps started from a random configuration at a temperature $5.0 > T_c$.

The key idea behind the algorithm is to emphasize the low energy subset of configurations and discount the rest. The Metropolis algorithm takes a random walk through the configuration space, visiting more frequently occurring spin configurations more often. In the words of the original Metropolis algorithm paper: "Instead of choosing configurations randomly, then weighting them with $e^{-E/(k_{\text{Boltz}}T)}$, instead choose configurations with probability $e^{-E/(k_{\text{Boltz}}T)}$ and weight them evenly". The Metropolis algorithm for the Ising model proceeds as follows:

(1) Visit all lattice sites (i, j) in some deterministic order. For example, visit across the first row, then across the second row, and so on to the last row.

(2) As each site is visited, calculate the change in energy that would result from flipping the spin at the selected site. If ω is the visited configuration and ω' is the configuration with the flipped spin at the visited site, then

$$\Delta E = E(\omega') - E(\omega) = -2J(\omega'_{(i,j)} - \omega_{(i,j)}) \sum_{k=1}^{4} \omega_{\langle i,j \rangle [k]}$$

where the sum is only over the four nearest neighbors of the (i, j) site. The computational cost of updating a site is both small and independent of the size of the lattice.
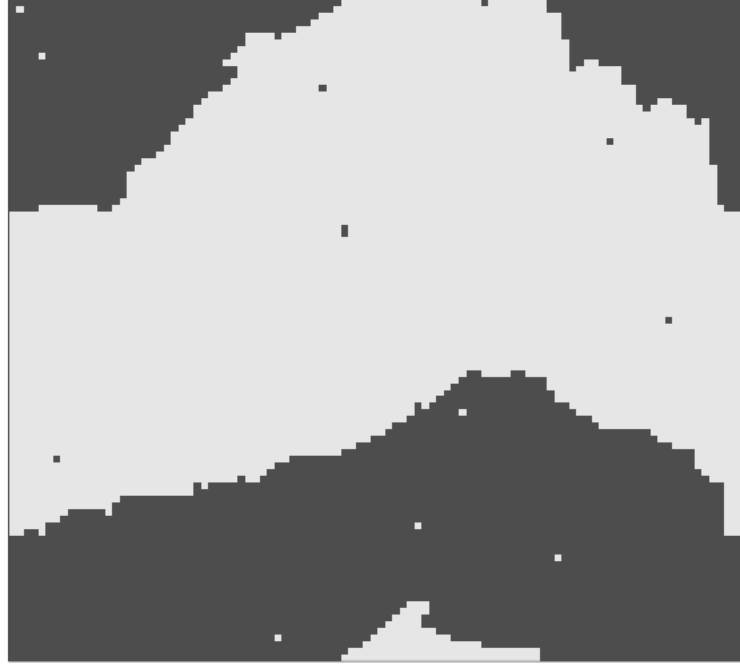
Figure 2. A 100×100 Ising model at $T = 2.2$ after 500,000 steps.

(3) The acceptance rule is: if energy increases, $E(\omega') > E(\omega)$ or equivalently $\text{Boltz}(\omega') < \text{Boltz}(\omega)$, let $\omega^\star = \omega'$ with acceptance probability

$$\frac{\text{Boltz}(\omega')}{\text{Boltz}(\omega)} = e^{-\Delta E / k_{\text{Boltz}} T}.$$

(4) Otherwise, energy decreases, so accept the change, $\omega^\star = \omega'$.

Verifying the Markov chain of configuration changes is irreducible and aperiodic is routine.

The alternative Glauber dynamics algorithm (or in a different context the *random chain algorithm*) is:

(1) The visitation rule between configurations is to pick a lattice site (i, j) uniformly at random from $i, j = 1, 2, 3, \ldots, N$.

(2) As each site is visited, calculate the change in energy that would result from flipping the spin at the selected site. If ω is the visited configuration and ω'
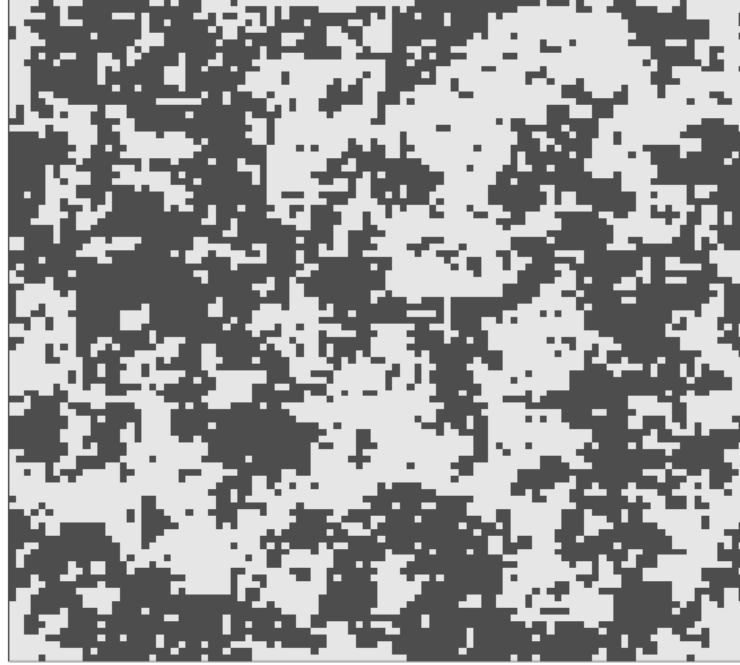
Figure 3. A 100×100 Ising model at $T = 5.0$ after 50,000,000 steps.

is the configuration with the flipped spin at the visited site, then

$$\Delta E = E(\omega') - E(\omega) = -2J\big(\omega'_{(i,j)} - \omega_{(i,j)}\big) \sum_{k=1}^{4} \omega_{\langle i,j \rangle [k]}$$

where the sum is only over the four nearest neighbors of the (i, j) site. The computational cost of updating a site is both small and independent of the size of the lattice.

(3) The current configuration ω becomes the flipped site configuration ω' with acceptance probability

$$\frac{\mathrm{e}^{-\Delta E / \mathrm{k_{Boltz}} T}}{1 + \mathrm{e}^{-\Delta E / \mathrm{k_{Boltz}} T}}.$$

(4) Otherwise, $\omega^{\star} = \omega$.

Verifying the Markov chain of configuration changes is irreducible and aperiodic is routine.

Each algorithm has advantages and disadvantages. For the Metropolis algorithm at low temperatures unfavorable flips are rare, resulting in a (nearly) deterministic flip rule along a deterministic path. This can result in the algorithm getting trapped in patterns in the lattice. On the other hand, with random sampling of sites in the Glauber algorithm, taking 10,000 steps on a 100×100 lattice will miss about $3{,}679 = 10{,}000/e$ sites, see the exercises. Therefore, it takes more steps than the size of the lattice to ensure visiting all sites. Also, the acceptance probability in the Glauber algorithm is less than the acceptance probability in the Metropolis algorithm, see the exercises. So the Glauber algorithm converges more slowly than the Metropolis algorithm. The survey article by Diaconis and Saloff [**DSC98**] presents estimates on rates of convergence indicating slow convergence for the Ising model algorithms considered here.

The two scripts below illustrate the global approach to the Ising model and the Monte Carlo Markov chain method. The script `isingDist.R` calculates the entire configuration space, the individual energy of each configuration and the partition function. However it is only practical for $N = 2, 3, 4$, not refined enough to demonstrate the phase transition. The second script `isingMetropolis.R` illustrates the Monte Carlo Markov chain approach. The script directly implements the Metropolis method above but is not general. For a general approach, see the R package `IsingSampler` [**Eps20**] and for another implementation in Python, see[**Sch18**].

The comprehensive and very readable survey article [**Hay21**] provides interactive graphics for comparing the Metropolis and Glauber algorithms directly, mixing the visitation and acceptance rules for the Metropolis and Glauber algorithms, using different visitation rules for the Metropolis algorithm, and testing the effects of different initial and boundary conditions. The article also has figures for the Boltzman weight function at various temperatures, comparison of the acceptance functions, comparisons of rates of convergence, and explanations for the growth of shapes in the graphics. Finally, the article has a well-researched history of the Metropolis and Glauber algorithms.

15.4. Chapter Ending Answer

A phase transition is dramatic change in state as a parameter passes through a critical value. In physical and chemical systems, the transition occurs from the addition or removal of energy, usually in the form of heat, but it can also result from pressure changes or other causes. Examples of phase transitions are the freezing of water into ice, or the reverse melting, the vaporization of water to steam or the reverse condensation. Biological systems exhibit phase transitions at the molecular, cellular, and even the organism level. In card-shuffling the rapid transition from relatively ordered to completely disordered after $O(n \log n)$ shuffles is an example of a mathematical phase transition.

15.5. Chapter Summary

Key Concepts.

(1) The Ising model is a lattice model of spins for atoms with magnetic dipoles either up or down in some fixed direction. The Ising model exhibits a **phase**

transition in the magnetization, a dramatic change in state as a parameter passes through a critical value.

(2) It is impractical to directly compute the expected magnetization since the number of configurations is too large.

(3) Estimates on rates of convergence indicate slow convergence for the Ising model Metropolis algorithm considered here. The slow convergence is consistent with the number of steps needed to ensure visiting most sites.

Vocabulary.

(1) The **Ising model** is a simple model of a ferromagnet.

(2) A **phase transition** is a dramatic change in state as a parameter passes through a critical value.

(3) The **Boltzmann weight** $\mathrm{Boltz}(\omega) = \mathrm{e}^{-E(\omega)/(\mathrm{k_{Boltz}}T)}$ where $\mathrm{k_{Boltz}}$ is the Boltzmann constant, so $\mathrm{k_{Boltz}}T$ has units of energy.

(4) For a configuration ω let

$$M(\omega) = \sum_{i,j=1}^{N} \omega_{i,j}.$$

(5) The **magnetization** at temperature T is $\mathbb{E}\left[M\right]_T$ is the expected value of $M(\omega)$.

(6) **Glauber dynamics** is the algorithm that selects sites at random in the lattice and computes magnetization in a Markov chain.

Notation.

(1) T_c – the temperature above which certain materials lose their permanent magnetic properties

(2) $\mathrm{k_{Boltz}}$ – the Boltzmann constant from statistical mechanics

(3) J – a proportionality constant for the energy of alignment

(4) N – number of rows and columns in 2-dimensional lattice with N^2 sites

(5) (i,j) – lattice site

(6) $\omega_{i,j} = \pm 1$ – spin at site (i,j)

(7) $\omega = (\omega_{1,1}, \omega_{1,2}, \ldots, \omega_{N,N})$ – configuration as a matrix of spins

(8) Ω – the set of all possible configurations

(9) $E(\omega)$ – magnetic energy of a configuration

(10) $H \geq 0$ – the external magnetic field

(11) $\langle i,j \rangle$ – the set of 4 sites that are periodic nearest neighbors

(12) $\mathrm{k_{Boltz}}$ – Boltzmann constant

(13) $\mathrm{Boltz}(\omega) = \mathrm{e}^{-E(\omega)/(\mathrm{k_{Boltz}}T)}$ – Boltzmann weight

(14) $M(\omega)$ – net spin

(15) $\mathbb{E}\left[M\right]_T$ – the expected value of $M(\omega)$

(16) $z = \sum_{\omega \in \Omega} \mathrm{e}-E(\omega)/(\mathrm{k_{Boltz}}T)$ – the partition function

(17) (i, j) – random site in the matrix

(18) $\omega'_{(i,j)}$ – flipped sign value at (i, j)

(19) ΔE – change in energy from flip

15.6. Sources

Ideas about the Ising model and the associated Metropolis algorithm are adapted from [**Hay19, Hay21, Ric10, Sch18**]. The exercises are inspired by [**Hay21**].

15.7. Reading Suggestion:

15.8. Outside Readings and Links:

(1)

(2)

(3)

(4)

15.9. Algorithms and Scripts

Algorithm.

Scripts.

R: R script for complete Ising model.

```r
## WARNING: USE ONLY FOR N = 2, 3, 4

allBinarySeq <- function(N) { # USE ONLY FOR N <= 4!!
# list of N copies of -1, 1, simplify=FALSE returns lists, not vectors
# then all possible combinations of \pm 1 over N places
# resulting in data.frame of 2^N observations of N variables
# transpose to make N by 2^N matrix
# recast back to data.frame
    as.data.frame(
        t(
            expand.grid(
                replicate(N, c(-1,1), simplify=FALSE)
            )
        )
    )
}

allConfigs <- function(N) { # USE ONLY FOR N <= 4
# returns LIST of all square configs
    X <- allBinarySeq(N*N)
    # data.frame of all 2^(N^2) observations over N^2 variables

    data = lapply(X, matrix, nrow=N, ncol=N)
}

downConfig <- function(N) { matrix(-1,N,N) } # 1 of 2 lowest energy configs
upConfig <- function(N) { matrix(1,N,N) }    # other lowest energy configs

energy  <-  function(config, J=1) {  #pass in a MATRIX
## J has units of energy
## Returns a potential energy, negative, with lowest energy -4N^2
    N   <- NROW(config)

    below  <- config[ c(2:N, 1), ]
```

Data: Dimension N

Result: Plot of magnetization versus normalized temperature near T_c

1 function *allBinarySeq(N)*

2 | creates all 2^N combinations of ± 1 over N places

3 function *allConfigs(N)*

4 | creates a list of all possible square configurations of ± 1 of size $N \times N$
      using `allBinarySeq(`N*N`)`

5 function *energy(config)*

6 | computes magnetization energy

7 function *allEnergies*

8 | applies energy function across list of all energies

9 function *boltzWeight(energy, kTemp)*

10 | computes $\mathrm{e}^{energy * kTemp}$

11 function *netMag(config)*

12 | computes net magnetization

13 function *boltzDist*

14 | computes list of all `boltzWeight(`*energy, kTemp*`)`

15 function *partitionFunct(configurations, kTemp)*

16 | computes sum of all energies using `boltzDist(`*configurations, kTemp*`)`

17

18 Set $J \leftarrow 1$

19 Set $N \leftarrow 3$

20 With `allConfigs(`N`)`, make a list of all configurations of size $N \times N$

21

22 Set number of normalized temperatures to compute

23 Set a list of normalized temperatures of that length from $(2, 2.5)$ covering
      the critical value of $T_c = 2.269$ when $J = 1$

24 Compute the magnetization over all configurations for each of the
      normalized temperatures

25 Plot the magnetization as a function of normalized temperature

Algorithm 16: Ising model for magnetization for sizes $2, 3, 4$.

```
35       left   <- config[ , c(N, 1:(N-1)) ]
36       above  <- config[c(N, 1:(N-1)), ]
37       right  <- config[ , c(2:N, 1) ]
38
39       nearNeighProd <- config * (below + left + above + right)
40       ## element-wise multiplication, note factorization of sum
41
42       nrg  <- -J *sum(nearNeighProd)
43  }
44
45  allEnergies  <- function(configurations) { #list of configurations
46       nrgs  <-  lapply(configurations, energy, J)
47  }
48
49  boltzWeight <- function(nrg, kTemp) { exp(- nrg/kTemp) }
50
51  netMag  <-  function(config) { abs( sum( config ) ) }
52
53  boltzDist  <- function(configurations, kTemperature) {
```

Data: Dimension N, normalized temperature, number of sweeps
Result: Plot of magnetization after many steps of the Metropolis
    algorithm

1 function *Unif(n)*

2 | creates a sample of n uniform random variables from $(0, 1)$

3 function *randomConfig(N)*

4 | creates a random $N \times N$ configuration of ± 1

5 function *boltzWeight(energy, kTemp)*

6 | computes $\mathrm{e}^{-energy/kTemp}$

7 function *surroundNeighbors(site)*

8 | find the 4 periodic surrounding neighbors of *site*

9 function *sweepMetropolis(config, N)*

10 | **for** $x \leftarrow 1 : N$ **do**

11 | | **for** $y \leftarrow 1 : N$ **do**

12 | | | $configAtSite \leftarrow config[x, y]$

13 | | | $configPrimeAtSite \leftarrow configAtSite * (-1)$

14 | | | $neigh \leftarrow \mathsf{surroundNeighbors}(site)$

15 | | | Compute $deltaEnergy$ using $neigh$

16 | | | **if**
    $Unif(1) < \min(1, boltzWeight(deltaEnergy, kTemperature))$
    then

17 | | | | $config \leftarrow configPrime$

18 | | **end**

19 | **end**

20

21 Set $J \leftarrow 1$, $kTemp$, N, $mcmcSteps$

22 Choose random configuration with $\mathsf{randomConfig}(N)$

23 for $1 : mcmcSteps$ **do**

24 | $config \leftarrow \mathsf{sweepMetropolis}(config, N)$

25 end

26 Plot image of resulting configuration

Algorithm 17: Metropolis algorithm applied to Ising model.

```
54      bzs   <- unlist(
55          lapply(allEnergies(configurations), boltzWeight, kTemperature)
56      )
57 }
58
59 partitionFunc  <- function(configurations, kTemperature){ #list of energies
60          Z   <- sum( boltzDist(configurations, kTemperature) )
61 }
62
63 netSpin <- function(configurations, N) {   #list of configurations
64      lm <- lapply(configurations, matrix, N, N)
65      sm   <- abs( unlist( lapply(lm, sum) ) ) #absolute value sum of entries
        in each matrix
66      # unlist converts a list to a vector
67 }
68
69 magnetization   <- function(configurations, kTemperature) {
70      Z <- partitionFunc(configurations, kTemperature)
```

```
71      mag <- (1/Z) * sum(  netSpin(configurations, N) * boltzDist(
        configurations, kTemperature) )
72 }
73
74 J <- 1
75 N   <- 3
76
77 configs  <- allConfigs(N)
78
79 nKTemperatures  <-   11
80 Temperatures  <- seq(2,2.5, length=nKTemperatures)
81 # surrounding the critical value kT_c/J = 2.269 when J=1
82
83 m  <- rep(0, nKTemperatures)
84
85 for (i in 1:nKTemperatures) {
86   m[i] = magnetization(configs, Temperatures[i])
87   cat("T = ", Temperatures[i], "mag = ",  m[i], "\n" )
88 }
89
90 plot(Temperatures, m)
```

.

R: R script for Ising Metropolis model.

```
1 randomConfig <- function(N) {
2   config <- matrix(2*sample(0:1, N*N, replace=TRUE)-1, N,N)
3 }
4
5 boltzWeight <- function(nrg, kBoltzT) { exp(- nrg/kBoltzT) }
6
7 surroundNeighbors  <- function(site) {
8     belowSite  <- if (site[1] == N) c(1, site[2]) else c(site[1]+1, site
       [2])
9     leftSite   <- if (site[2] == 1) c(site[1], N) else c(site[1], site
       [2]-1)
10    aboveSite  <- if (site[1] == 1) c(N, site[2]) else c(site[1]-1, site
       [2])
11    rightSite  <- if (site[2] == N) c(site[1], 1) else c(site[1], site
       [2]+1)
12    neighbors  <- c(belowSite, leftSite, aboveSite, rightSite)
13 }
14
15 sweepMetropolis <- function(config) {
16   for (x in 1:N) {
17     for (y in 1:N) {
18         site <- c(x,y)
19         configAtSite  <- config[ site[1], site[2] ]
20         configPrimeAtSite  <- configAtSite * (-1)
21
22         neigh  <- surroundNeighbors( site )
23         deltaEnergy <- -2*J * (configPrimeAtSite - configAtSite)*
24             (config[ neigh[1], neigh[2] ] +
25              config[ neigh[3], neigh[4] ] +
26              config[ neigh[5], neigh[6] ]+
27              config[ neigh[7], neigh[8] ]
28             )
29         if ( runif(1) <= min(1, boltzWeight(deltaEnergy, kBoltzT)) )
30         {
31         config[ site[1], site[2] ]  <- configPrimeAtSite
32         }
33
34     ## IF deltaEnergy < 0, so energy(configPrime) < energy(config)
35     ## so boltzWeight(deltaEnergy, kBoltzT) > 1,
36     ## THEN  runif(1) is certain to be less than p so certainly
37     ## change config to configPrime
38     ## ELSE deltaEnergy > 0, so
39     ## energy(configPrime) > energy(config), so
40     ## boltzWeight(deltaEnergy, kBoltzT) < 1,
```

```
41        ## so change on chance that runif(1) < p
42      }
43    }
44    return(config)
45  }
46
47  J   <- 1
48  N   <- 100
49  kBoltzT   <- 2.2
50  mcmcSteps   <- 50
51
52  config   <- randomConfig(N)
53
54  for (i in 1:mcmcSteps ) {
55      config   <- sweepMetropolis(config)
56  }
57
58  image(1:N, 1:N, config[N:1, ],
59        col=gray.colors(2),
60        xlab="", ylab="", xaxt="n", yaxt="n")
61  ## config[N:1, ] so that the plot corresponds to matrix order
62  ## i.e. first row of matrix is at top of plot, last row at bottom
```

.

15.10. Problems to Work for Understanding

1: Show that with random sampling of sites, taking 10,000 steps on a 100×100 lattice will miss about $3{,}679 \approx 10{,}000/e$ sites.

2: With the nearest neighbor energy model, show that ΔE can only have the values $-16, -8, 0, +8, +16$. (Use the convention that $J = 1$.)

3: Compare the Metropolis and Glauber acceptance functions by drawing the graphs of the acceptance probabilities over the range of energy changes $-17 \leq \Delta E \leq 17$, one plot with the two functions for each of $k_{\text{Boltz}}T = 10.0, 2.27, 1.0$, and 0.1. Draw a dot on each graph at the possible values of $\Delta T = -16, -8, 0, +8, +16$

4: Show that for both the Metropolis and Glauber Markov chains, all configurations are accessible, so the chain is irreducible.

5: A 30×30 grid of squares contains 900 fleas, initially one flea per square. When a bell is rung, each flea independently jumps to a horizontally or vertically adjacent square at random. For most squares, this is one of 4 possibilities, except for fleas in one of the 116 squares on the edge of the grid or at the corners. For edge or corner squares a fleas will jump to one of the 3 or 2 horizontally or vertically adjacent squares. (Note that this is different from the periodic boundary conditions applied to the Ising model.) What is the expected number of unoccupied squares after 50 rings of the bell? Give your answer rounded to six decimal places.

The Gibbs Sampler

The **Gibbs sampler** is an algorithm for generating random variables from a marginal distribution indirectly, without having to calculate the density. This chapter illustrates the algorithm by exploring several examples. In this chapter, Gibbs sampling is based only on elementary properties of Markov chains.

Mathematically Mature: may contain mathematics beyond calculus with proofs.

16.1. Chapter Starter Question

What is Bayes' Rule for 2 jointly distributed continuous random variables X and Y?

16.2. General Theory of Gibbs Sampling

Given a joint density $f(x, y_1, \ldots, y_p)$, the goal is to find characteristics of the marginal density

$$f_X(x) = \int \cdots \int f(x, y_1, \ldots, y_p) \, \mathrm{d}y_1 \ldots \mathrm{d}y_p,$$

such as the mean or variance. Often the integrations are extremely difficult to do, either analytically or numerically. In such cases the Gibbs sampler provides an alternate method for obtaining $f_X(x)$.

Gibbs sampling effectively generates a sample $X_1, \ldots, X_m \sim f_X(x)$ without requiring $f_X(x)$. By simulating a large enough sample, the mean, variance, or any other characteristic of $f_X(x)$ can be calculated to the desired degree of accuracy.

First consider the two-variable case. Starting with a pair of random variables (X, Y), the Gibbs sampler generates a sample from $f(x)$ by sampling instead from the conditional distributions $f(x \mid y)$ and $f(y \mid x)$. The conditional distributions are often already known in statistical models. This is done by generating a *Gibbs sequence* of random variables

(16.1) $$Y_0', X_0', Y_1', X_1', Y_2', X_2', \ldots, Y_n', X_n'.$$

The first value $Y_0' = y_0'$ is specified and the rest of the sequence is obtained iteratively by alternately generating values from

$$X_j' \sim f(x \mid Y_j' = y_j')$$
$$Y_{j+1}' \sim f(y \mid X_j' = x_j').$$

The generation of the sequence is **Gibbs sampling**. Under reasonable general conditions, the distribution of X_n' converges to $f_X(x)$, the true marginal of X, as $n \to \infty$. Thus for n large enough, the last observation X_n' is effectively a sample point from $f_X(x)$.

16.3. Bivariate Binomial Example

This section discusses Gibbs sampling in detail for the simplest case of a 2×2 table. Suppose X and Y are each marginally Bernoulli random variables with joint distribution (the probabilities X are by rows, with $Y = 0$ the first row, $Y = 1$ the second row)

$$
\begin{array}{cc}
 & X = 0 \quad X = 1 \\
\begin{array}{c} Y = 0 \\ Y = 1 \end{array} & \left(\begin{array}{cc} p_1 & p_2 \\ p_3 & p_4 \end{array} \right)
\end{array}
$$

or in terms of the joint probability distribution

$$
\begin{pmatrix} f_{X,Y}(0,0) & f_{X,Y}(1,0) \\ f_{X,Y}(0,1) & f_{X,Y}(1,1) \end{pmatrix} = \begin{pmatrix} p_1 & p_2 \\ p_3 & p_4 \end{pmatrix}.
$$

The marginal distribution of X is

$$f_X = (f_X(0), f_X(1)) = (p_1 + p_3, p_2 + p_4),$$

a Bernoulli distribution with success probability $p_2 + p_4$. The conditional probabilities can be expressed in two matrices

$$
P_{Y|X} = \begin{pmatrix} \frac{p_1}{p_1 + p_3} & \frac{p_3}{p_1 + p_3} \\ \frac{p_2}{p_2 + p_4} & \frac{p_4}{p_2 + p_4} \end{pmatrix}
$$

and

$$
P_{X|Y} = \begin{pmatrix} \frac{p_1}{p_1 + p_2} & \frac{p_2}{p_1 + p_2} \\ \frac{p_3}{p_3 + p_4} & \frac{p_4}{p_3 + p_4} \end{pmatrix}
$$

where $P_{Y|X}$ has the conditional probabilities of Y given X and $P_{X|Y}$ has the conditional probabilities of X given Y.

As an example, to generate the marginal distribution of X uses the X' sequence from (16.1). From X_0' to X_1' goes through Y_0', so the Gibbs sequence is $X_0' \to Y_1' \to X_1'$. Thus $X_0' \to X_1'$ is a two-stage Markov chain, with transition probability

$$\mathbb{P}\left[X_1' \mid X_0'\right] = \sum_y \mathbb{P}\left[Y_1' = y \mid X_0'\right] \cdot \mathbb{P}\left[X_1' \mid Y_1' = y\right]$$

For example, to go from $X = 1$ to $X = 0$ takes the dot product of the second row of $P_{Y|X}$ with the first column of $P_{X|Y}$. Generally, $\mathbb{P}\left[X_1' \mid X_0'\right]$ is the right matrix multiplication of $P_{Y|X}$ by $P_{X|Y}$, so

$$P_{X|X} = P_{Y|X} P_{X|Y}$$

is the transition probability matrix for the X' sequence. Observe carefully the subscript notation, the Xs are "inside"; the Ys "outside". The Ys disappear as the variable of summation from the two-stage Markov chain. The matrix that gives $\mathbb{P}\left[X'_n = x_n \mid X'_0 = x_0\right]$ is $(P_{X|X})^n$. Letting $f_n = (f_n(0), f_n(1))$ denote the marginal probability distribution of X'_n then $f_n = f_0 P^n_{X|X} = f_0(P^{n-1}_{X|X})P_{X|X} = f_{n-1}P_{X|X}$. By the Fundamental Theorem for Markov Chains, $f_n \to \pi$ as $n \to \infty$, the stationary distribution π satisfying $\pi P_{X|X} = \pi$. If the Gibbs sequence converges, the π satisfying $\pi P_{x|x} = \pi$ must be the marginal distribution of X. In this small example, it is straightforward to check that $\pi = (p_1 + p_3, p_2 + p_4)$ satisfies $\pi P_{X|X} = \pi$. So stopping the iteration scheme at a large enough value of n gives random variables distributed approximately as π. The larger the value of n, the better the approximation. No general guidance on choosing such n is available. However, one possible approach is to monitor density estimates from m independent Gibbs sequences, and choosing the first point at which these densities agree to a satisfactory degree.

The algebra for the 2×2 case works for any $n \times m$ joint distribution of X's and Y's. Analogously define the $n \times n$ transition matrix $P_{X|X}$ whose stationary distribution will be the marginal distribution of X.

If either (or both) of X and Y are continuous, then with suitable assumptions, all of the theory still goes through, so the Gibbs sampler still produces a sample from the marginal distribution of X. The conditional density of X_1 given X_0 is $f_{X_1|X_0}(X_1 \mid X_0) = \int f_{X_1|Y_1}(x_1 \mid y)f_{Y_1|X_0}(y \mid x_0)\, \mathrm{d}y$. Then, step by step, write the conditional densities of $X'_2 \mid X'_0$, $X'_3 \mid X'_0$, $X'_4 \mid X'_0$, ... Similar to the n-step transition matrix $(P_{x|x})^n$, derive an "continuous transition probability matrix" with entries satisfying the relationship

$$f_{X'_n|X'_0}(x \mid x_0) = \int f_{X'_n|X'_{n-1}}(X \mid t)f_{X'_{n-1}|X'_0}(t \mid x_0)\, \mathrm{d}t,$$

the continuous version of the right matrix multiplication. As $n \to \infty$ it again follows that the stationary distribution is the marginal density of X, the density to which $f_{X'_n|X'_0}$, converges.

16.4. Gibbs Sampling from the Bivariate Normal

Recall that the probability density function for the bivariate normal distribution with means 0 and variance 1 for the variables (for simplicity) and correlation ρ between the two variables is
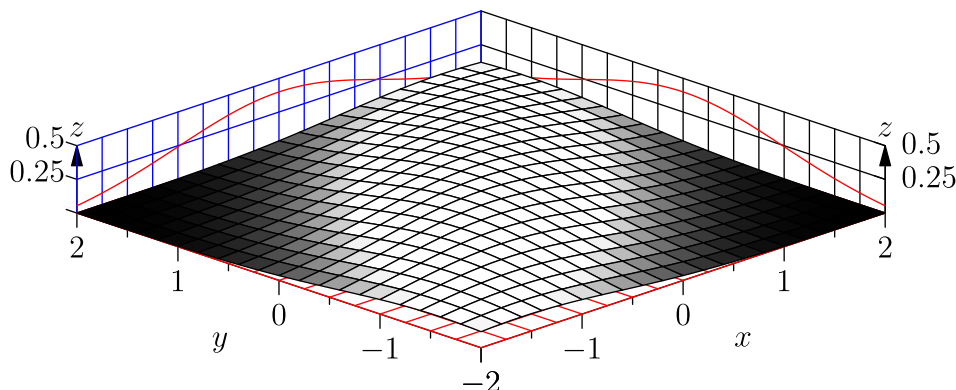
$$f(x, y) = \frac{1}{2\pi\sqrt{1 - \rho^2}} \mathrm{e}^{-\frac{x^2 - 2\rho x y + y^2}{2(1 - \rho^2)}}.$$

A standard exercise is to show that the marginal densities are in fact

$$f_x(x) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-x^2/2}$$

and

$$f_y(y) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-y^2/2}.$$

Figure 1. Bivariate normal probability density with marginals.

Another standard exercise is to show the conditional probability densities are

$$f_{x|y}(x) = \frac{1}{\sqrt{2\pi(1-\rho^2)}} e^{-(x-\rho y)^2/(2(1-\rho^2))}$$

and

$$f_{y|x}(y) = \frac{1}{\sqrt{2\pi(1-\rho^2)}} e^{-(y-\rho x)^2/(2(1-\rho^2))}.$$

That is, $X_i \mid X_j \sim N(x_j, (1-\rho^2))$. Figure 1 illustrates the bivariate normal density with $\rho = 0.8$. The scaling of the z-axis is approximately equal to the x and y axes, so that the density appears less "hill-like" than is often illustrated. The marginal densities are projected onto coordinate planes at $x = 2$ and $y = 2$.

Because the marginals are known, using the Gibbs sampler is not necessary to simulate them. However, Gibbs sampling using this elementary example is illustrative as in Figure 2. Apply the algorithm for 1000 steps. To allow for convergence to the stationary distribution, the first 500 steps are discarded, using only the last 500 steps. The first two subgraphs show the frequencies of the sampled marginal distributions, along with the theoretical densities in red. The correspondence appears close. Better than a visual comparison of the densities is to use a Q-Q plot comparing the sample quantiles to the quantiles of the standard normal distribution. The straight line correspondence of the quantiles over several standard deviations demonstrates the excellent match of simulation to the theoretical density. The third row shows the scatter of the bivariate samples, with the characteristic elliptical distribution The second figure in the third row shows the autocorrelation of the sample points by connecting successive points with segments. The last two plots illustrate the "white noise"-like aspect of the marginal distributions, as expected. Altogether, these reinforce the intuition that the Gibbs sample has produced a satisfactory sample of the marginal distributions.

16.5. A Gibbs Sampler for a Spam Filter

Consider a Bayesian Inference model for an email spam filter. Some algorithm examines each email and classifies the email as *spam* or *not-spam*. The goal is
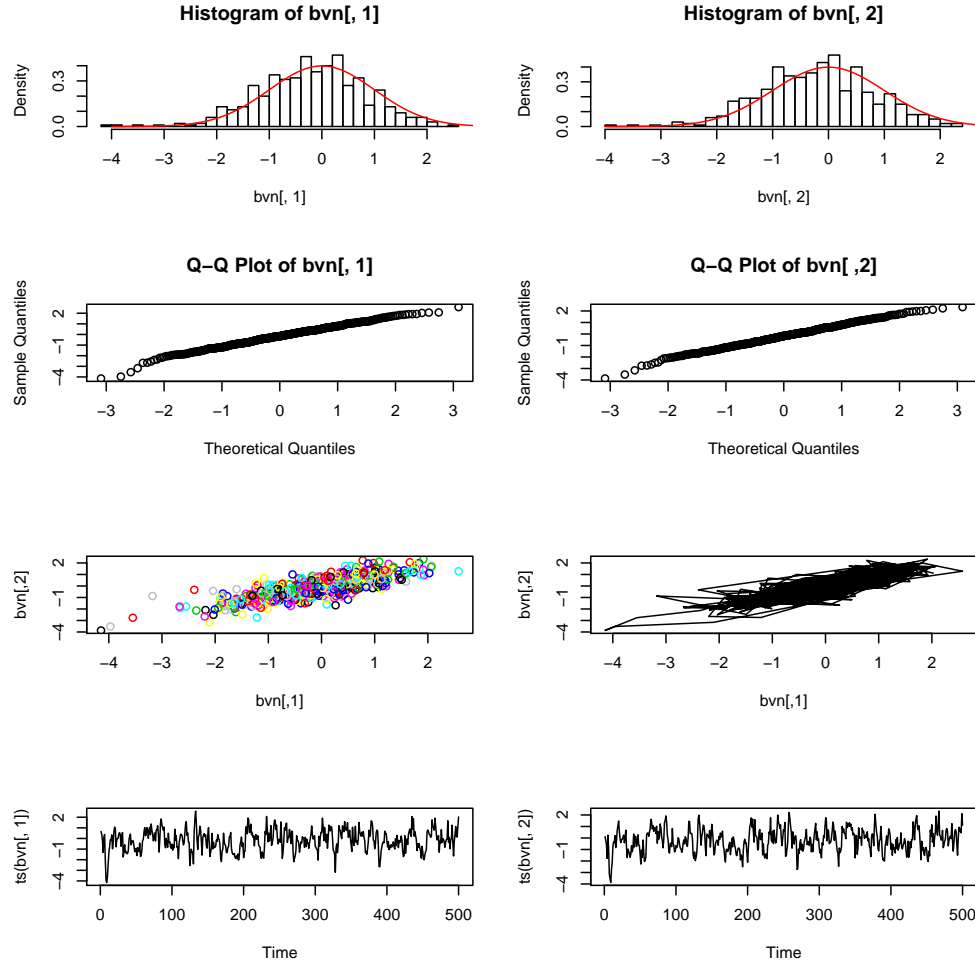
Figure 2. Outcomes from Gibbs sampling of a bivariate normal distribution,
illustrating the quality of the simulation.

to estimate the prevalence ψ of email spam without having to directly examine
emails and count the instances of email spam. The model has the following random
variables and events:

- $[S = 1]$, the event that the email is in fact spam, a positive outcome occurring
  with probability $\mathbb{P}[S = 1] = \psi$,

- $[S = 0]$, the event that the email is *not* spam, a negative outcome with
  $\mathbb{P}[S = 0] = 1 - \psi$,

- $[R = 1]$, the event that the algorithm correctly marks spam, a *true positive*,

- $[R = 0]$, the event that the algorithm *incorrectly* marks spam, a *false negative*.

Then define the following parameters, known or set for the algorithm:

- the *sensitivity* of the algorithm is $\eta = \mathbb{P}[R = 1 \mid S = 1] = 0.90$ (so the false positive rate is $\mathbb{P}[R = 0 \mid S = 1] = 0.10$),
- *specificity* of the algorithm: $\theta = \mathbb{P}[R = 0 \mid S = 0] = 0.95$ (so the false negative rate is $\mathbb{P}[R = 1 \mid S = 0] = 0.05$),
- *prevalence* $\psi = \mathbb{P}[S = 1]$.

Then

$$\tau = \mathbb{P}[R = 1] = \mathbb{P}[R = 1 \mid S = 1]\,\mathbb{P}[S = 1] + \mathbb{P}[R = 1 \mid S = 0]\,\mathbb{P}[S = 0]$$
$$= \psi\eta + (1 - \psi)(1 - \theta).$$

Rewrite the equation to solve for the prevalence ψ, usually the parameter of interest:

$$\psi = \frac{\tau + \theta - 1}{\eta + \theta - 1}.$$

Assume as an example, that the algorithm marks $r = 233$ emails as spam out of a total of $n = 1000$ emails. Then the empirical rate of rejection $\hat{\tau} = r/n = 0.233$. From the relation for the prevalence ψ in terms of the other parameters the empirical estimate of prevalence is $\hat{\psi} = 0.215$. Since this is a Bernoulli random variable, the traditional frequentist confidence interval is $\tau \pm 1.96\sqrt{\tau(1 - \tau)/n}$ which becomes $[0.207, 0.259]$. Using these in the relation for the prevalence ψ gives an interval $\psi \in [0.184, 0.246]$.

Unfortunately, if r is relatively small compared to n, the relation can give nonsensical estimates. For example, $\eta = 0.99$, $\theta = 0.97$, $r = 5$ from $n = 250$ gives $\tau = 0.02$ but $\psi = -0.0104$.

Before starting the Gibbs sampling, first use simple Bayesian inference starting from Bayes' Formula

$$\text{Posterior} \propto \text{Prior} \cdot \text{Likelihood}$$

or in probability notation

$$\mathbb{P}[\tau \mid r] \propto \mathbb{P}[\tau] \cdot \mathbb{P}[r \mid \tau].$$

Start with a noninformative prior where τ is uniformly distributed on $(0, 1)$, but use a Beta density

$$\mathbb{P}[\tau = x] = \frac{1}{B(\alpha, \beta)} x^{\alpha - 1}(1 - x)^{\beta - 1}$$

with initial shape parameters $\alpha = 1$ and $\beta = 1$ so the density is uniform. Recall that $\mathbb{P}[r \mid \tau]$ is binomial, so

$$\mathbb{P}[\tau \mid r] \propto \left(\tau^{\alpha_0 - 1}(1 - \tau)^{\beta_0 - 1}\right) \cdot \left(\tau^r(1 - \tau)^{n - r}\right)$$
$$\propto \tau^{\alpha_0 + r - 1}(1 - \tau)^{\beta_0 + n - r - 1}.$$

The right side is again the form of the Beta distribution. This correspondence between the Beta families for the prior and the posterior and the Binomials for the likelihood makes them a **conjugate pair**. The mean of the Beta distribution with shape parameters α and β is $\alpha/(\alpha + \beta)$ and the variance is $\alpha\beta/((\alpha + \beta)^2(\alpha + \beta + 1))$.

For the spam classification problem with the previous data

$$\alpha = \alpha_0 + r = 1 + 233 = 234$$
$$\beta = \beta_0 + n - r = 1 + 1000 - 233 = 768.$$

This allows using the resulting Beta density posterior for $p(\tau \mid r)$ to derive the mean and confidence interval for τ. Using the facts about the mean and variance above, the posterior density mean is $\alpha/(\alpha + \beta) = 0.234$. The 95% confidence interval using the Beta c.d.f defined by $\text{Beta}(\tau_-, \alpha, \beta) = 0.025$ and $\text{Beta}(\tau_+, \alpha, \beta) = 0.975$ is $(0.208, 0.260)$. This is approximately the same as the frequentist confidence interval.

Next use a Gibbs sampler to get point and interval estimates for ψ. Unlike the estimate for τ where the count r of the spam-marked emails is available, a Gibbs sampler is useful because it is not feasible to directly count the spam emails. Again using Bayes' Formula

$$p = \mathbb{P}\left[S = 1 \mid R = 1\right] = \mathbb{P}\left[S = 1, R = 1\right]/\mathbb{P}\left[R = 1\right] =$$
$$\mathbb{P}\left[R = 1 \mid S = 1\right]\mathbb{P}\left[S = 1\right]/\mathbb{P}\left[R = 1\right] = \psi\eta/\tau$$

and

$$1 - p = \mathbb{P}\left[S = 1 \mid R = 0\right] = \psi(1 - \eta)/(1 - \tau).$$

Simulate the *latent counts*

$$X \mid (r, \psi) \sim \text{Binom}(r, p)$$
$$Y \mid (r, \psi) \sim \text{Binom}(n - r, 1 - p)$$
$$\psi \mid (X, Y) \sim \text{Beta}(\alpha_n, \beta_n)$$

where $V = X + Y$, $\alpha_n = \alpha_0 + V$ and $\beta_n = n - V$. Use one value of ψ to find the next in the typical Gibbs iteration. This is in effect a Markov chain for which the limiting distribution is the posterior of ψ. At each iteration, use the prior distribution and the data. The continued and recursive use of this information will cause the convergence of the simulated values toward the appropriate posterior distribution. Figure 3 illustrates the results of this Gibbs Sampler method.

16.6. Gibbs Sampling in Statistics

Suppose $f(x_1, x_2, \ldots, x_N)$ is a joint probability density in which the variables represent parameters of a statistical model. The goal is to get point and interval estimates for these parameters. To fit this into the Gibbs sampling framework, assume that all the single-variable conditional probability densities

$$f(x_i \mid x_j, j \neq i)$$

are available, that is, are a type for which samples can be obtained using standard algorithms. Examples of available densities include the uniform, the normal, the gamma, the Poisson, and any finite distribution. To generate a sequence of samples, select $\mathbf{x}^0 = (x_1^0, x_2^0, \ldots, x_N^0)$ arbitrarily and then create $\mathbf{x}^1 = (x_1^1, x_2^1, \ldots, x_N^1)$ as follows:

(1) Generate a sample x_1^1 from $f(x_1 \mid x_2^0, x_3^0, \ldots x_N^0)$.
(2) Generate a sample x_2^1 from $f(x_2 \mid x_1^1, x_3^0, x_4^0 \ldots x_N^0)$.

Histogram of psi
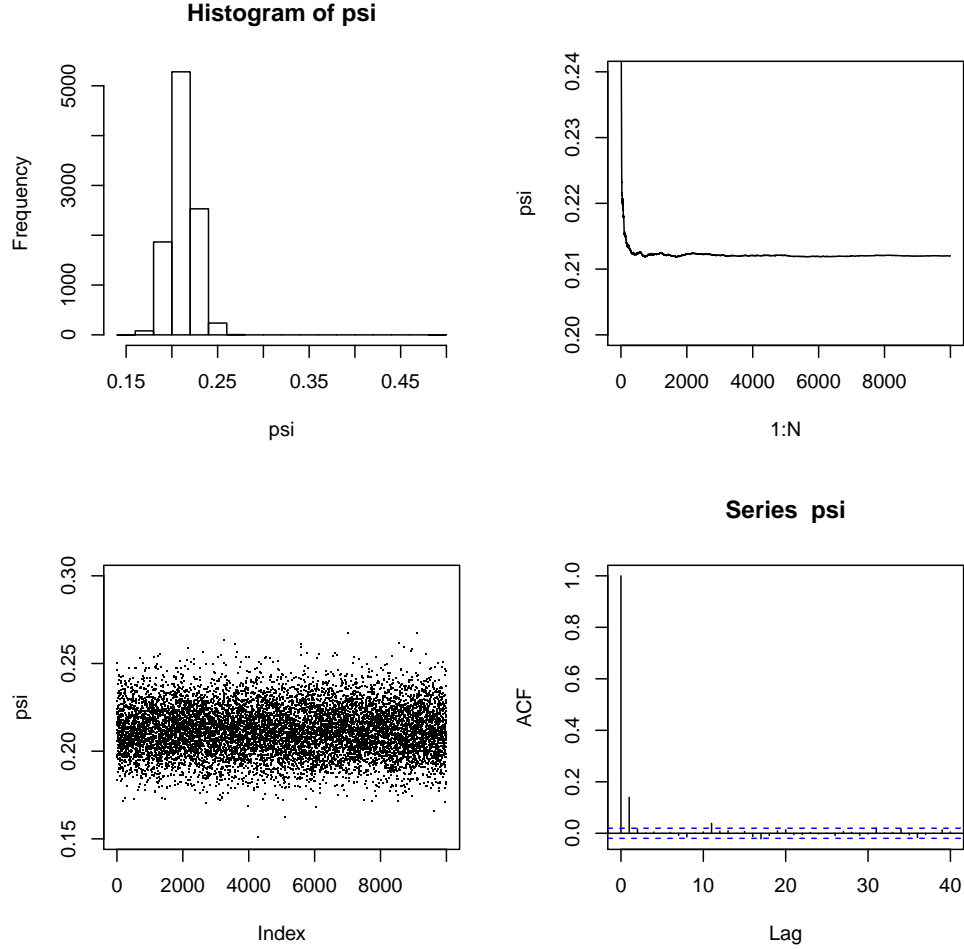
Figure 3. Results of the Gibbs sampler applied to the spam filter.

(3) Generate a sample x_3^1 from $f(x_3 \mid x_1^1, x_2^1, x_4^0, \ldots x_N^0)$.

(4) ...

N. Generate a sample x_N^1 from $f(x_N \mid x_1^1, x_2^1, \ldots x_{N-1}^1)$.

One cycle, similar to a sweep of the lattice in the Ising model or a raster scan of an image, produces a new value $\mathbf{x}^1$. Repeating this process M times produces

$$\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^M$$

which approximates a sample from the probability density $f(x_1, x_2, \ldots, x_N)$.

Using this sample, almost any property of the probability density can be investigated. For example, focusing on only the first component of each $\mathbf{x}^k$ produces a

sample

$$x_1^0, x_1^1, x_1^2, \ldots, x_1^M$$

from the marginal probability distribution of the first component, formally given by the integral

$$f(x_1) = \int_{x_2} \cdots \int_{x_N} f(x_1, x_2, \ldots, x_N) \, \mathrm{d}x_N \ldots \mathrm{d}x_2 .$$

In this way, Gibbs sampling can be thought of as a multi-dimensional numerical integration algorithm. The expected value of the first component x_1,

$$\mathbb{E}\left[x_1\right] = \int_{x_1} x_1 f(x_1) \, \mathrm{d}x_1$$

is estimated by the arithmetic mean of the sample $x_1^0, x_1^1, x_1^2, \ldots, x_1^M$. A 95% confidence interval for x_1 can be taken directly from the sample.

16.7. Gibbs Sampling for Normal Parameters

As another example of the Gibbs sampler, suppose that $X \sim N(\mu, 1/\tau)$ with μ and τ unknown. Based on a reasonably sized sample, the goal is to get the posterior distributions of μ and τ using the Gibbs sampler. Here μ is the population mean, and τ, called the *population precision*, is the reciprocal of the variance, n is the sample size, $\bar{x}$ is the sample mean and $s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$ is the sample variance. Then make a sequence of iterations $i = 1, \ldots N$; with a sample $\mu^{(i)}$ from $f(\mu \mid \tau^{(i-1)}, \text{data})$ (see below for the definition) and sample $\tau^{(i)}$ from $f(\tau \mid \mu^{(i-1)}, \text{data})$ (see below for the definition). Then the theory behind Gibbs sampling ensures that after a sufficiently many iterations, T, the set $\left\{(\mu^{(i)}, \tau^{(i)}) : i = T+1, \ldots, N\right\}$ can be seen as a random sample from the joint posterior distribution. The priors are that $f(\mu, \tau) = f_\mu(\mu) \cdot f_\tau(\tau)$ with $f_\mu(\mu) \propto 1$ and $f_\tau(\tau) \propto 1/\tau$. With these definitions and assumptions, standard theory shows the sample mean from a normal distribution given the population precision has a conditional posterior distribution

$$(\mu \mid \tau, \text{data}) \sim N\left(\bar{x}, \frac{1}{n\tau}\right).$$

Next, the claim is that the precision (the reciprocal of the variance), given the mean, has a conditional posterior distribution

$$(\tau \mid \mu, \text{data}) \sim \text{Gamma}\left(\frac{n}{2}, \frac{2}{(n-1)s^2 + n(\mu - \bar{x})^2}\right).$$

The derivation of the claim is:

$$\text{Posterior} \propto \text{Prior} \cdot \text{Likelihood}$$

or in probability notation

$$\mathbb{P}\left[\tau \mid \mu, \text{data}\right] \propto \mathbb{P}\left[\tau\right] \cdot \mathbb{P}\left[\mu, \text{data} \mid \tau\right].$$

Then using $\frac{1}{\sigma^n} = \tau^{n/2}$

$$\mathbb{P}\left[\tau \mid \mu, \text{data}\right] \propto \tau^{-1} \cdot \tau^{n/2} \exp\left(-\frac{\tau}{2} \sum_{\nu=1}^{n} (x_\nu - \bar{x})^2\right)$$

$$\propto \tau^{-1} \cdot \tau^{n/2} \exp\left(-\frac{\tau}{2} \sum_{\nu=1}^{n} (x_\nu - \mu + \mu - \bar{x})^2\right)$$

$$\propto \tau^{-1} \cdot \tau^{n/2} \exp\left(-\frac{\tau}{2} \left[\sum_{\nu=1}^{n} (x_\nu - \mu)^2\right.\right.$$

$$\left.\left. + 2\sum_{\nu=1}^{n} (x_\nu - \mu)(\mu - \bar{x}) + \sum_{\nu=1}^{n} (\mu - \bar{x})^2\right]\right)$$

$$\propto \tau^{-1} \cdot \tau^{n/2} \exp\left(-\frac{\tau}{2} \left[\sum_{\nu=1}^{n} (x_\nu - \mu)^2 + \sum_{\nu=1}^{n} (\mu - \bar{x})^2\right]\right)$$

$$\propto \tau^{n/2-1} \cdot \exp\left(-\frac{\tau}{2} \left[(n-1)s^2 + n(\mu - \bar{x})^2\right]\right).$$

The last expression is now recognizable as a Gamma distribution with shape parameter $\frac{n}{2}$ and rate (inverse scale parameter) $(n-1)s^2 + n(\mu - \bar{x})^2$. Having this distribution is convenient for implementing the Gibbs sampler. (As a side remark, in some derivations for Bayesian statistics, the prior for τ is taken to be Gamma(α, β) which leads to a similar looking Gamma posterior including the additional parameters α and β, but for Gibbs sampling, this is unnecessary because the theory shows that the sampling will converge even with the simpler choice.)

The R script below implements the Gibbs sampler for these parameters, with $n = 30$, a typical value, and postulating $\bar{x} = 15$, and $s^2 = 3$. The number of iterations is $N = 11{,}000$ with a transient period of 1000 iterations. Figure 4 shows the results of the of Gibbs sampler for the parameters of a normal distribution.

16.8. Bayesian Hierarchical Models

Apply Gibbs sampling formalism to a **Bayesian hierarchical model**. Hierarchical Bayesian models naturally describe the connections among data, observed parameters and other unobserved parameters, sometimes called **latent variables**. A simple three-level hierarchical model uses Bayes' rule to bind together data, X; a parameter to be estimated, λ; and an additional hyper-parameter, β. Both λ and β can be vectors. These are connected in the following way:

(1) At the first level, X is described by its likelihood function $f(X \mid \lambda)$, the probability of observing X conditioned on λ.

(2) At the next level, λ is modeled by a probability density function, $g(\lambda \mid \beta)$, conditioned on the parameter β.

(3) At the third level, the hyper-parameter β is modeled with another density function $h(\beta)$. The choice of $h(\beta)$ reflects the modeler's prior beliefs about the likely values of β.

Histogram of mu

Histogram of tau
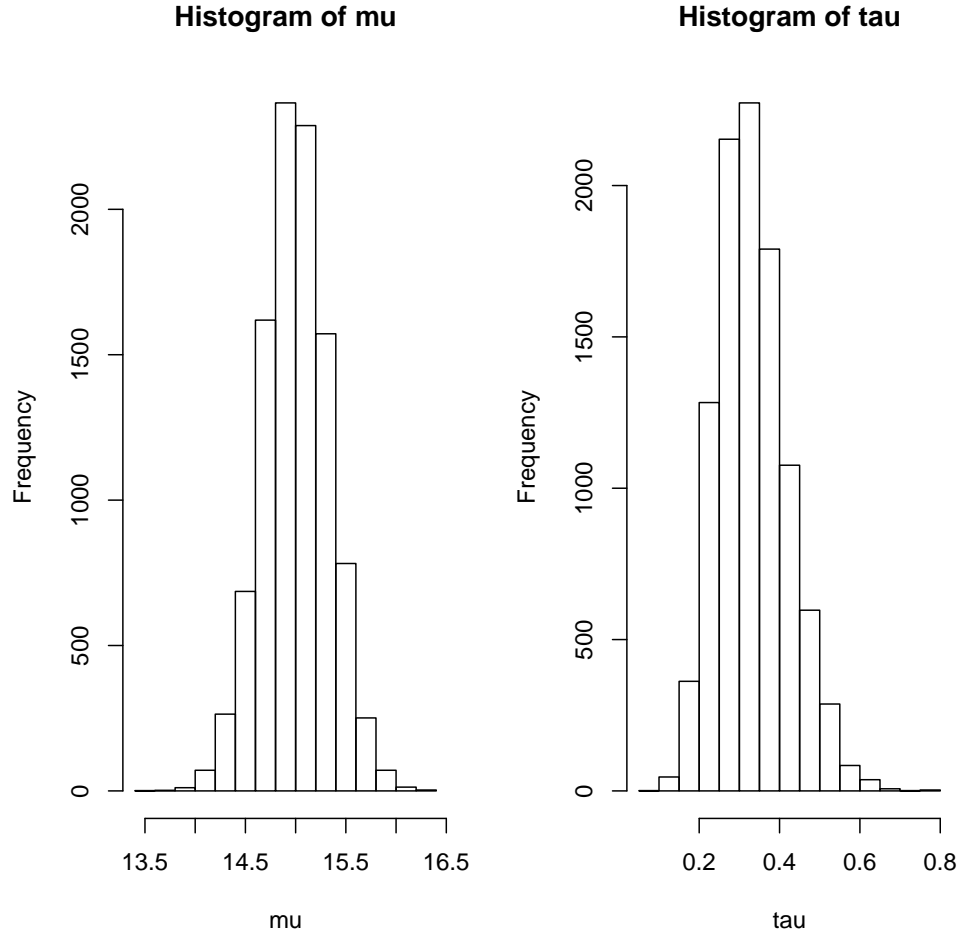
Figure 4. Results of the Gibbs sampler for the parameters of a normal distribution.

The three density functions are combined with Bayes' rule, producing a probability density function for λ and β conditioned on the data X

$$F(\lambda, \beta \mid X) \propto f(X \mid \lambda)g(\lambda \mid \beta)h(\beta).$$

The constant of proportionality is the reciprocal of

$$\int_\lambda \int_\beta f(X \mid \lambda)g(\lambda \mid \beta)h(\beta) \, \mathrm{d}\beta \, \mathrm{d}\lambda$$

which is independent of the parameters λ and β, though dependent on the data X. The integrals, (or sums, in the case of discrete distributions) are over all values λ and β. In most cases, the integral or sum is impossible to evaluate. However, as explained before, the Gibbs sampling algorithm avoids this expression.

As a specific example, consider a model of water pump failure rates. The data, X, are given by pairs (s_i, t_i) for $i = 1, 2, \ldots, 10$. Each pair represents failure information for an individual pump. For each pump, assume the number of failures s_i in time t_i is given by a Poisson distribution with parameter $\lambda_i t_i$, that is

$$f_i(s_i \mid \lambda_i) = \frac{(\lambda_i t_i)^{s_i} \mathrm{e}^{-\lambda_i t_i}}{s_i!}, \quad i = 1, 2, \ldots, 10.$$

Assuming the failures occur independently, the likelihood function for $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{10})$ is the product:

$$f(X \mid \lambda) = \prod_{i=1}^{10} \frac{(\lambda_i t_i)^{s_i} \mathrm{e}^{-\lambda_i t_i}}{s_i!}.$$

The traditional frequentist approach is to use $\bar{\lambda}_i = s_i / t_i$ as the point estimate of λ_i for $i = 1, 2, \ldots, 10$. The Bayesian approach is to assume that the individual λ_i's are linked together by a common distribution. A natural choice is a gamma distribution with shape parameter α and scale parameter β, so that the density for the ith parameter is

$$g_i(\lambda_i \mid \alpha, \beta) = \frac{\lambda_i^{\alpha-1} \mathrm{e}^{-\lambda_i / \beta}}{\beta^\alpha \Gamma(\alpha)}.$$

The remaining hyper-parameter β is described by an inverse gamma distribution with shape parameter γ and scale parameter δ, so that

$$h(\beta) = \frac{\delta^\gamma \mathrm{e}^{-\delta/\beta}}{\beta^{\gamma+1} \Gamma(\gamma)}.$$

The parameters γ and δ are selected to make the top-level inverse gamma reasonably *diffuse*. A diffuse distribution tries to convey as little prior information as possible about the parameters. An extreme case of a noninformative distribution is the uniform distribution on the parameter space.

The resulting posterior joint density for the parameters $\lambda_1, \lambda_2, \ldots, \lambda_{10}$ along with the scale parameter β is

$$F(\lambda_1, \lambda_2, \ldots, \lambda_{10}, \beta \mid X) \propto$$
$$\left[\prod_{i=1}^{10} \frac{(\lambda_i t_i)^{s_i} \mathrm{e}^{-\lambda_i t_i}}{s_i!} \right] \left[\prod_{i=1}^{10} \frac{\lambda_i^{\alpha-1} \mathrm{e}^{-\lambda_i / \beta}}{\beta^\alpha \Gamma(\alpha)} \right] \left[\frac{\delta^\gamma \mathrm{e}^{-\delta/\beta}}{\beta^{\gamma+1} \Gamma(\gamma)} \right].$$

For $i = 1, 2, \ldots, 10$, the density for λ_i conditioned on the other parameters is proportional to

$$\lambda_i^{s_i + \alpha - 1} \mathrm{e}^{-\lambda_i (t_i + 1/\beta)}.$$

The constant of proportionality is obtained by absorbing all factors independent of λ_i. The form of the density for λ_i shows that $\mathbb{P}[\lambda_i \mid \lambda_j, j \neq i, X, \beta]$ is a gamma distribution with shape parameter $s_i + \alpha$ and scale parameter $1/(t_i + 1/\beta)$. Since the gamma distribution is available, Gibbs sampling can be applied at this step. The density for β, conditioned on the other parameters, is proportional to

$$\frac{\mathrm{e}^{-(\sum_{i=1}^{10} \lambda_i + \delta)/\beta}}{\beta^{10\alpha + \gamma + 1}}$$

showing that $\mathbb{P}\left[\beta \mid \lambda_1, \lambda_2, \ldots, \lambda_{10}, X\right]$ is an inverse gamma distribution with shape parameter $\gamma + 10\alpha$ and $\sum_{i=1}^{10} \lambda_i + \delta$. This too is an available distribution.

This model is an example of a *conjugate hierarchical model*. In a conjugate hierarchical model the intermediate distributions, in this case for λ_i and β, are similar to the original distributions in the hierarchical model. This fits with the Gibbs sampling requirement that these distributions be available.

16.9. Gibbs Sampling for Digital Images

A simple model of a digital image consists of pixel elements arranged on a rectangular lattice with N sites. Each pixel takes a value from a set $S = \{1, 2, \ldots, K\}$, indicating grayscale or color. For simplicity and analogy to the Ising model, take a black and white image with pixel levels $-1, +1$. An image is a configuration $\omega \in \Omega$ assigning a level to each of the N sites. Even modestly sized images result in immensely large configuration spaces. For a 100×100 black and white image, $\text{card}\,(\Omega) = 2^{10,000}$.

Consider a model for image degradation with additive noise, modeled by N independent identically distributed random variables $\mathcal{N} = \{\eta_1, \eta_2, \ldots \eta_N\}$. Specifically, take noise with the η_i normally distributed with mean 0 and variance σ^2, that is $\eta_i \sim N(0, \sigma^2)$. Letting ω^{blurred} indicate the degraded or blurred image, $\omega^{\text{blurred}} = \omega + \mathcal{N}$. Since the values of ω^{blurred} are real numbers, the resulting image is determined by rounding each value to the nearest value in S.

The relationship between the original image and the degraded version is probabilistic, given any image ω, there is some probability a particular ω^{blurred} is the degraded version of ω. Image reconstruction looks at the problem the other way around; given ω^{blurred}, there is some probability ω is the original image. This leads to an application of Bayes' Rule. The *posterior distribution* for ω conditioned on ω^{blurred} is

$$\mathbb{P}\left[\omega \mid \omega^{\text{blurred}}\right] = \frac{\mathbb{P}\left[\omega^{\text{blurred}} \mid \omega\right] \mathbb{P}\left[\omega\right]}{\mathbb{P}\left[\omega^{\text{blurred}}\right]}.$$

The goal is to find the configuration maximizing $\mathbb{P}\left[\omega \mid \omega^{\text{blurred}}\right]$, called the **maximum a posteriori estimate**. The technique is to formulate a new version of the Metropolis algorithm with Gibbs sampling.

By the Law of Total Probability the denominator is

$$\mathbb{P}\left[\omega^{\text{blurred}}\right] = \int_{\omega \in \Omega} \mathbb{P}\left[\omega^{\text{blurred}} \mid \omega\right] \mathbb{P}\left[\omega\right] \, d\omega.$$

This integral (or sum) is over all $\omega \in \Omega$ and does not depend on ω. This is reminiscent of the partition function and recalling the Metropolis algorithm just ignore it.

The likelihood function $\mathbb{P}\left[\omega^{\text{blurred}} \mid \omega\right]$ is

$$\mathbb{P}\left[\omega^{\text{blurred}} \mid \omega\right] \propto \prod_{i=1}^{N} e^{-\frac{(\omega_i^{\text{blurred}} - \omega_i)^2}{2\sigma^2}} = e^{-\frac{1}{2\sigma^2} \sum_{i=1}^{N} (\omega_i^{\text{blurred}} - \omega_i)^2}$$

where any constant of proportionality will be absorbed into the denominator above.

An image has patterns which are contiguous regions of similar pixel values. This is reminiscent of the Ising model, where after magnetization the lattice has long-range correlation between pixels, that is, image-like features. On the other hand, if neighboring values reminiscent of the Ising model at high temperatures.

This suggests using the Boltzmann probability distribution with the Ising potential energy function as the prior distribution on images, that is

$$\mathbb{P}\left[\omega\right] \propto e^{-E_{\text{ising}}(\omega)/(k_{\text{Boltz}}T)}$$

where

$$E_{\text{ising}}(\omega) = -J \sum_{i=1}^{N} \sum_{k=1}^{4} \omega_i \omega_{\langle i \rangle [k]}$$

is the nearest neighbor affinity. To keep the idea of correlated pixel values, let $k_{\text{Boltz}}T/J = 1 < T_c$, the critical temperature below which a phase transition occurs.

Putting all the parts together, the posterior distribution, $\mathbb{P}\left[\omega \mid \omega^{\text{blurred}}\right]$ is

$$\mathbb{P}\left[\omega \mid \omega^{\text{blurred}}\right] \propto e^{-\frac{1}{2\sigma^2} \sum_{i=1}^{N} (\omega_i^{\text{blurred}} - \omega_i)^2} \cdot e^{-E_{\text{ising}}}$$

$$\propto e^{-\left[\frac{1}{2\sigma^2} \sum_{i=1}^{N} (\omega_i^{\text{blurred}} - \omega_i)^2 + E_{\text{ising}}\right]}.$$

Viewing this from a statistical mechanics perspective leads to an analog of an energy function

$$E_{\text{image}}(\omega \mid \omega^{\text{blurred}}) = \frac{1}{2\sigma^2} \sum_{i=1}^{N} (\omega_i^{\text{blurred}} - \omega_i)^2 + E_{\text{ising}}$$

$$= \frac{1}{2\sigma^2} \sum_{i=1}^{N} (\omega_i^{\text{blurred}} - \omega_i)^2 - \sum_{\langle i \rangle [j]} \omega_i \omega_j.$$

where $\langle i \rangle$ indicates the set of nearest neighbor sites j for site i.

Finding the most probable original image ω given ω^{blurred} is equivalent to minimizing $E_{\text{image}}(\omega \mid \omega^{\text{blurred}})$. The first term is a positive potential energy penalty for straying too far from the data ω^{blurred} while the second term represents a negative potential energy reflecting the desire to align neighboring pixel values. The optimal solution balances these two conflicting objectives.

Creating an energy function to minimize leads to a contrast in methods. The Ising model starts with an objective function and interprets it as a energy function, using this to convert it to a probability from a physical interpretation. The image reconstruction model starts with a probabilistic situation with a Bayesian structure, leading to an energy function.

To implement Gibbs sampling, the probability of ω_i conditioned on *all* the other sites depends on *only* the sites in the nearest neighborhood set. Suppressing the dependence on ω^{blurred}, this means

$$\mathbb{P}\left[\omega_i \mid \omega_j, j \neq i\right] = \mathbb{P}\left[\omega_i \mid \omega_j, j \in \langle i \rangle\right]$$

$$\propto e^{-E_i(\omega_i \mid \omega_j, j \in \langle i \rangle)}$$

where

$$E_i(\omega_i \mid \omega_j, j \in \langle i \rangle) = \frac{1}{2\sigma^2} (\omega_i^{\text{blurred}} - \omega_i)^2 - \sum_{\langle i \rangle [j]} \omega_i \omega_j.$$

A probability distribution whose conditional probabilities depend on only the values in a neighborhood system is called a **Gibbs distribution** and is an example of a larger notion called a *Markov random field*.

A standard way to implement Gibbs sampling for images is to use a sequence of raster scans, in order by rows or columns, guaranteeing all sites are visited many times. At a selected site i, select $\omega_i = k$ with probability

$$\mathbb{P}\left[\omega_i = \ell\right] \propto e^{-\frac{1}{2\sigma^2}(\omega_i^{\text{blurred}} - \ell)^2 - \sum_{\langle i \rangle [j]} \ell \omega_j}.$$

Repeating this with many raster scans results in a sequence of images that approximates a sample from the posterior distribution. Note the connection to the previous simple Gibbs sampling algorithm, generating a sample from $f(x)$ by sampling iteratively from the conditional distributions, except that now there are many more conditional distributions, one for each pixel.

Gibbs sampling fits into the Hastings generalization of the Metropolis algorithm in the following way: In the second step of the Metropolis-Hastings algorithm, the probabilities α_{ij} are all equal to 1. However, the transitions are no longer time-independent, since each depends on the site choice. As a result, the proof is somewhat more involved than the original proofs of convergence given by Hastings. Gibbs sampling will produce a sequence representing a sample from $\mathbb{P}\left[\omega \mid \omega^{\text{blurred}}\right]$. The full algorithm also includes a "temperature" parameter T to create a simple form of simulated annealing.

Theorem 16.1. *Assume*

(1) an image with N pixels,

(2) T_k is a any decreasing sequence of temperatures such that
    (a) $T_k \to 0$ as $k \to \infty$,
    (b) $T_k \geq N\Delta / \ln(k)$ for all sufficiently large k and constant Δ.

Then starting at $\omega^{(0)} = \omega^{blurred}$, the Gibbs sampling sequence $\omega^{(k)}$ for $k = 0, 1, 2, \ldots$ converges in distribution to the distribution which is uniform on the minimum vales of $E_{image}(\omega)$ and 0 otherwise.

In other words, following a prescribed annealing schedule, Gibbs sampling must, in theory, produce a maximum a posteriori estimate of $\mathbb{P}\left[\omega \mid \omega^{\text{blurred}}\right]$.

Even though this result guarantees convergence to the most likely image, the rate of convergence is slow. Theoretically, for a 100×100 lattice with $N = 10^4$ pixels, using the theorem requires $e^{20,000}$ steps to go from $T = 4$ to $T = 0.5$. In practice, it takes about $300 - 1000$ raster scans to produce acceptable results.

For a black and white ($k = -1$ or 1) image Gibbs sampling with annealing is especially straightforward to implement using the ideas of the section on Gibbs sampling. At the pixel ω_i define

$$E^k = \frac{1}{2\sigma^2}(k - \omega_i^{\text{blurred}})^2 + k \sum_{\langle i \rangle [j]} \omega_j.$$

Set $\omega_i = k$ with probability

$$\frac{e^{-E^k / k_{\text{Boltz}} T}}{e^{-E^{-1}/(k_{\text{Boltz}} T)} + e^{-E^1/(k_{\text{Boltz}} T)}}.$$

16.10. Chapter Ending Answer

For two continuous random variables X and Y, Bayes' rule can be stated in terms of the conditional and marginal densities,

$$f_{X|Y=y}(x) = \frac{f_{X,Y}(x,y)}{f_Y(y)},$$

$$f_{Y|X=x}(y) = \frac{f_{X,Y}(x,y)}{f_X(x)}.$$

Therefore

$$f_{X|Y=y}(x) = \frac{f_{Y|X=x}(y)f_X(x)}{f_Y(y)}.$$

16.11. Chapter Summary

Key Concepts.

(1) The Gibbs sampler is an algorithm for generating random variables from a marginal distribution indirectly, without having to calculate the density. Gibbs sampling is based only on elementary properties of Markov chains.

(2) The simple case of a 2×2 table with multinomial sampling clearly illustrates the Markov chain nature of the process.

(3) The simple case of a bivariate normal density with Gibbs sampling clearly illustrates Gibbs sampling for continuous distributions.

(4) A simple Bayesian model for a spam filter illustrates typical Gibbs sampling.

(5) Suppose $f(x_1, x_2, \ldots, x_N)$ is a probability distribution in which the variables represent parameters of a statistical model. Gibbs sampling obtains point and interval estimates for these parameters.

(6) Suppose $X \sim N(\mu, 1/\tau)$ with μ and τ unknown. Based on a reasonably sized sample, Gibbs sampling obtains the posterior distributions of μ and τ.

(7) Hierarchical Bayesian models naturally describe the connections among data, observed parameters and other unobserved parameters. A simple three-level hierarchical model uses Bayes' rule to bind together data, X, a parameter to be estimated, λ, and an additional hyper-parameter, β.

(8) In image degradation, a version of the Metropolis algorithm called Gibbs sampling the configuration maximizes $\mathbb{P}\left[\omega \mid \omega^{\text{blurred}}\right]$, called the maximum a posteriori estimate.

Vocabulary.

(1) The **Gibbs sampler** is a technique for generating random variables from a marginal distribution indirectly, without having to calculate the density.

(2) **Gibbs sampling** obtains a Gibbs sequence iteratively by alternately generating values from

$$X'_j \sim f(x \mid Y'_j = y'_j)$$
$$Y'_{j+1} \sim f(y \mid X'_j = x'_j).$$

(3) The functional correspondence between the Beta families for the prior and the posterior and the Binomials for the likelihood makes them a **conjugate pair**.

(4) **Hierarchical Bayesian models** naturally describe the connections among data, observed parameters and other unobserved parameters, sometimes called **latent variables**. A simple three-level hierarchical model uses Bayes' rule to bind together data, X, a parameter to be estimated, λ, and an additional hyper-parameter, β.

(5) The configuration maximizing the probability of a configuration given a randomly changed or blurred configuration, $\mathbb{P}\left[\omega \mid \omega^{\text{blurred}}\right]$, is the **maximum a posteriori estimate**.

(6) A version of the Metropolis algorithm called **Gibbs sampling** maximizes $\mathbb{P}\left[\omega \mid \omega^{\text{blurred}}\right]$, called the **maximum a posteriori estimate**.

(7) A probability distribution whose conditional probabilities depend on only the values in a neighborhood system is called a **Gibbs distribution** and is part of a larger notion called a **Markov random field**.

Notation.

(1) $f(x, y_1, \ldots, y_p)$ – a joint density

(2) $f_X(x) = \int \cdots \int f(x, y_1, \ldots, y_p) \, \mathrm{d}y_1 \ldots \mathrm{d}y_p$ – a marginal density

(3) p – number of parameters in the joint density

(4) $X_1, \ldots, X_m \sim f(x)$ – sample from the density $f(x)$

(5) m – number of elements in the sample

(6) (X, Y) – Starting pair of random variables

(7) $f(x \mid y)$ and $f(y \mid x)$ – conditional distributions

(8) $Y_0', X_0', Y_1', X_1', Y_2', X_2', \ldots, Y_n', X_n'$ – a *Gibbs sequence* of random variables

(9) p_1, p_2, p_3, p_4 – probabilities in a 2×2 table

(10) $P_{y|x}$ and $P_{x|y}$ – conditional probability of Y given X and of X given Y expressed in matrices

(11) $P_{x|x} = P_{y|x} P_{x|y}$ – the transition probability matrix for the X' sequence

(12) $f_n = (f_n(0), f_n(1))$ – the marginal probability distribution of X_n'

(13) π – stationary distribution satisfying $\pi P_{x|x} = \pi$

(14) ρ – correlation in bivariate normal distribution

(15) $[S = 1]$, the event that the email is in fact spam

(16) $[R = 1]$, the event that a spam email is correctly marked spam, a *true positive*,

(17) ψ – probability an email is in fact spam

(18) η – the false positive rate

(19) θ – the false negative rate

(20) τ – probability that a spam email is correctly marked spam

(21) $alpha, \beta$ – shape parameters for the Beta distribution

(22) $f(x_1, x_2, \ldots, x_N)$ – a joint probability density in which the variables represent parameters of a statistical model

(23) $\overrightarrow{x}^0 = (x_1^0, x_2^0, \ldots, x_N^0)$, $\overrightarrow{x}^1 = (x_1^1, x_2^1, \ldots, x_N^1)$ – sequence of samples

(24) $X \sim N(\mu, 1/\tau)$ – normal distribution with μ the population mean, and τ is the population precision

(25) $\bar{x}, s^2$– sample mean and variance

(26) T – number of iterations in Gibbs sampling

(27) X – parameter to estimate

(28) λ, β – parameter and hyper-parameter in Bayesian hierarchical model

(29) $\alpha, \beta, \gamma, \delta$ parameters for distributions

16.12. Sources

The sections on the general theory of the Gibbs sampler and the bivariate binomial are adapted from the article "Explaining the Gibbs Sampler" by George Casella and Edward George [**CG92**]. The section on Gibbs Sampling from the Bivariate Normal is adapted from "A simple Gibbs sampler" by Darren Wilkinson [**Wil**]. The section on A Gibbs Sampler for a Spam Filter is adapted from "Three Simple Applications of Markov Chains and Gibbs Sampling" by Gui Larangeira [**Lar**]. The section on theory and Bayesian hierarchical models is adapted from "The Evolution of Markov Chain Monte Carlo Methods" by Richey [**Ric10**].

16.13. Reading Suggestion:

16.14. Outside Readings and Links:

(1)

(2)

(3)

(4)

16.15. Algorithms and Scripts

> **Data:** Number of iterations N, "Burn in" iterations T, distributions and
>   parameters
> **Result:** Plots of Gibbs sampler results
> **1** Initialize X_0, Y_0
> **2 for** $1 : N$ **do**
> **3** $\quad$ Generate $X_j' \sim f(x \mid Y_j' = y_j')$
> **4** $\quad$ Generate $Y_{j+1}' \sim f(y \mid X_j' = x_j')$
> **5 end**
> **6** Plot images of resulting distributions and parameters from T to N;
>   **Algorithm 18:** Gibbs sampler algorithm

   Algorithm.

Scripts.

R: R script for Gibbs sampler for marginals from bivariate normal.

```
 1  gibbs<-function (n, rho)
 2  {
 3          mat <- matrix(ncol = 2, nrow = n)
 4          x <- 0
 5          y <- 0
 6          mat[1, ] <- c(x, y)
 7          for (i in 2:n) {
 8                  x <- rnorm(1, rho * y, sqrt(1 - rho^2))
 9                  y <- rnorm(1, rho * x, sqrt(1 - rho^2))
10                  mat[i, ] <- c(x, y)
11          }
12          mat
13  }
14
15  f   <- function(x) { return( (1/sqrt(2 * pi)) * exp(-x^2/2) )}
16
17  N   <-  1000
18  rho   <-  0.8
19
20  bvngs <- gibbs(N, rho)
21  bvn <- bvngs[(N/2 + 1):N, ]
22
23  par(mfrow=c(4,2))
24  hist(bvn[,1], freq=FALSE, 40)
25  curve(f, -4, 4, add=TRUE, col="red")
26  hist(bvn[,2], freq=FALSE, 40)
27  curve(f, -4, 4, add=TRUE, col="red")
28  qqnorm(bvn[ , 1], main="Q-Q Plot of bvn[, 1]")
29  qqnorm(bvn[ , 2], main="Q-Q Plot of bvn[ ,2]")
30  plot(bvn, col=1:500)
31  plot(bvn, type="l")
32  plot(ts(bvn[ , 1]))
33  plot(ts(bvn[ , 2]))
34  par(mfrow=c(1,1))
```

.

R script for Gibbs sampler for normal parameters.

```
 1  # summary statistics of sample
 2  n      <- 30
 3  ybar <- 15
 4  s2     <- 3
 5
 6  # sample from the joint posterior (mu, tau | data)
 7  mu       <- rep(NA, 11000)
 8  tau <- rep(NA, 11000)
 9  T        <- 1000     # burnin
10  tau[1] <-   1
11  ## tau[1]   <- 1   # initialisation
12  for(i in 2:11000) {
13      mu[i]   <- rnorm(n = 1, mean = ybar, sd = sqrt(1 / (n * tau[i - 1])))
14  ##     sigmasq[i] <- sigmasq[i-1] * rchisq(n = 1, n-1) / (n - 1)
15      tau[i] <- rgamma(n = 1, shape = n / 2, scale = 2 / ((n - 1) * s2 + n *
        (mu[i] - ybar)^2))
16  }
17  mu   <- mu[-(1:T)]    # remove burnin
18  tau <- tau[-(1:T)] # remove burnin
19
20  par(mfrow=c(1,2))
21  hist(mu)
22  hist(tau)
23  par(mfrow=c(1,1))
```

.

R script for Gibbs sampler for estimating spam prevalence.

```r
N <- 10000              # Number of iterations
Nb <- 2000; N1 <- N+1   # Burn-in

psi <- numeric(N)
psi[1] <- .5            # Initial value

alpha.0  <- 1.0
beta.0   <- 1.0
eta <- 0.99
theta <- 0.97
r <- 233
n   <- 1000

for(i in 2:N) {         # Gibbs Sampler Loop
tau<-psi[i-1]*eta+(1-psi[i-1])* (1-theta)
X <- rbinom(1, r, psi[i-1]* eta/tau)
Y <- rbinom(1, n-r, psi[i-1]*(1-eta)/(1-tau))
psi[i] <- rbeta(1, alpha.0+X+Y, beta.0+n-X-Y)
}

gspsi  <- mean(psi[Nb:N])

par(mfrow=c(2,2))
hist(psi)
plot(1:N,cumsum(psi)/(1:N),type="l",ylab= "psi", ylim=c(0.20,0.24))
plot(psi,type='p',pch='.',ylim=c(0.15,0.30))
acf(psi)
```

.

16.16. Problems to Work for Understanding

1:   For the simplest case where X and Y are each marginally Bernoulli random variables with joint distribution

$$
\begin{array}{cc}
 & \begin{array}{cc} X = 0 & X = 1 \end{array} \\
\begin{array}{c} Y = 0 \\ Y = 1 \end{array} & \left(\begin{array}{cc} p_1 & p_2 \\ p_3 & p_4 \end{array} \right)
\end{array}
$$

show that $f = (p_1 + p_3, p_2 + p_4)$ satisfies $f P_{X|X} = f$.

2:   The probability density function for the bivariate normal distribution with means 0 and variance 1 for the variables and correlation ρ between the two variables is

$$
f(x,y) = \frac{1}{2\pi\sqrt{1-\rho^2}} \mathrm{e}^{-\frac{x^2 - 2\rho xy + y^2}{2(1-\rho^2)}} .
$$

Show that the x marginal is

$$
f_x(x) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-x^2/2}.
$$

3:   The probability density function for the bivariate normal distribution with means 0 and variance 1 for the variables and correlation ρ between the two variables is

$$
f(x,y) = \frac{1}{2\pi\sqrt{1-\rho^2}} \mathrm{e}^{-\frac{x^2 - 2\rho xy + y^2}{2(1-\rho^2)}} .
$$

Show that the x conditional given y is

$$f_{x|y}(x \mid y) = \frac{1}{\sqrt{2\pi(1-\rho^2)}} e^{-\frac{(x-\rho y)^2}{2(1-\rho^2)}}.$$

Fastest Mixing Markov Chain

This chapter considers the problem of assigning transition probabilities to the edges+ of a simple linear network in such a way that the resulting Markov chain mixes as rapidly as possible. The problem is equivalent to assigning probabilities to a random walk with partially reflecting boundaries with each step either to the vertex itself or its nearest neighbor. It is also equivalent to certain urn models. The article proves the fastest mixing is obtained when each edge has a transition probability of 1/2. This result is intuitive.

Mathematicians Only: prolonged scenes of intense rigor.

17.1. Chapter Starter Question

If an irreducible aperiodic Markov chain is symmetric, that is, represented by a symmetric matrix, then what is the specific stationary distribution? What determines the rate at which the Markov chain approaches the stationary distribution?

17.2. Introduction, Motivation, Application

Consider a network (or mathematical graph) with $n \geq 2$ vertices, labeled $1, 2, \ldots n$ with $n - 1$ edges connecting adjacent vertices and with a loop at each vertex, as in Figure 1. Consider the Markov chain, or random walk, on this graph, with transition probability from vertex i to vertex j denoted P_{ij}. The requirement that transitions can occur only on an edge or loop of the graph is equivalent to $P_{ij} = 0$ when $|i - j| > 1$. Thus P is a tridiagonal matrix. Since P_{ij} are transition probabilities, $P_{ij} \geq 0$ and $\sum_j P_{ij} = 1$. Since the row sums are 1, P is a **stochastic matrix**. In matrix-vector terms,

$$(17.1) \qquad\qquad P\mathbf{1} = \mathbf{1}$$

where $\mathbf{1}$ is the $n \times 1$ vector whose entries are all 1. Therefore, 1 is an eigenvalue of the matrix P.
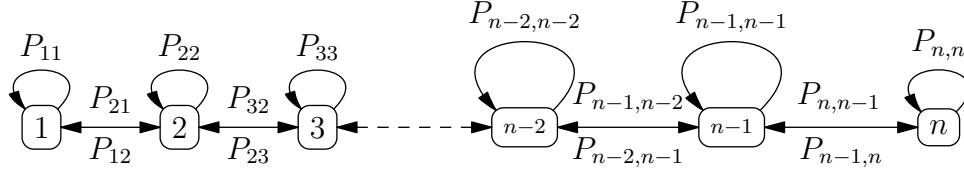
Figure 1. A random walk graph with transition probabilities.

Furthermore, assuming the transition probabilities are symmetric, so that $P_{ij} = P_{ji}$ means P is a symmetric, doubly-stochastic, tridiagonal matrix. Since $P\mathbf{1} = \mathbf{1}$, then $(1/n)\mathbf{1}^T P = (1/n)\mathbf{1}^T$, and the uniform distribution is a stationary distribution for the Markov chain.

The asymptotic rate of convergence of the Markov chain to the stationary distribution depends on the second-largest eigenvalue modulus of P, called the **mixing rate**. Since P is symmetric, the eigenvalues of P are real, and bounded in magnitude by 1. Ordering the eigenvalues by nonincreasing magnitude

$$1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq -1$$

the mixing rate is

$$\text{mix}(P) = \max_{i=2,\ldots,n} |\lambda_i(P)| = \max(\lambda_2, -\lambda_n)$$

See below for more information and proofs. The smaller $\text{mix}(P)$ is, the faster the Markov chain converges to its stationary distribution.

The following is an application and motivation for this Markov chain. A processor is located at each vertex of the graph. Each edge represents a direct network connection between the adjacent processors. The processor can be a computer in a network or it can be human worker, such as a line of assemblers in a workshop. The line has a constant total amount of work to be done. Each processor has a job load or queue to finish, say processor j has load $q_j(t)$ at time t where $q_j(t)$ is a positive real number. The load balancing is done in steps *before* any processing or work begins, so the total amount of work to be done is constant. At each step, the goal is shift loads between adjacent vertices to balance the load. In other words the goal is to have $q_j(t) \rightarrow (1/n)\sum_j q_j(0)$ as fast as possible as $t \rightarrow \infty$. The results here show the balance can be accomplished fastest by shifting one-half of the load on each vertex to the adjacent vertices.

17.3. Proofs about Fastest Mixing

Lemma 17.1. *If P is an $n \times n$ symmetric stochastic matrix, then*

$$\text{mix}(P) = \|P - (1/n)\mathbf{1}\mathbf{1}^T\|_2$$

where $\| \cdot \|_2$ denotes the spectral norm.

Remark 17.2. The **spectral norm**, also called the operator norm, is the natural norm of a matrix induced by the L^2 or Euclidean vector norm. The spectral norm is also the maximum singular value of the matrix, that is the square root of the maximum eigenvalue of $A^H A$ where A^H is the conjugate transpose.

R: Proof. (1) Note that $\mathbf{1}$ is the eigenvector of P associated with the eigenvalue $\lambda = 1$ by equation (17.1). Also $((1/n)\mathbf{1}\mathbf{1}^T)\mathbf{1} = \mathbf{1}$.

(2) Let $\mathbf{u}^{(2)}, \ldots \mathbf{u}^{(n)}$ be the other $n - 1$ eigenvectors of P corresponding to the eigenvalues $\lambda_2, \ldots \lambda_n$. Because P is symmetric, the eigenvectors are orthogonal. Taking the inner product of $\mathbf{u}^{(j)}$ with eigenvector $\mathbf{1}$, observe that $\sum_{\nu=1}^{n} u_\nu^{(j)} = 0$ for $j = 2, \ldots, n$. Therefore

$$((1/n)\mathbf{1}\mathbf{1}^T)\mathbf{u}^{(j)}(1/n)\mathbf{1}(\mathbf{1}^T\mathbf{u}^{(j)}) = \mathbf{0}.$$

(3) Then the eigenvalues of $P - (1/n)\mathbf{1}\mathbf{1}^T$ are $\lambda_1 = 0, \lambda_2, \ldots \lambda_n$ with eigenvectors $\mathbf{1}, \mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(n)}$ respectively.

(4) Since $P - (1/n)\mathbf{1}\mathbf{1}^T$ is symmetric, it is equivalent to a diagonal matrix, and its spectral norm is equal to the maximum magnitude of its eigenvalues, i.e. $\max\{|\lambda_2|, \ldots, |\lambda_n|\}$, which is $\mathrm{mix}(P)$.

$\square$

Lemma 17.3. *If P is an $n \times n$ symmetric stochastic matrix and if $\mathbf{y}$ and $\mathbf{z}$ in $\mathbb{R}^n$ satisfy*

$$\text{(17.2)} \qquad\qquad \mathbf{1}^T\mathbf{y} = 0$$

$$\text{(17.3)} \qquad\qquad \|\mathbf{y}\|_2 = 1$$

$$\text{(17.4)} \qquad\qquad (z_i + z_j)/2 \leq y_i y_j \text{ for } i, j \text{ with } P_{ij} \neq 0$$

then $\mathrm{mix}(P) \geq \mathbf{1}^T\mathbf{z}$.

Proof. (1) Let the eigenvectors of P be $\{\mathbf{u}^{(1)} = \mathbf{1}, \mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(n)}\}$. By the Spectral Theorem these eigenvectors are an orthonormal basis of $\mathbb{R}^n$.

(2) Let $\mathbf{y}$ be as specified in the hypotheses (17.2) and (17.3) and let

$$\mathbf{y} = \sum_{\nu=1}^{n} \alpha_\nu \mathbf{u}^{(\nu)}.$$

(3) By hypothesis (17.3), $\|\mathbf{y}\|_2 = \sqrt{\sum_{\nu=1}^{n} \alpha_\nu^2} = 1$.

(4) The eigenvalues of $P - (1/n)\mathbf{1}\mathbf{1}^T$ are $\lambda_1 = 0, \lambda_2, \ldots \lambda_n$ with eigenvectors $\mathbf{1}, \mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(n)}$ so

$$(P - (1/n)\mathbf{1}\mathbf{1}^T)\left(\sum_{\nu=1}^{n} \alpha_\nu \mathbf{u}^{(\nu)}\right) = \sum_{\nu=1}^{n} \alpha_\nu \lambda_\nu \mathbf{u}^{(\nu)} = \sum_{\nu=2}^{n} \alpha_\nu \lambda_\nu \mathbf{u}^{(\nu)}.$$

The last sum on the right side intentionally starts at 2 because the first eigenvalue is 0.

(5) By Lemma 17.1

$$\mathrm{mix}(P) = \|P - (1/n)\mathbf{1}\mathbf{1}^T\|_2.$$

By definition of the spectral norm

$$\|P - (1/n)\mathbf{1}\mathbf{1}^T\|_2 = \max_{\|\mathbf{w}\|=1} \|(P - (1/n)\mathbf{1}\mathbf{1}^T)\mathbf{w}\|_2.$$

(6) Specializing to the vector $\mathbf{y}$ with $\|\mathbf{y}\|_2 = 1$ and using the definition of the 2-norm

$$\max_{\|\mathbf{w}\|=1} \|(P - (1/n)\mathbf{1}\mathbf{1}^T)\mathbf{w}\|_2$$

$$\geq \|(P - (1/n)\mathbf{1}\mathbf{1}^T)\mathbf{y}\|_2$$

$$= \sqrt{\mathbf{y}^T(P - (1/n)\mathbf{1}\mathbf{1}^T)^T(P - (1/n)\mathbf{1}\mathbf{1}^T)\mathbf{y}}$$

$$= \sqrt{\left(\sum_{\nu=1}^{n} \alpha_\nu \mathbf{u}^{(\nu)}\right)^T (P - (1/n)\mathbf{1}\mathbf{1}^T)^T(P - (1/n)\mathbf{1}\mathbf{1}^T)\left(\sum_{\nu=1}^{n} \alpha_\nu \mathbf{u}^{(\nu)}\right)}$$

$$= \sqrt{\left(\sum_{\nu=1}^{n} \alpha_\nu \lambda_\nu \mathbf{u}^{(\nu)}\right)^T \left(\sum_{\nu=1}^{n} \alpha_\nu \lambda_\nu \mathbf{u}^{(\nu)}\right)}$$

$$= \sqrt{\sum_{\nu=1}^{n} \alpha_\nu^2 \lambda_\nu^2}.$$

(7) Apply Jensen's Inequality to the concave down function $\sqrt{\cdot}$ over the convex combination defined by α_i^2 which has $\sum_{i=1}^{n} \alpha_i^2 = \|\mathbf{y}\|_2 = 1$

$$\sqrt{\sum_{i=1}^{n} \alpha_i^2 \lambda_i^2} \geq \sum_{i=1}^{n} \alpha_i^2 |\lambda_i| \geq \sum_{i=1}^{n} \alpha_i^2 \lambda_i.$$

(8) Now unwind the expression back into vector notation, again using the orthogonality of the eigenvectors

$$\sum_{\nu=1}^{n} \alpha_\nu^2 \lambda_\nu = \left(\sum_{i=1}^{n} \alpha_i \mathbf{u}^{(i)}\right)^T (P - (1/n)\mathbf{1}\mathbf{1}^T)\left(\sum_{i=1}^{n} \alpha_i \mathbf{u}^{(i)}\right)$$

$$= \mathbf{y}^T(P - (1/n)\mathbf{1}\mathbf{1}^T)\mathbf{y}$$

$$= \mathbf{y}^T P \mathbf{y}$$

$$= \sum_{i,j} P_{ij} y_i y_j.$$

(9) Now use the first inequality in step 6 and hypothesis (17.4)

$$\sum_{i,j} P_{ij} y_i y_j \geq \sum_{i,j} (1/2)(z_i + z_j) P_{ij}$$

$$= (1/2)(\mathbf{z}^T P \mathbf{1} + \mathbf{1}^T P \mathbf{z})$$

$$= \mathbf{1}^T \mathbf{z}.$$

The second equality follows from the fact that P is doubly-stochastic. This establishes the lemma.

$$\square$$

Lemma 17.4. *The eigenvalues of the $n \times n$ tridiagonal matrix*

$$P_{1/2} = \begin{pmatrix} 1/2 & 1/2 & 0 & \cdots & & 0 \\ 1/2 & 0 & 1/2 & \cdots & & 0 \\ 0 & \ddots & \ddots & \ddots & & 0 \\ 0 & \cdots & 1/2 & 0 & & 1/2 \\ 0 & \cdots & & 0 & 1/2 & 1/2 \end{pmatrix}$$

are $\lambda_1 = 1$ and $\lambda_j = \cos\left(\frac{(j-1)\pi}{n}\right)$ for $j = 2, \ldots, n$.

Remark 17.5. The following proof is adapted from Feller [**Fel73**, Section XVI.4]. In fact, Feller finds the eigenvalues and eigenvectors for the more general tridiagonal matrix

$$P_p = \begin{pmatrix} q & p & 0 & \cdots & & 0 \\ q & 0 & p & \cdots & & 0 \\ 0 & \ddots & \ddots & \ddots & & 0 \\ 0 & \cdots & q & 0 & & p \\ 0 & \cdots & & 0 & q & p \end{pmatrix}.$$

The exercises derive the eigenvalues of this more general matrix.

Proof. (1) The proof directly finds the eigenvalues permitting solution of the linear system $P_{1/2}\mathbf{u} = \lambda\mathbf{u}$. The problem is treated as a linear system in the $(n-2) \times (n-2)$ system defined by equations $2, \ldots n-1$ in the variables $u_2, \ldots, u_{n-1}$ and then using the first and last equation as boundary conditions determining values of λ that permit a nontrivial solution.

(2) The equations are

(17.5) $$\lambda u_1 = (1/2)u_1 + (1/2)u_2$$

(17.6) $$\lambda u_j = (1/2)u_{j-1} + (1/2)u_{j+1}, j = 2, \ldots. n-1$$

(17.7) $$\lambda u_n = (1/2)u_{n-1} + (1/2)u_n.$$

(3) Equation (17.6) is satisfied by $u_j = s^j$ provided

$$\lambda s = (1/2) + (1/2)s^2$$

or $s_+ = \lambda + \sqrt{\lambda^2 - 1}$ and $s_- = \lambda - \sqrt{\lambda^2 - 1}$. Then the general solution is of the form $u_j = A(\lambda)s_+^j + B(\lambda)s_-^j$.

(4) Apply the first equation ((17.5)) to get

$$\lambda(As_+ + Bs_-) = (1/2)(As_+ + Bs_-) + (1/2)(As_+^2 + Bs_-^2)$$
$$0 = A[(1 - 2\lambda)s_+ + s_+^2] + B[(1 - 2\lambda)s_- + s_-^2]$$
$$0 = A[s_+ - 1] + B[s_- - 1]$$
$$A[s_+ - 1] = -B[s_- - 1].$$

(5) Apply the last equation (17.7) to get

$$\lambda(As_+^n + Bs_-^n) = (1/2)(As_+^{n-1} + Bs_-^{n-1}) + (1/2)(As_+^n + Bs_-^n)$$
$$0 = As_+^{n-1}[1 - 2\lambda s_+ + s_+] + Bs_-^{n-1}[1 - 2\lambda s_- + s_-]$$
$$0 = As_+^{n-1}[-s_+^2 + s_+] + Bs_-^{n-1}[-s_-^2 + s_-]$$
$$0 = As_+^n[-s_+ + 1] + Bs_-^n[-s_- + 1]$$
$$-As_+^n[-s_+ + 1] = Bs_-^n[-s_- + 1]$$
$$A[s_+ - 1]s_+^n = -B[s_- - 1]s_-^n.$$

(6) Combining these equations $s_+^n = s_-^n$. Note that $s_+ s_- = 1$, so $s_+^{2n} = 1$ and $s_-^{2n} = 1$. That is, both s_+ and s_- are $2n$th roots of unity. Therefore, s_+ and s_- can be written in the form

$$e^{i\pi j/n} = \cos\left(\frac{\pi j}{n}\right) + i\sin\left(\frac{\pi j}{n}\right)$$

for $j = 0, 1, 2, \ldots, 2n - 1$.

(7) Thus, the eigenvalues λ must be a solution of

$$s_+(\lambda) = e^{i\pi j/n}$$

or

$$\lambda + \sqrt{\lambda^2 - 1} = e^{i\pi j/n}$$
$$\sqrt{\lambda^2 - 1} = -\lambda + e^{i\pi j/n}$$
$$\lambda^2 - 1 = \lambda^2 - 2\lambda e^{i\pi j/n} + e^{2i\pi j/n}$$
$$-1 = -2\lambda e^{i\pi j/n} + e^{2i\pi j/n}$$
$$\frac{1}{2e^{i\pi j/n}} + \frac{e^{i\pi j/n}}{2} = \lambda$$
$$\frac{e^{-i\pi j/n}}{2} + \frac{e^{i\pi j/n}}{2} = \lambda$$
$$\cos \pi j/n = \lambda.$$

(8) So to each j there is a root λ_j, namely

$$\lambda_j = \cos(\pi j/n) \qquad j = 0, 1, 2, \ldots, 2n - 1.$$

However, some of these roots are repeated, since

$$\cos(\pi j/n) = \cos(\pi(2n - j)/n)$$

for $j = 1, \ldots, n$. The n eigenvalues are $\lambda_1 = 1$ and $\lambda_j = \cos\left(\frac{(j-1)\pi}{n}\right)$ for $j = 2, \ldots, n$.

$\square$

Remark 17.6. The following lemma is a purely technical lemma about sums of cosines of arithmetic sequences. The next two lemmas use results of this lemma to shorten the proofs.

Lemma 17.7. *(1) For an integer j*

$$\sum_{\ell=1}^{n} \cos((2\ell-1)(j-1)\pi/n) = 0.$$

(2) In the special case $j = 2$

$$\sum_{\ell=1}^{n} \cos((2\ell-1)\pi/n) = 0.$$

(3) For an integer j

$$\sum_{\ell=1}^{n} \cos^2((2\ell-1)(j-1)\pi/(2n)) = \frac{n}{2}.$$

Proof. (1)

$$\sum_{\ell=1}^{n} \cos((2\ell-1)(j-1)\pi/n)$$

$$= \Re\left(\sum_{\ell=1}^{n} \exp(\mathrm{i}(2\ell-1)(j-1)\pi/n)\right)$$

$$= \Re\left(\sum_{\ell=1}^{n} \exp(2\ell(j-1)\pi\mathrm{i}/n)\exp(-\mathrm{i}(j-1)\pi/n)\right)$$

$$= \Re\left(\exp(-\mathrm{i}(j-1)\pi/n)\sum_{\ell=1}^{n}[\exp(2(j-1)\pi\mathrm{i}/n)]^{\ell}\right)$$

$$= \Re\left(\exp(-\mathrm{i}(j-1)\pi/n)\exp(2(j-1)\pi\mathrm{i}/n)\sum_{\ell=1}^{n}[\exp(2(j-1)\pi\mathrm{i}/n)]^{\ell-1}\right)$$

$$= \Re\left(\exp(-(j-1)\pi\mathrm{i}/n)\exp(2(j-1)\pi\mathrm{i}/n)\sum_{\ell=0}^{n-1}[\exp(2(j-1)\pi\mathrm{i}/n)]^{\ell}\right)$$

$$= \Re\left(\exp(\mathrm{i}(j-1)\pi/n)\frac{1-[\exp(2(j-1)\pi\mathrm{i}/n)]^{n}}{1-\exp(2(j-1)\pi\mathrm{i}/n)}\right)$$

$$= \Re\left(\exp(\mathrm{i}(j-1)\pi/n)\frac{1-\exp(2(j-1)\pi\mathrm{i})}{1-\exp(2(j-1)\pi\mathrm{i}/n)}\right)$$

$$= 0.$$

(2) This is an immediate corollary of the previous with $j = 2$.

(3) Use the half-angle formula for the cosine function so

$$\cos^2((2\ell-1)(j-1)\pi/(2n)) = \frac{1}{2} + \cos((2\ell-1)(j-1)\pi/n)$$

and the result follows from the first part.

$\square$

Lemma 17.8. *The normalized eigenvectors of the $n \times n$ tridiagonal matrix*

$$P_{1/2} = \begin{pmatrix} 1/2 & 1/2 & 0 & \cdots & 0 \\ 1/2 & 0 & 1/2 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 1/2 & 0 & 1/2 \\ 0 & \cdots & 0 & 1/2 & 1/2 \end{pmatrix}$$

are

$$\lambda_1 = 1, \qquad\qquad \mathbf{u}^{(1)} = \frac{1}{\sqrt{n}}\mathbf{1}$$

$$\lambda_j = \cos\left(\frac{(j-1)\pi}{n}\right), \qquad u_k^{(j)} = \sqrt{\frac{2}{n}}\cos\left(\frac{(2k-1)(j-1)\pi}{2n}\right)$$

for $j = 2, \ldots, n$ and $k = 1, \ldots, n$.

Proof. (1) Since $P_{1/2}$ is symmetric, already it is established that for $\lambda_1 = 1$, the corresponding normalized eigenvector is $\mathbf{u}^{(1)} = \frac{1}{\sqrt{n}}\mathbf{1}$.

(2) For $j \geq 2$, $\lambda_j = \cos\left(\frac{(j-1)\pi}{n}\right)$. The proof of Lemma 17.4 shows the corresponding unnormalized eigenvectors have components

$$u_k^{(j)} = A(\lambda_j)s_+^k(\lambda_j) + B(\lambda_j)s_-^k(\lambda_j)$$

where $s_\pm(\lambda_j) = \lambda_j \pm \sqrt{\lambda_j^2 - 1} = \cos((j-1)\pi/n) \pm \mathrm{i}\sin((j-1)\pi/n) = \mathrm{e}^{\pm(j-1)\pi\mathrm{i}}$.

(3) The coefficients $A(\lambda_j)$ and $B(\lambda_j)$ are determined from the boundary conditions (17.5) and (17.7). But since

$$u_k^{(j)} = A(\lambda_j)\left[s_+^k(\lambda_j) + \frac{B(\lambda_j)}{A(\lambda_j)}s_-^k(\lambda_j)\right]$$

and any scalar multiple of an eigenvector is an eigenvector, only the ratio $\frac{B(\lambda_j)}{A(\lambda_j)}$ is needed.

(4) For notational simplicity, temporarily drop the dependence of $s_\pm$ on λ_j. Expanding (17.5)

$$2\lambda As_+ + 2\lambda Bs_- = As_+ + Bs_- + As_+^2 + Bs_-^2.$$

(5) Rearranging and using $s_\pm^2 - 2\lambda s_\pm = -1$

$$\frac{B}{A} = -\frac{s_+ - 1}{s_- - 1}$$

and the components of the unnormalized eigenvector are

$$u_k^{(j)} = \left[s_+^k(\lambda_j) - \frac{s_+(\lambda_j) - 1}{s_-(\lambda_j) - 1}s_-^k(\lambda_j)\right].$$

Factoring out $s_+ - 1$ and remembering that any scalar multiple of an eigenvector is an eigenvector, the components of the unnormalized eigenvector are

$$u_k^{(j)} = \left[\frac{s_+^k(\lambda_j)}{s_+(\lambda_j) - 1} - \frac{s_-^k(\lambda_j)}{s_-(\lambda_j) - 1}\right].$$
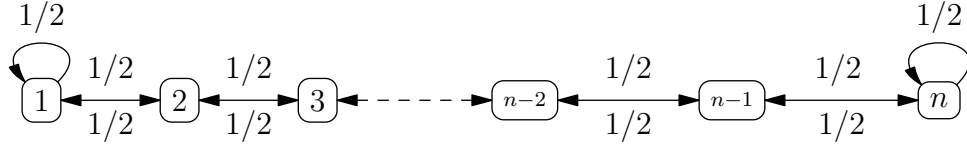
Figure 2. The random walk graph with fastest mixing.

(6) Recalling $s_\pm(\lambda_j) = e^{\pm i(j-1)\pi/n}$, each fraction can be simplified by multiplying numerator and denominator by the complex conjugate of the denominator

$$\frac{e^{\pm i(j-1)\pi k/n}}{e^{\pm i(j-1)\pi/n} - 1} \cdot \frac{e^{\mp i(j-1)\pi k/n} - 1}{e^{\mp i(j-1)\pi/n} - 1} = \frac{e^{\pm i\pi(j-1)(k-1)/n} - e^{\pm i\pi(j-1)k/n}}{2 - 2\cos(\pi/n)}$$

(7) Once again remembering that any scalar multiple of an eigenvector is an eigenvector, the constant $2 - 2\cos(\pi/n)$ in the denominator can be ignored.

(8) This leaves

$$u_k^{(j)} = (e^{+i(j-1)(k-1)\pi/n} - e^{-i\pi(j-1)(k-1)/n}) - (e^{+i(j-1)k\pi/n} - e^{-i(j-1)k\pi/n})$$
$$= 2\sin((j-1)(k-1)\pi/n) - 2\sin((j-1)k\pi/n)$$
$$= 4\sin((j-1)(-1)\pi/(2n))\cos((j-1)(2k-1)\pi/(2n)).$$

(9) Finally, factoring out the constant $-4\sin((j-1)\pi/(2n))$, the kth component of the eigenvector corresponding to λ_j is

$$u_k^{(j)} = \cos\left(\frac{(2k-1)(j-1)\pi}{2n}\right).$$

(10) Using Lemma 17.7, the norm is $\sqrt{n/2}$ and the kth component of the normalized eigenvector corresponding to λ_j is

$$u_k^{(j)} = \sqrt{\frac{2}{n}}\cos\left(\frac{(2k-1)(j-1)\pi}{2n}\right).$$

$\square$

Theorem 17.9. *The value* $\mathrm{mix}(P_{1/2}) = \cos(\pi/n)$ *for*

$$P_{1/2} = \begin{pmatrix} 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ & \ddots & \ddots & \ddots & \\ & & 1/2 & 0 & 1/2 \\ & & & 1/2 & 1/2 \end{pmatrix}$$

is the smallest among all symmetric stochastic tridiagonal matrices.

See Figure 2 for the graph associated with this transition probability matrix.

Proof. (1) The proof proceeds by constructing a pair of vectors **y** and **z** that satisfy the assumptions of Lemma 17.3 for any symmetric stochastic tridiagonal matrix P. Furthermore, $\mathbf{1}^T\mathbf{z} = \cos(\pi/n)$, so the mixing rate $\mathrm{mix}(P_{1/2}) = \cos(\pi/n)$ is the fastest possible.

(2) By Lemma 17.4 and the definition of the mixing rate

$$\text{mix}(P_{1/2}) = \lambda_2 = -\lambda_n = \cos(\pi/n).$$

(3) Take $\mathbf{y} = \mathbf{u}^{(2)}$, the normalized eigenvector of $P_{1/2}$ corresponding to λ_2, so the assumptions (17.2) and (17.3) in Lemma 17.3 are automatically satisfied.

(4) Take $\mathbf{z}$ to be the vector with

$$z_j = \frac{1}{n} \left[\cos\left(\frac{\pi}{n}\right) + \cos\left(\frac{(2j-1)\pi}{n}\right) \middle/ \cos\left(\pi/n\right) \right]$$

for $j = 1, \ldots, n$.

(5) By Lemma 17.7

$$\sum_{j=1}^{n} \cos((2j-1)\pi/n) = 0.$$

Then $\mathbf{1}^T \mathbf{z} = \cos(\pi/n)$.

(6) Now check that $\mathbf{y}$ and $\mathbf{z}$ satisfy hypothesis (17.4) of Lemma 17.3.

$$\frac{z_j + z_{j+1}}{2} =$$
$$\frac{1}{n} \left[\cos\left(\frac{\pi}{n}\right) + \frac{1}{2} \left(\cos\left(\frac{(2j-1)\pi}{n}\right) + \cos\left(\frac{(2j+1)\pi}{n}\right) \right) \middle/ \cos\left(\pi/n\right) \right].$$

(7) Using the cosine sum formula

$$\frac{1}{2} \left(\cos\left(\frac{(2j-1)\pi}{n}\right) + \cos\left(\frac{(2j+1)\pi}{n}\right) \right) = \cos\left(\frac{\pi}{n}\right) \cos\left(\frac{2j\pi}{n}\right)$$

this simplifies to

$$\frac{z_j + z_{j+1}}{2} = \frac{1}{n} \left[\cos\left(\frac{\pi}{n}\right) + \cos\left(\frac{2j\pi}{n}\right) \right].$$

(8) Using the cosine sum formula again

$$\frac{z_j + z_{j+1}}{2} = \frac{1}{n} \left[\cos\left(\frac{\pi}{n}\right) + \cos\left(\frac{2j\pi}{n}\right) \right]$$
$$= \frac{2}{n} \cos\left(\frac{(2j-1)\pi}{2n}\right) \cos\left(\frac{(2j+1)\pi}{2n}\right)$$
$$= y_j y_{j+1}.$$

(9) Therefore *equality* in inequality (17.4) holds for the entries resulting from the nonzero subdiagonal and superdiagonal entries.

(10) For the diagonal entries, check that $(z_j + z_j)/2 = z_j \leq y_j^2$. That is, check that

$$\cos\left(\frac{\pi}{n}\right) + \cos\left(\frac{(2j-1)\pi}{n}\right) \middle/ \cos\left(\frac{\pi}{n}\right) \leq 2\cos^2\left(\frac{(2j-1)\pi}{2n}\right).$$

(11) Using the double-angle formula for the cosine

$$2\cos^2\left(\frac{(2j-1)\pi}{n}\right) = 1 + \cos\left(\frac{(2j-1)\pi}{n}\right).$$

Therefore, the following inequality needs to be checked

$$\cos\left(\frac{\pi}{n}\right) + \cos\left(\frac{(2j-1)\pi}{n}\right) \Bigg/ \cos\left(\frac{\pi}{n}\right) \le 1 + \cos\left(\frac{(2j-1)\pi}{n}\right).$$

(12) Moving all terms to the right the inequality to check is

$$1 - \cos\left(\frac{\pi}{n}\right) - \cos\left(\frac{(2j-1)\pi}{n}\right) \Bigg/ \cos\left(\frac{\pi}{n}\right) + \cos\left(\frac{(2j-1)\pi}{n}\right) \ge 0.$$

(13) The left side can be factored as

$$\left[1 - \cos\left(\frac{\pi}{n}\right)\right]\left[1 - \cos\left(\frac{(2j-1)\pi}{n}\right) \Bigg/ \cos\left(\frac{\pi}{n}\right)\right].$$

(14) This is true because

$$\cos\left(\frac{(2j-1)\pi}{n}\right) \Bigg/ \cos\left(\frac{\pi}{n}\right) \le 1$$

for $j = 1, \ldots, n$.

$\square$

17.4. Chapter Ending Answer

Since the transition probability matrix P is stochastic, $P\mathbf{1} = \mathbf{1}$. Since the matrix P is symmetric, $\mathbf{1}^T P = \mathbf{1}^T$. Then normalizing, $(1/n)\mathbf{1}^T P = (1/n)\mathbf{1}^T$ so the uniform distribution is a stationary distribution. Intuitively, this make sense since if the matrix is symmetric, the transition probabilities are the same to go from i to j as from j to i, so the stationary distribution will "even out" to be uniform. The rate of convergence is determined by the eigenvalues, and the eigenvalue with second largest magnitude (since the largest eigenvalue is 1) will dominate the asymptotic rate.

17.5. Chapter Summary

Key Concepts.

(1) The asymptotic rate of convergence of the Markov chain to the stationary distribution depends on the second-largest eigenvalue modulus, called the mixing rate. The mixing rate is denoted by μ or $\mu(P)$ where P is the transition probability matrix.

(2) If P is an $n \times n$ symmetric stochastic matrix, then

$$\mu(P) = \|P - (1/n)\mathbf{1}\mathbf{1}^T\|_2$$

where $\|\cdot\|_2$ denotes the spectral norm.

(3) The eigenvalues and eigenvectors of the tridiagonal matrix

$$P_{1/2} = \begin{pmatrix} 1/2 & 1/2 & 0 & & & \\ 1/2 & 0 & 1/2 & & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1/2 & 0 & 1/2 \\ & & & & 1/2 & 1/2 \end{pmatrix}$$

can be determined by solving a recursive system of equations and are $\lambda_j = \cos\left(\frac{(j-1)\pi}{n}\right)$ for $j = 1, \ldots, n$. In particular, the largest eigenvalue is 1.

(4) The mixing rate $\mu(P_{1/2}) = \cos(\pi/n)$ for $P_{1/2}$ is the smallest among all symmetric stochastic tridiagonal matrices.

Vocabulary.

(1) A matrix for which the row sums $\sum_j P_{ij} = 1$ is a **stochastic matrix**.

(2) The asymptotic rate of convergence of the Markov chain to the stationary distribution depends on the second-largest eigenvalue modulus of P, called the **mixing rate**.

(3) The **spectral norm**, also called the operator norm, is the natural norm of a matrix induced by the L^2 or Euclidean vector norm. The spectral norm is also the maximum singular value of the matrix, that is the square root of the maximum eigenvalue of $A^H A$ where A^H is the conjugate transpose.

Notation.

(1) n – number of nodes in the network

(2) i, j – typical nodes in the network

(3) P_{ij} – transition probability from i to j

(4) P – transition probability matrix

(5) $P_{1/2}$ – tridiagonal transition probability matrix with every entry $1/2$.

(6) $\mathbf{1}$ – the $n \times 1$ vector whose entries are all 1

(7) $1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq -1$ – the eigenvalues ordered by nonincreasing magnitude

(8) $\mathrm{mix}(P) = \max_{i=2,\ldots,n} |\lambda_i(P)| = \max(\lambda_2, -\lambda_n)$ – the mixing rate

(9) A^H – conjugate transpose of matrix A

(10) $\mathbf{u}^{(2)}, \ldots \mathbf{u}^{(n)}$ – the $n - 1$ eigenvectors of P corresponding to the eigenvalues $\lambda_2, \ldots \lambda_n$

(11) $\mathbf{y}, \mathbf{z}$ – vectors in $\mathbb{R}^n$

(12) s_-, s_+ – solutions of the quadratic characteristic function.

17.6. Sources

This section is adapted from the article by Boyd, Diaconis, Sun and Xiao [**BDSX06**].

17.7. Reading Suggestion:

17.8. Outside Readings and Links:

(1)

(2)

(3)

(4)

17.9. Algorithms and Scripts

Data: Transition probability values
Result: Eigenvalues of tridiagonal stochastic matrix

Algorithm.

1 Initialize tridiagonal values for stochastic matrix **for** $n = 2 : 8$ **do**
2 | Build tridiagonal stochastic $n \times n$ matrix. Calculate eigenvalues of the matrix and display.
3 **end**

Scripts.

R: R script for .

```
      A simple R program using only base functions:
\begin{lstlisting}
r <- 0
p <- 1/2
q <- 1 - p

for (n in 2:8) {

   P <- diag(r, n, n)
   P[ row(P) - col(P)  ==  1] <- p
   P[ row(P) - col(P)  == -1] <- q
   P[1, 1] = q
   P[n, n] = p

   cat("Size of matrix: ", n , "Eigenvalues: ",
       unlist(eigen( P, only.values=TRUE )), "\n")
}
```

.

17.10. Problems to Work for Understanding

1: Show that balancing the workload of a line of processors by shifting one-half of the load on each vertex from the more loaded to the less loaded processor can be represented by the tridiagonal matrix $P_{1/2}$.

2: Show that $(1/n)\left(\mathbf{1}\mathbf{1}^T\right)\mathbf{1} = \mathbf{1}$.

3: Devise the exchange rules for an urn model with the $n \times n$ transition probability matrix P_p.

4: Show that

$$\frac{1}{2}\left(\cos\left(\frac{(2j-1)\pi}{n}\right) + \cos\left(\frac{(2j+1)\pi}{n}\right)\right) = \cos\left(\frac{\pi}{n}\right)\cos\left(\frac{2j\pi}{n}\right)$$

and

$$\frac{1}{n}\left[\cos\left(\frac{\pi}{n}\right) + \cos\left(\frac{2j\pi}{n}\right)\right] = \frac{2}{n}\cos\left(\frac{(2j-1)\pi}{2n}\right)\cos\left(\frac{(2j+1)\pi}{2n}\right).$$

5: Show that the four eigenvalues of

$$P_{1/2} = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix}$$

are $\cos\left(\frac{(j-1)\pi}{n}\right)$ for $j = 1, \ldots, 4$.

  6: Show that the eigenvalues and for the more general tridiagonal matrix

$$P_p = \begin{pmatrix} q & p & 0 & \cdots & 0 \\ q & 0 & p & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & q & 0 & p \\ 0 & \cdots & 0 & q & p \end{pmatrix}$$

are $\lambda_1 = 1$ and $\lambda_j = 2\sqrt{pq}\cos\left(\frac{(j-1)\pi}{n}\right)$ for $j = 2, \ldots, n$. Then show directly that $P_{1/2}$ with $p = q = \frac{1}{2}$ is the fastest mixing among all such tridiagonal matrices P.

  7:   Use mathematical software to numerically evaluate the eigenvalues of the matrices $P_{1/2}$ for sizes $n = 2, \ldots 8$ and show that the values agree with the exact eigenvalues in Lemma 17.4.

Card Shuffling as a Markov Chain

18.1. Rating

An unopened deck of cards has the face-up order (depending on manufacturer, but typically in the U.S.), starting with the Ace of Spades:

- Ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King of Spades,
- Ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King of Diamonds,
- King, Queen, Jack, 10, 9, 8, 7, 6, 5, 4, 3, 2, Ace of Clubs, then
- King, Queen, Jack, 10, 9, 8, 7, 6, 5, 4, 3, 2, Ace of Hearts.

Call this the initial order of the deck. Knowing this order is essential for some sleight of hand tricks performed by a magician. For card players, shuffling the deck to remove this order is essential so that cards dealt from the deck come "at random", that is, in an order uniformly distributed over all possible deck orders. The main question here is: Starting from this order, how many shuffles are necessary to obtain a "random" deck order from the uniform distribution?

In terms of Markov chains, the questions are: What is the state space, what is an appropriate transition probability matrix, what is the steady state distribution, hopefully uniform, and how fast does the Markov process approach the steady state distribution?

Mathematically Mature: may contain mathematics beyond calculus with proofs.

18.2. Chapter Starter Question

Why shuffle a deck of cards? What kind of shuffle do you use? How many shuffles are sufficient to achieve the purpose of shuffling?

18.3. General Setting

For simplicity and definiteness, let the cards in the initial deck order

- Ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King of Spades,
- Ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King of Diamonds,
- King, Queen, Jack, 10, 9, 8, 7, 6, 5, 4, 3, 2, Ace of Clubs, then
- King, Queen, Jack, 10, 9, 8, 7, 6, 5, 4, 3, 2, Ace of Hearts.

be numbered 1 to 52. It will also be convenient to study much smaller decks of cards having D cards. The set of states for a Markov process modeling the order of the deck is S_D, the set of permutations on D cards. For convenience, set the initial state X_0 to be the identity permutation with probability 1. In other words, choose the initial distribution as a not yet shuffled deck.

Consider a shuffle, that is, a re-ordering operation on a state that takes an order to another order. For example, the riffle shuffle, also called a dovetail shuffle or leafing the cards, is a common type of shuffle that interleaves packets of cards. A perfect riffle shuffle, also called a faro shuffle, splits the deck exactly in half, then interleaves cards alternately from each half. A perfect rifle shuffle is difficult to perform except for practiced magicians. More commonly, packets of adjacent cards from unevenly split portions interleave, creating a new order for the deck that nevertheless preserves some of the previous order in each packet. Thus a particular riffle shuffle is one of a whole family of riffle shuffles, chosen with a probability distribution on the family. This probability distribution then induces a transition probability from state to state, and thus a Markov process.

Other types of shuffles have colorful names such as the Top-to-Random-Shuffle, Hindu shuffle, pile shuffle, Corgi shuffle, Mongean shuffle, and Weave shuffle. Some shuffle types are a family of possible re-orderings with probability distributions different from the riffle shuffle distribution, leading to different transition probabilities, and thus different Markov processes.

Going from card order π to σ is the same as composing π with the permutation $\pi^{-1} \circ \sigma$. Now identify shuffles as functions on $\{1, \ldots, n\}$ to $\{1, \ldots, D\}$, that is, permutations. Since a particular riffle shuffle is one of a whole family of riffle shuffles, chosen with a probability distribution Q from the family, the transition probabilities are $p_{\pi\sigma} = \mathbb{P}\left[X_{t+1} = \sigma \mid X_t = \pi\right] = Q(\pi^{-1} \circ \sigma)$. So now the goal is to describe the probability distribution Q and apply it to the Markov process.

Remark 18.1. This chapter uses a list notation for permutations. For example, the notation $\pi = [231]$ represents the permutation with $\pi(1) = 2$, $\pi(2) = 3$ and $\pi(3) = 1$. A common alternative matrix notation for the same permutation is

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}.$$

Writing the permutation in matrix form makes finding the inverse obvious, $\pi^{-1} = [312]$.

Recall also that sequential permutations are applied from right to left. Composing π with the permutation $\pi^{-1} \circ \sigma$ gives $\pi \circ (\pi^{-1} \circ \sigma) = \sigma$. If $\sigma = [132]$, then $\pi^{-1} \circ \sigma = [321]$ and $[132] = [231] \circ [321]$.

This chapter does not use cycle notation for permutations.

18.4. Top-to-Random-Shuffle

A particularly simple shuffle is the **Top-to-Random-Shuffle**, abbreviated TTRS. The TTRS takes the top card from a stack of D cards and inserts it in the gap between the $(k-1)$th card and the kth card in the deck. See Figure 1. Note that $k = 1$ is possible, in which case the top card returns to the top. Likewise, $k = n+1$ is also permitted, in which case the top card moves to the bottom of the card stack.

Consider the order of the cards to be a permutation on n symbols. The TTRS is naturally a finite Markov chain X_t for $t \geq 0$ with $X_t \in S_n$. Set $X_0 = \sigma_0$, the identity permutation. The transition probabilities are

$$\mathbb{P}\left[X_{t+1} = \sigma' \mid X_t = \sigma\right] = \begin{cases} \frac{1}{n} & \sigma' \text{ is a TTRS of } \sigma \\ 0 & \text{otherwise} \end{cases}$$

defining the transition probability matrix P. Then after t TTRS shuffles, the order of the deck with D card has a probability distribution $X_0 P^t$ on S_D, where with an overload of notation X_0 is the vector with a 1 in the position for σ_0 and 0 elsewhere, representing the initial state. The Markov chain X_t induced by the TTRS is irreducible, see the exercises. It is also immediate that X_t is aperiodic since it is possible that the top card can recur back on top. Therefore, this Markov chain must converge to a stationary distribution and this section will later prove that $X_0 P^t \to \text{unif}(S_D)$.

Example 18.2. The transition matrix for the TTRS on a deck with three cards is

$$\begin{array}{c} \\ \begin{array}{c} [123] \\ [213] \\ [231] \\ [132] \\ [312] \\ [321] \end{array} \end{array} \begin{array}{cccccc} [123] & [213] & [231] & [132] & [312] & [321] \\ \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} \end{pmatrix} \end{array}.$$

If the card deck is initially in order 1 to D from top to bottom, how many TTRS shuffles does it take for the deck to be sufficiently shuffled? Starting with the identity ordering, the density of the permutations after 7 top-to-random shuffles is the first row of P^7. Numerically,

$$P^7 = \begin{pmatrix} 0.16690 & 0.16690 & 0.16690 & 0.16644 & 0.16644 & 0.16644 \\ 0.16690 & 0.16690 & 0.16644 & 0.16690 & 0.16644 & 0.16644 \\ 0.16644 & 0.16644 & 0.16690 & 0.16644 & 0.16690 & 0.16690 \\ 0.16644 & 0.16644 & 0.16644 & 0.16690 & 0.16690 & 0.16690 \\ 0.16690 & 0.16644 & 0.16644 & 0.16690 & 0.16690 & 0.16644 \\ 0.16644 & 0.16690 & 0.16690 & 0.16644 & 0.16644 & 0.16690 \end{pmatrix}.$$

That is, 7 shuffles of the 3-card deck gets close to the stationary density, which turns out to be the uniform density. The eigenvalues of P are $1, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0$.

$\triangledown$

Stack Position
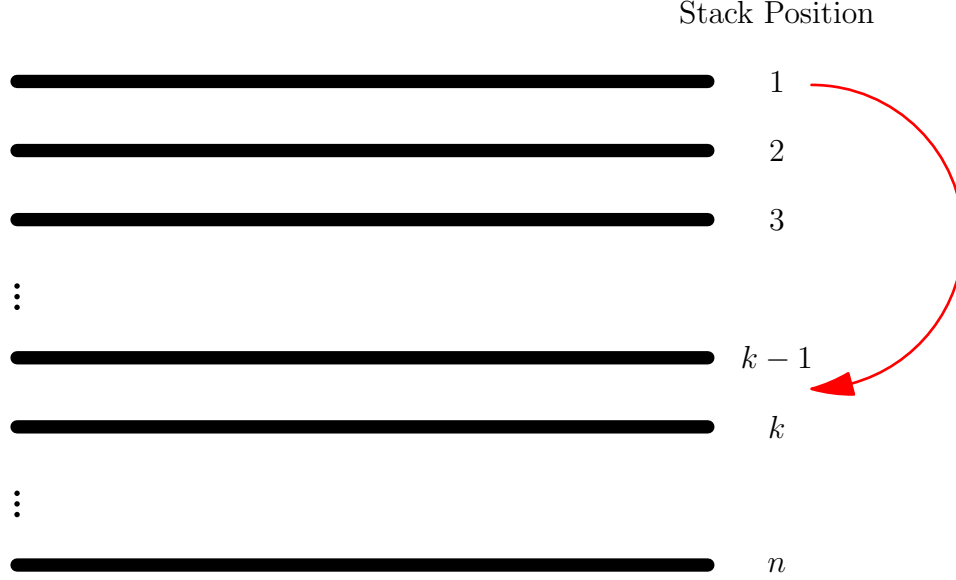
Figure 1. Schematic drawing of the Top-to-Random-Shuffle.
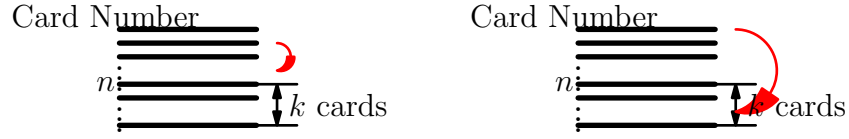
Figure 2. Schematic diagram of the proof of the Lemma.

Lemma 18.3. *At any time t, if k cards appear beneath the card labeled D, then these cards appear in any order with equal probability.*

Proof. The proof is by induction on t. The base case $t = 0$ is trivial. Suppose that the claim is true for some $t > 0$. In the transition to $t + 1$, two cases can occur, see Figure 2 for a schematic diagram. First, the top card is randomly placed above the card labeled n that is somewhere in the stack. Then nothing is changed and the proof is complete. Otherwise, the top card is placed in one of the $k + 1$ available spaces below the last card (labeled n) that is somewhere in the stack. The probability of any particular one of these arrangements is

$$\frac{1}{k!} \cdot \frac{1}{k+1} = \frac{1}{(k+1)!}$$

where $\frac{1}{k!}$ comes from the induction hypothesis and the $\frac{1}{k+1}$ comes from the TTRS. The proof is complete. □

Theorem 18.4. *Let τ_{top} be the first time that card D reaches the top of the deck. Then $X_0 P^{\tau_{top}+1}$ is uniform on S_D. Furthermore, whatever permutation arises at time $\tau_{top} + 1$ is independent of τ_{top}.*

Proof. The proof follows from the Lemma, since at time τ_{top} the $D-1$ cards below card D will be uniformly distributed over the $(D-1)!$ possible permutations. Then at time $\tau_{\text{top}} + 1$ card n is inserted uniformly at random in the deck. $\qquad\square$

Remark 18.5. Waiting for τ_{top} is the same as waiting for completion in the "coupon collectors problem in reverse". More precisely, collecting a coupon here is putting the top card below the card labeled D. The first card is hard to put under D, in fact it happens with probability $\frac{1}{n+1}$ but it gets easier with more shuffles. This motivates the later assertions that $\mathbb{E}\left[\tau_{\text{top}} + 1\right] = \Theta(n \log n)$ and that $\mathbb{P}\left[\tau_{\text{top}} + 1 \geq n \log n + cn\right] \leq e^{-c}$ for all $c \geq 0$. See below for more details.

Definition 18.6. If μ and ν are probability distributions on Ω, the **total variation distance** of μ from ν is

$$\|\mu - \nu\|_{TV} = \sup_{A \subset \Omega} |\mu(A) - \nu(A)| = \frac{1}{2} \sum_{x \in \Omega} |\mu(x) - \nu(x)|.$$

$\diamond$

Remark 18.7. Probability distributions μ and ν are far apart in total variation distance if there is a "bad event" A such that μ and ν measure A differently.

Remark 18.8. Recall that a *stopping time* is a rule which tells a stochastic process to "stop" depending on the current value of the process.

Definition 18.9. Define the stopping time τ as a **strong stationary time** for X_t, $t \geq 0$ if $X_{\tau+1} \sim \text{unif}(S_D)$, and $X_{\tau+1}$ is independent of τ. $\diamond$

Lemma 18.10. *Let Q be a probability distribution on a finite group G inducing an irreducible and aperiodic Markov chain with transition probabilities $Q(\pi^{-1} \circ \sigma)$ from π to σ. Let τ be a strong stationary time for Q and U the uniform distribution. Then*

$$\|Q^\tau - U\|_{TV} \leq \mathbb{P}\left[\tau > \ell\right]$$

for all $\ell \geq 0$

Remark 18.11. The hypotheses irreducible and aperiodic may not be strictly necessary, but occur here because both are common in theorems about Markov chains.

Proof. For any $A \subset G$

$$
\begin{aligned}
Q^\ell(A) &= \mathbb{P}\left[X_\ell \in A\right] \\
&= \sum_{j \leq \ell} \mathbb{P}\left[X_\ell \in A, \tau = j\right] + \mathbb{P}\left[X_\ell \in A, \tau > \ell\right] \\
&= \sum_{j \leq \ell} U(A)\mathbb{P}\left[\tau = j\right] + \mathbb{P}\left[X_\ell \in A \mid \tau > \ell\right]\mathbb{P}\left[\tau > \ell\right] \\
&= U(A) + (\mathbb{P}\left[X_\ell \in A \mid \tau > \ell\right] - U(A))\mathbb{P}\left[\tau > \ell\right] /
\end{aligned}
$$

Because $|\mathbb{P}\left[X_\ell \in A \mid \tau > \ell\right] - U(A)| \leq 1$ in any case

$$\|Q^\tau - U\|_{TV} \leq \mathbb{P}\left[\tau > \ell\right].$$

$\square$

Lemma 18.12. *Sample uniformly with replacement from an urn with n balls. Let V be the number of draws required until each ball has been drawn at least once. Then*

$$\mathbb{P}\left[V > n\log n + cn\right] \le e^{-c}$$

for $c \ge 0$ and $n \ge 1$.

Remark 18.13. The lemma statement is another formulation of the coupon collectors problem. The usual formulation has n different types of coupons or prizes in a cereal box. On each draw, one obtains a coupon or prize equally likely to be any one of the n types. The goal is to find the expected number of coupons one needs to gather before obtaining a complete set of at least one of each type.

Proof. Let $m = n\log n + cn$. For each ball b let A_b be the event "ball b not drawn in the first m draws". Then

$$\mathbb{P}\left[V > m\right] = \mathbb{P}\left[\bigcup_{b=1}^{n} A_b\right] \le \sum_{b=1}^{n} \mathbb{P}\left[A_b\right] = n\left(1 - \frac{1}{n}\right)^m \le n e^{-m/n} = e^{-c}.$$

See the exercises for a proof of the second inequality. $\square$

For simplicity in what follows, set $d_P(\ell) = \|P^\ell - U\|_{TV}$. Then $d_P(\ell)$ measures how close ℓ repeated shuffles get the deck to being shuffled according to the uniform density.

Theorem 18.14. *For the TTRS shuffle, and for $\epsilon < 1$*

(1) $d_P(n\log n + n\log \epsilon^{-1}) \le \epsilon$ for n sufficiently large, and

(2) $d_P(n\log n - n\log(C\epsilon^{-1})) \ge 1 - \epsilon$ for n sufficiently large.

Proof. (1) Theorem 18.4 shows that τ_{top}, the first time that the original bottom card has come to the top and been inserted into the deck is a strong uniform time for the TTRS.

(2) The goal is to show that τ_{top} has the same distribution as V in Lemma 18.12. Then the upper bound follows from Lemma 18.12 and Lemma 18.10.

(3) Write

$$\tau_{\text{top}} = \tau_1 + (\tau_2 - \tau_1) + \cdots + (\tau_{D-1} - \tau_{D-2}) + (\tau_{\text{top}} - \tau_{D-1})$$

where τ_i is the time until card i is placed under the original bottom card.

(4) When exactly i cards are under the original bottom card b, the chance that the current top card is inserted below b is $\frac{i+1}{n}$ and hence the random variable $(\tau_{i+1} - \tau_i)$ has geometric distribution

$$\mathbb{P}\left[(\tau_{i+1} - \tau_i) = j\right] = \frac{i+1}{n}\left(1 - \frac{i+1}{n}\right)^{j-1}$$

for $j \ge 1$.

(5) The random variable V in Lemma 18.12 can be written as

$$V = (V - V_{n-1}) + (V_{n-1} - V_{n-2}) + \cdots + (V_2 - V_1) + V_1$$

where V_i is the number of draws required until i distinct balls have been drawn at least once.

(6) After i distinct balls have been drawn, the chance that a draw produces a not-previously-drawn ball is $\frac{n-i}{n}$. So $V_i - V_{i-1}$ has distribution

$$\mathbb{P}\left[V_i - V_{i-1} = j\right] = \frac{n-i}{n}\left(1 - \frac{n-i}{n}\right)^{j-1}$$

for $j \geq 1$.

(7) Comparing, the corresponding terms $(\tau_{i+1} - \tau_i)$ and $V_{n-i} - V_{(n-i)-1}$ have the same distribution. Since the summands in each sum are independent, it follows that the sums τ and V have the same distribution, as required.

(8) To prove the lower bound, fix j and let A_j be the set of configurations of the deck such that the bottom j original cards stay in their original relative order with the remaining top cards interspersed in any order. There are $\binom{D}{j} = \frac{D!}{j!(D-j)!}$ positions to put these j cards in order, then $(D-j)!$ arrangements for cards in the top portion of the deck, out of $D!$ total arrangements of, so the uniform distribution probability of A_j is $U(A_j) = \frac{1}{j!}$.

(9) Let $\ell(n) = n\log n - c_n n$ where $c_n \to \infty$. The goal is to show $P^{\ell(n)}(A_j) \to 1$ as $n \to \infty$ for fixed j. Then $d_P(\ell(n)) \geq \sup_j\{P^{\ell(n)}(A_j) - U(A_j)\} \to 1$ as $n \to \infty$ for fixed j, establishing the lower bound.

(10) To prove $P^{\ell(n)}(A_j) \to 1$ as $n \to \infty$, note $P^{\ell(n)}(A_j) \geq \mathbb{P}\left[\tau_{\text{top}} - \tau_{j-1} > \ell(n)\right]$ because $\tau - \tau_{j-1}$ is distributed as the time for the card initially jth from the bottom to come to the top and be inserted. If this has not happened by time $\ell(n)$, then the original bottom j cards must still be in their relative order at time ℓ.

(11) It suffices to show that $\mathbb{P}\left[\tau_{\text{top}} - \tau_{j-1} \leq \ell\right] \to 0$ as $n \to \infty$ for fixed j. This follows from Chebyshev's inequality. Note that

$$\mathbb{E}\left[(\tau_{i+1} - \tau_i)\right] = \frac{n}{i+1},$$
$$\text{Var}\left[(\tau_{i+1} - \tau_i)\right] = \left(\frac{n}{i+1}\right)^2\left(1 - \frac{i+1}{n}\right).$$

(12) Using the Euler-Maclaurin formula $H_\nu = \log\nu + \gamma + \frac{1}{2\nu} + \epsilon_\nu$ for the harmonic numbers

$$\mathbb{E}\left[(\tau_{\text{top}} - \tau_j)\right] = \sum_{\nu=j}^{n-1}\frac{n}{\nu+1} = n\log n + O(n)$$

where

$$n\left(\log j - \frac{1}{2j} - \frac{1}{8j^2}\right) \leq O(n) \leq n\left(\log j + \frac{1}{2} - \frac{1}{8n^2}\right).$$

(13) Likewise

$$\text{Var}\left[(\tau_{\text{top}} - \tau_j)\right] = \sum_{\nu=j}^{n-1} \left(\frac{n}{\nu+1}\right)^2 \left(1 - \frac{\nu+1}{n}\right)$$

$$= n^2 \sum_{\nu=j}^{n-1} \frac{1}{(\nu+1)^2} - \sum_{\nu=j}^{n-1} \frac{n}{\nu+1}$$

$$= n^2 \sum_{\nu=j}^{n-1} \frac{1}{(\nu+1)^2} - (n\log n + O(n)) \qquad = O(n^2).$$

(14) Chebyshev's Inequality says the probability that a random variable differs from the mean by more than a quantity a is less than the variance divided by a^2:

$$\mathbb{P}\left[|Z - \mathbb{E}\left[Z\right]| \geq a\right] \leq \frac{\text{Var}\left[Z\right]}{a^2}.$$

Applied to the event $[\tau_{\text{top}} - \tau_{j-1} \leq k]$, the mean is increasing as $O(n\log n)$, so the distance of $\mathbb{E}\left[\tau_{\text{top}} - \tau_{j-1}\right]$ from k is increasing as $O(n\log n)$. The variance is increasing as $O(n^2)$, so the upper estimate of probability of $\mathbb{P}\left[\tau_{\text{top}} - \tau_{j-1} \leq k\right]$ from Chebyshev's Inequality is decreasing as

$$\frac{O(n^2)}{O(n\log n)^2} = O((\log n)^{-2}).$$

Thus $\mathbb{P}\left[\tau_{\text{top}} - \tau_{j-1} \leq k\right] \to 0$ as $n \to \infty$.

$\square$

Remark 18.15. The strong stationary time property of τ_{top} played no role in establishing the lower bound. The proof gets lower bounds by guessing some set A for which $P^k(A) - U(A)$ should be large and then using

$$d_{\text{top}}(k) = \|P^k - U\|_{\text{TV}} \geq \left|P^k(A) - U(A)\right|.$$

Note that $n\log n + n\log \epsilon^{-1} = n\log n(1 + o(1))$ and $n\log n - n\log \epsilon^{-1} = n\log n(1 - o(1))$. This gives the sense that $n\log n$ shuffles is about the right number of shuffles needed to bring the deck close to being uniformly shuffled. This gives a cut-off phenomenon, that is $n\log n$ is a critical number of shuffles such that $d_P(n\log n + o(n)) \approx 0$ but $d_P(n\log n - o(n)) \approx 1$. The distance from the stationary density changes abruptly at some value, see Figure 3.

Remark 18.16. Note that this is quite different from the asymptotics of $d_P(n) = \|P^n - U\|_{TV}$. Perron-Frobenius theory says $d_P(n) \sim a\lambda^n$ where λ is the second largest eigenvalue, but the long-time asymptotics miss the cut-off.

18.5. The Riffle Shuffle

A more realistic model of shuffling a deck cards is the commonly used **riffle shuffle**, occasionally also called the "dovetail shuffle". This method of randomizing the card order separates the deck into two parts and then leafs through each parts with the thumbs so the cards intermix. The riffle shuffle is sometimes called the GSR shuffle since Gilbert and Shannon and independently Reeds first analyzed it. As a
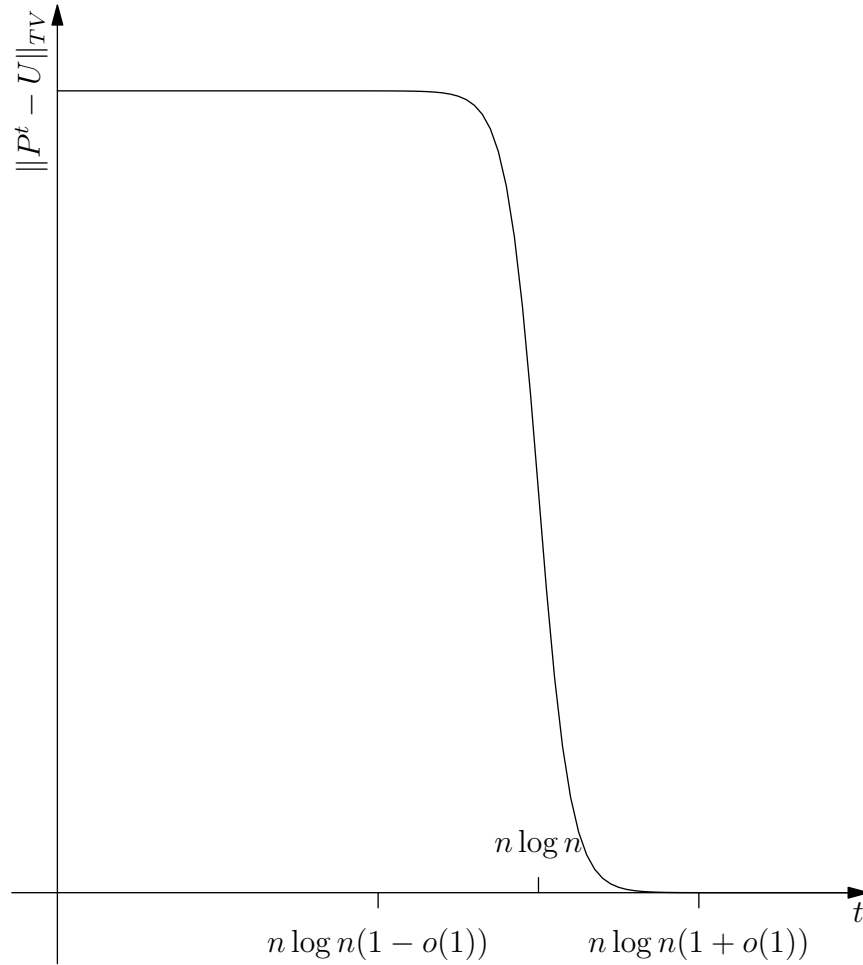
Figure 3. Schematic graph of the cut-off phenomenon for the Total Variation distance of the Markov chain distribution from the uniform distribution as a function of the number of steps.

more mathematical description, first cut the deck randomly into two packets, one containing ℓ cards and the other containing $n - \ell$ cards. Choose the number of cards cut, ℓ, according to the binomial density, meaning that the probability of the cut occurring after ℓ cards is exactly $\frac{1}{2^n} \binom{n}{\ell}$.

Once the deck is cut into two packets, interleave the cards from each packet in any possible way, such that the cards from each packet keep their own relative order. This means the cards originally in positions $1, 2, 3, \ldots, \ell$ must still be in the same order after shuffling, even if there are other cards in between. The same goes for cards originally in positions $\ell + 1, \ell + 2, \ldots, n$. This requirement is quite natural, considering how a person shuffles two packets of cards, one in each hand. The cards from the left hand must still be in the same relative order in the shuffled deck, no matter how they interleave with the cards in the other packet, because the cards

i	π_i
1	2
2	4
3	5
4	7
5	1
6	3
7	6
8	8
9	9
10	10

Figure 4. A riffle shuffle on a 10-card deck cut into a top packet of 4 cards and bottom packet of 6 cards.

drop in order while shuffling. The same goes for the cards in the right hand. See Figure 4 for an illustration of a riffle shuffle on a 10-card deck.

Choose among all possible interleavings uniformly with ℓ locations among n places for the first packet, fixing the locations for the cards of the other packet. This is the well-known "stars and bars" counting argument, with the first packet playing the role of the "stars", the second packet the "bars" creating $\binom{n}{\ell}$ possible interleavings. Call this the "uniform interleaving" method. With uniform choice, this means the probability of any one interleaving has probability $1/\binom{n}{\ell}$ of occurring. Hence the probability of any particular cut, followed by any particular interleaving is $\frac{1}{2^n}\binom{n}{k} \cdot 1/\binom{n}{\ell} = \frac{1}{2^n}$. Note that this probability has no information about the cut or the interleaving. The density on possible cuts and interleaving is uniform.

A special case of the riffle shuffle is the **perfect shuffle**, also known as the **faro shuffle**, with two equal packets completely interleaved, one card from each hand following one card from the other hand. A perfect shuffle is easy to describe but difficult to perform, except for practiced magicians. The TTRS is also a special case of the riffle shuffle with $k = 1$ occurring with probability $n/2^n$.

A second description of the riffle shuffle is useful for computer scripts simulating the shuffle, see the example in the Scripts section. Begin by choosing a left stack of cards of size ℓ from the top of the deck according to the binomial distribution $\frac{1}{2^n}\binom{n}{\ell}$. This creates a right stack of size $n - \ell$. Then the cards are dropped from each stack into the shuffled deck with probability proportional to the stack size. Thus, the probability that a card is first dropped from the left stack is ℓ/n. If this happens, the probability that the next card is dropped from the left stack is $(\ell - 1)/(n - 1)$ and so on. Call this the "left-right drop" method.

Lemma 18.17. *The "uniform interleaving" and "left-right drop" methods yield the same probability distribution of shuffles.*

Proof. Suppose ℓ cards have been cut off to make the two stacks. For the "uniform interleaving", a given shuffle is specified by a sequence $D_1, D_2, \ldots, D_n$ where each D_i is an L or a R with ℓ of the D_is is L with probability $1/\binom{n}{\ell}$.

Table 1. Probability distribution for a riffle shuffle on a 3 card deck.

π	[123]	[213]	[231]	[132]	[312]	[321]
$R(\pi)$	$\frac{1}{2}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	0.

For the "left-right drop", the sequence $D_1, D_2, \ldots, D_n$ where each D_i is an L or a R with ℓ of the D_is is L occurs with probability (after commuting the probability factors) $\ell!(n - ell)!/n!$.

The two probability distributions are the same. $\qquad\qquad\square$

The uniform density on the set of cuts and interleavings now induces in a natural way a density on the set of permutations. Call the density a *riffle shuffle* and denote it by R. That is, $R(\pi)$ is the sum of probabilities of each cut and interleaving that gives the rearrangement of the deck corresponding to π. In short, the chance of any arrangement of cards occurring under riffle shuffling is the proportion of cuts and interleavings that give that arrangement.

Example 18.18. Consider the riffle shuffle on a 3-card deck as a Markov chain. The probability distribution for R is in Table 1. To obtain the entries in the transition probability matrix, systematically go through the possible cuts and interleavings. Cutting three cards into the left packet, and none in the right packet, the only possible interleaving trivially leaves the deck unchanged. With a cut into 2 cards on the left, 1 card on the right, one interleaving drops the right packet card on the bottom, the left packet cards as the top 2, leaving the deck unchanged. Two other interleavings move the card in the right packet to the middle or the top. The other two cuts are symmetric to the cuts described above, so 4 of the 8 cuts and interleavings keep the deck in the original order. However, one shuffle each moves the formerly bottom card labeled 3 to the middle or top position, leaving cards 1 and 2 in that order in the shuffled deck. A single riffle shuffle cannot reverse the order of the deck.

For the entries in the transition probability matrix, consider the computation for a typical element say $P_{\pi,\sigma}$ with $\pi = [213]$ and $\sigma = [132]$. Then $\pi^{-1} = [213]$ and $\pi^{-1} \circ \sigma = [231]$. Now $R([231]) = \frac{1}{8}$, giving $P_{[213][132]} = \frac{1}{8}$ in the transition probability matrix.

The full transition probability matrix under this ordering of the permutations is

$$
\begin{array}{c c}
& \begin{array}{cccccc} [123] & [213] & [231] & [132] & [312] & [321] \end{array} \\
\begin{array}{c} [123] \\ [213] \\ [231] \\ [132] \\ [312] \\ [321] \end{array} &
\left(\begin{array}{cccccc}
\frac{1}{2} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 \\
\frac{1}{8} & \frac{1}{2} & \frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{8} \\
\frac{1}{8} & \frac{1}{8} & \frac{1}{2} & 0 & \frac{1}{8} & \frac{1}{8} \\
\frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{2} & \frac{1}{8} & \frac{1}{8} \\
\frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{2} & \frac{1}{8} \\
0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{2}
\end{array}\right).
\end{array}
$$

Although in this case, the $n = 3$ riffle shuffle, the matrix is symmetric, this is in general not true, the riffle shuffle with deck sizes greater than 3 is nonsymmetric, see the exercises. $\qquad\qquad\triangledown$

First note that the Markov chain for riffle shuffling is regular, that is, any permutation has a positive probability of appearing after sufficiently many shuffles, see the exercises. In fact, any number of shuffles greater than $\log_2 n$ will do. Since the riffle shuffle Markov chain is regular, there is a unique stationary density, which is the uniform density on S_n.

Starting with the identity ordering, the density of the permutations after 7 riffle shuffles is the first row of P^7. With matrix multiplication, the density is nearly uniform. In fact,

$$P^7 = \begin{pmatrix} 0.17059 & 0.16666 & 0.16666 & 0.16666 & 0.16666 & 0.16278 \\ 0.16666 & 0.17059 & 0.16666 & 0.16666 & 0.16278 & 0.16666 \\ 0.16666 & 0.16666 & 0.17059 & 0.16278 & 0.16666 & 0.16666 \\ 0.16666 & 0.16666 & 0.16278 & 0.17059 & 0.16666 & 0.16666 \\ 0.16666 & 0.16278 & 0.16666 & 0.16666 & 0.17059 & 0.16666 \\ 0.16278 & 0.16666 & 0.16666 & 0.16666 & 0.16666 & 0.17059 \end{pmatrix}.$$

That is, 7 shuffles of the 3-card deck gets close to the stationary density, which turns out to be the uniform density.

18.6. Probability of a Permutation Under Riffle Shuffle

Define a **rising sequence** of a permutation as a maximal increasing subsequence within the permutation, potentially with gaps in position. In more detail applied to shuffled decks of cards, say x is a particular card from the deck. After the position of x, look for the card labeled $x + 1$. If found, repeat the procedure, looking after the $x + 1$ card for the $x + 2$ card. Keep going in this manner until it is not possible to find the next card adjacent. Now go back to the original card x and look for the $x - 1$ card immediately prior and so on. When done, the subsequence of the permutation containing x is a rising sequence. A little thought shows that a deck breaks down as a disjoint union of its rising sequences, since the union of any two consecutively increasing subsequences with that element is a rising subsequence.

Example 18.19. Suppose that a permutation of a 8-card deck is 45162378. Start with any card, say 3. Look for 4 and do not find it. Look before the 3 and find 2 and before it, with a gap, find 1. So one of the rising sequences of this permutation is 123. Now start again, this time with 6. After a gap, find 7 and then after it 8. Before 6 with a gap find 5 and then 4. So another rising sequence is 45678. This accounts for all cards and the deck has only two rising sequences. Writing the sequence as $45_16_{23}78$, offsetting the two subsequences, makes this clear.   ▽

Example 18.20. The riffle shuffle in Figure 4 has two rising sequences, $\pi(1) < \pi(2) < \pi(3) < \pi(4)$ and $\pi(5) < \pi(6) < \pi(7) < \pi(8) < \pi(9) < \pi(10)$.   ▽

Example 18.21. Note that $45_16_{23}78$ is a possible result of a riffle shuffle. Here the cut must divide the deck into two packets such that the length of each is the same as the length of the corresponding rising sequence. So if the deck started in the natural order and the deck is cut into 123 on left and 45678 on the right, then the shuffle interleaves by dropping on the bottom 8, then 7, then 3, then 2, then 6, then 1, then 5 and 4, thus obtaining the given top down order through riffling.   ▽

In general, a permutation π of n cards in original order made by a riffle shuffle will have exactly 2 rising sequences (unless it is the identity with exactly 1 rising sequence). This is a consequence of the definition of a riffle shuffle as a cut and an interleaving. Conversely any permutation of n cards with 1 or 2 rising sequences can be obtained by a physical shuffle. The lengths of the rising sequences define the size of the cut, the gaps in a subsequence define the interleavings. Therefore a mathematical definition of a riffle shuffle can be made as "a permutation with 1 or 2 rising sequences." Suppose c cards are cut off the top. Then there are $\binom{n}{c}$ possible riffle shuffles, (one of which is the identity shuffle). As in Figure 4, after the shuffle, the red and black cards form a binary n-tuple with c red cards, there are $\binom{n}{c}$ such n-tuples, one of which is the original order. The total number of possible riffle shuffles is

$$1 + \sum_{c=0}^{n}\left(\binom{n}{c} - 1\right) = 2^n - n.$$

The next goal is to get similar results about what happens after multiple riffle shuffles. This can be done by considering a-shuffles. A a-**shuffle** is another probability density on S_n. Let a be any positive integer. Cut the deck into a packets of nonnegative sizes $m_1, m_2, \ldots, m_a$ with $m_1 + \cdots + m_a = n$ but some of the m_i may be zero. The probability of this particular packet structure is given by the multinomial density:

$$\frac{1}{a^n}\binom{n}{m_1, m_2, \ldots, m_a}.$$

Interleave the cards from each packet in any way, so long as the cards from each packet keep the relative order among themselves. With a fixed packet structure, consider all interleavings equally likely. Count the number of such interleavings as the number of ways of choosing among n positions in the deck, m_1 places for things of the first type, m_2 places for things of the second type and so on. The count is the multinomial coefficient

$$\binom{n}{m_1, m_2, \ldots, m_a}.$$

Hence the probability of a particular rearrangement, i.e. a cut of the deck and an interleaving is

$$\frac{1}{a^n}\binom{n}{m_1, m_2, \ldots, m_a} \bigg/ \binom{n}{m_1, m_2, \ldots, m_a} = \frac{1}{a^n}.$$

So it turns out that each combination of a particular cut into packets and an interleaving is equally likely, just as in the riffle shuffle. The induced density on the permutations leading to the cuts and shuffles is then called the a-shuffle, with notation R_a. The riffle shuffle is just the 2-shuffle, so $R_2 = R$.

An equivalent description of the a-shuffle begins the same way, by cutting the deck into packets multinomially. Then drop cards from the bottom of the packets, one at a time, such that the probability of choosing a particular packet to drop is proportional to the relative size of that packet compared to the number of all cards in the packets. The proof of this description is exactly analogous to the $a = 2$ case.

A third equivalent description is cutting the deck multinomially into packets of size $m_1, m_2, \ldots m_n$ and riffling m_1 and m_2 together, meaning choose uniformly

among all interleavings that the keep the relative order of each packet, then riffling the resulting pile with m_3, then riffling that resulting pile with m_4 and so on.

It turns out that when performing a single a-shuffle, the probability of achieving a particular permutation π does not depend on the detailed information in π, but only on the number of rising sequences that π has. In other words, the permutations [12534], [34512], [51234], and [23451] have the same probability under any a-shuffle, since each has exactly two rising sequences.

A useful code based on n-digit base-a numbers specifies how to make a particular a-shuffle. Here a "shuffle" indicates a particular way of rearranging the deck, not the probability density on all such rearrangements. Let A be an n-digit base-a number. Count the number of 0s in A, this will be the size of the first packet m_1 in the a-shuffle. Then m_2 is the number of 1s in A and so on up to m_a, the number of $(a-1)$s. This cuts the deck into a packets. Now take the beginning packet of cards of size m_1. Envision placing these cards on top of all the 0 digits keeping their order as a rising sequence. Do the same for the next packet of size m_2, placing them on the 1s. Continue up through the $(a-1)$s. This particular way of rearranging the cards will then be the particular cut and the interleaving corresponding to A. Note that the number of such encodings is a^n.

Example 18.22. Let an 8-card deck start in natural order. Let $A = 23004103$ be the code for a particular 5-shuffle of the 8-card deck. The code has three 0s, one 1, one 2, two 3s and one 4. Thus $m_1 = 3$, $m_2 = 1$, $m_3 = 1$, $m_4 = 2$ and $m_5 = 1$. So cut the deck into $123|4|5|67|8$. We put 123 where the 0s are in A, 4 where the 1 is, 5 where the 2 is, 67 where the 3s are, and 8 where the 4 is. Then get a shuffled deck of 56128437 after applying A to the natural order. ∇

This code gives a bijective correspondence between n-digit base-a numbers and the set of all ways of cutting and interleaving an n-card deck according to the a-shuffle. In fact, putting the uniform density on the set of n-digit base-a numbers transfers to the correct uniform probability density for cutting and interleaving in an a-shuffle which means the correct density gets induced on S_n.

Theorem 18.23. *The probability of achieving a permutation π when doing an a-shuffle on an n-card deck is*

$$\frac{1}{a^n}\binom{n+a-r}{n},$$

where r is the number of rising sequences in π.

Proof. (1) Establish and fix where the $(a-1)$ cuts occur in an a-shuffle, then whatever permutations can actually be achieved by interleaving the cards from this cut and packet structure can be achieved in exactly one way: Just drop the cards in exactly the order of the permutation.

(2) Thus the probability of achieving a particular permutation is the number of possible ways of making cuts that could actually cause that permutation, divided by the total number of ways of making cuts and interleaving for an a-shuffle.

(3) Having r rising sequences in π determines exactly where $(r-1)$ of the cuts must have been: between pairs of consecutive cards in the naturally ordered deck such that the first card of the pair ends one rising sequence of π.

(4) This means that $(a-1)-(r-1)=a-r$ is unspecified, or that free cuts that can go anywhere.

(5) So count the number of ways of putting $a-r$ cuts among n cards. The standard "stars and bars" combinatorial argument counts

$$\binom{n+a-r}{n}$$

ways to do this, i.e. choosing n places among $n+(a-r)$. This is the numerator of the probability.

(6) The denominator is the number of possible ways to cut and interleave for an a-shuffle. The encoding of the shuffles as the number of n-digit base a numbers gives a^n ways to do this.

□

Remark 18.24. Compare the results of this theorem with the entries of the matrix

$$
\begin{array}{c}
 \\
[123] \\
[213] \\
[231] \\
[132] \\
[312] \\
[321]
\end{array}
\begin{array}{c}
\begin{array}{cccccc}
[123] & [213] & [231] & [132] & [312] & [321]
\end{array} \\
\left(
\begin{array}{cccccc}
\frac{1}{2} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 \\
\frac{1}{8} & \frac{1}{2} & \frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{8} \\
\frac{1}{8} & \frac{1}{8} & \frac{1}{2} & 0 & \frac{1}{8} & \frac{1}{8} \\
\frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{2} & \frac{1}{8} & \frac{1}{8} \\
\frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{2} & \frac{1}{8} \\
0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{2}
\end{array}
\right).
\end{array}
$$

Application of the theorem is easier than counting the shuffles carrying one permutation to another.

18.7. Rate of Convergence to Stationary

Recall the modeling of card shuffling as a Markov chain. The initial state X_0 is the identity permutation, that is cards in the initial deck order numbered 1 to n, with probability 1. A particular riffle shuffle, one of a whole family of riffle shuffles, is chosen with a probability distribution from the family. This probability distribution then induces a transition probability from state to state. Going from card order π to σ is the same as composing π with the permutation $\pi^{-1}\circ\sigma$. Now identify shuffles as functions on $\{1,\ldots n\}$ to $\{1,\ldots n\}$, that is, permutations. Since a particular riffle shuffle is one of a whole family of riffle shuffles, chosen with a probability distribution Q from the family, the transition probabilities are $p_{\pi\sigma}=\mathbb{P}\left[X_{t+1}=\sigma\mid X_t=\pi\right]=Q(\pi^{-1}\circ\sigma)$.

The rate of convergence of X_t to the stationary density, measured by the total variation distance or some other metric, is determined by the eigenvalues of the transition matrix. The entries of P^k are the probabilities of certain permutations

being achieved under k riffle shuffles. These probabilities are of the form

$$\frac{1}{2^{nk}}\binom{2^k + n - r}{n},$$

for the probability of a permutation with r rising sequences being achieved after k riffle shuffles.

Mann [**Man94**] asserts that the eigenvalues of the transition probability matrix for a single riffle shuffle are exactly $1, \frac{1}{2}, \frac{1}{4}, \dots \frac{1}{2^n}$, see the exercises. The second largest eigenvalue determines the rate of convergence to the stationary distribution. For riffle shuffling, this eigenvalue is $\frac{1}{2}$. Once the variation distance gets to the cutoff, it decreases by a factor of approximately $\frac{1}{2}$ with each shuffle.

Example 18.25. The riffle shuffle matrix for the deck of three cards is

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & 0 \\ \frac{1}{8} & \frac{1}{2} & \frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{2} & 0 & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{2} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{2} & \frac{1}{8} \\ 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{2} \end{pmatrix}.$$

The eigenvalues of this matrix are $1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}$. ∇

A comparison of the total variation distance from uniform after n shuffles for simulations of the TTRS and the riffle shuffle for a 10-card deck is in Figure 5. Both show features of the cut-off phenomenon. Naturally, the riffle shuffle converges to uniform faster since it mixes more cards at each step. Note that 7 riffle shuffles suffice to thoroughly mix a 10-card deck.

18.8. Chapter Ending Answer

For card players, shuffling the deck to remove any order is essential so that cards dealt from the deck come "at random", that is, in an order uniformly distributed over all possible deck orders. For the Top-to-Random-Shuffle 7 shuffles of the 3-card deck gets close to the uniform density, which turns out to be the stationary density. With a riffle shuffle on a standard deck, 7 shuffles gets close to the uniform density.

18.9. Chapter Summary

Key Concepts.

(1) Card deck shuffles are a family of possible re-orderings with probability distributions, leading to transition probabilities, and thus Markov processes. The most well-studied type of shuffle is the riffle shuffle and that is the main focus here.

(2) Going from card order π to σ is the same as composing π with the permutation $\pi^{-1} \circ \sigma$. Now identify shuffles as functions on $\{1, \dots n\}$ to $\{1, \dots n\}$, that is, permutations. Since a particular shuffle is one of a whole family of shuffles, chosen with a probability distribution Q from the family, the transition
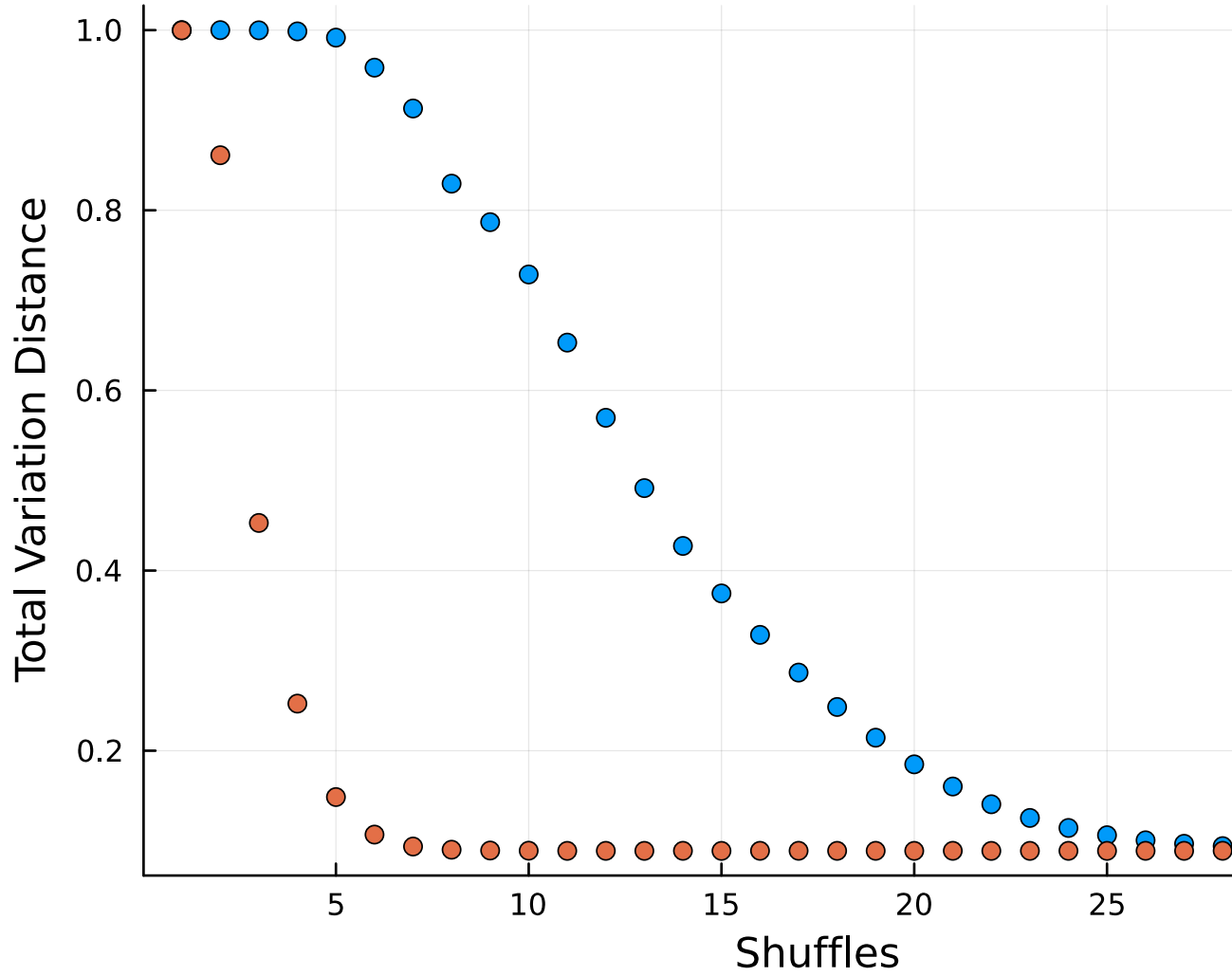
Figure 5. Comparison of the total variation distance from uniform after n shuffles for the TTRS shuffle in blue and the riffle shuffle in red.

probabilities are

$$p_{\pi\sigma} = \mathbb{P}\left[X_{t+1} = \sigma \mid X_t = \pi\right] = Q(\pi^{-1} \circ \sigma).$$

(3) The identification of shuffles or operations with permutations gives a probability distribution on S_D.

(4) A **Top-to-Random-Shuffle**, takes the top card from a stack of n cards and inserts it in the gap between the $(\ell-1)$th card and the ℓth card in the deck, where ℓ is chosen randomly.

(5) The Top-To-Random-Shuffle demonstrates the cut-off phenomenon for the Total Variation distance of the Markov chain distribution from the uniform distribution as a function of the number of steps.

(6) One realistic model of shuffling a deck of cards is the **riffle shuffle**.

(7) The set of cuts and interleavings in a riffle shuffle induces in a natural way a density on the set of permutations. Call this a **riffle shuffle** and denote it by R. That is, $R(\pi)$ is the sum of probabilities of each cut and interleaving that gives the rearrangement of the deck corresponding to π.

(8) 7 shuffles the of 3-card deck gets very close to the uniform density, which turns out to be the stationary density.

(9) The probability of achieving a permutation π when doing an a-shuffle is

$$\frac{1}{a^n} \binom{n+a-r}{n},$$

where r is the number of rising sequences in π.

(10) The eigenvalues of the transition probability matrix for a riffle shuffle are 1, $\frac{1}{2}$, $\frac{1}{4}$ and $\frac{1}{2^n}$. The second largest eigenvalue determines the rate of convergence to the stationary distribution. For riffle shuffling, this eigenvalue is $\frac{1}{2}$.

(11) For a finite, irreducible, aperiodic Markov chain Y_t distributed as Q^t at time t and with stationary distribution π, and τ is a strong stationary time, then

$$\|Q^\tau - \pi\|_{TV} \leq \mathbb{P}\left[(\,|\,\tau \geq t).\right.$$

(12) Set $d_n(t) = \|P^{\tau_{\text{top}}+1} - U\|_{TV}$. Then for $\epsilon > 0$,
  (a) $d_n(n \log n + n \log \epsilon^{-1}) \leq \epsilon$ for n sufficiently large.
  (b) $d_n(n \log n - n \log(C\epsilon^{-1})) \geq 1 - \epsilon$ for n sufficiently large.

Vocabulary.

(1) A **Top-to-Random-Shuffle**, takes the top card from a stack of n cards and inserts it in the gap between the $(k-1)$th card and the kth card in the deck.

(2) The **total variation distance** of μ from ν is

$$\|\mu - \nu\|_{TV} = \max_{A \subset \Omega} |\mu(A) - \nu(A)| = \frac{1}{2} \sum_{x \in \Omega} |\mu(x) - \nu(x)|.$$

(3) A stopping time τ_{top} is a **strong stationary time** for X_t, $t \geq 0$ if $X_{\tau_{\text{top}}+1} \sim \text{unif}(S_n)$, and $X_{\tau_{\text{top}}+1}$ is independent of τ_{top}.

(4) The **riffle shuffle** first cuts the deck randomly into two packets, one containing k cards and the other containing $n-k$ cards. Choose k, the number of cards cut according to the binomial density. Once the deck is cut into two packets, interleave the cards from each packet in any possible way, such that the cards from each packet keep their own relative order.

(5) A special case of the riffle shuffle is the **perfect shuffle**, also known as the **faro shuffle** wherein the two packets are completely interleaved.

(6) A **rising sequence** of a permutation is a maximal consecutive increasing subsequence.

(7) A a-**shuffle** is another probability density on S_n. Let a be any positive integer. Cut the deck into a packets of nonnegative sizes $m_1, m_2, \ldots, m_a$ with $m_1 + \cdots + m_a = n$ but some of the n_i may be zero. Interleave the cards from each packet in any way, so long as the cards from each packet keep the relative order among themselves. With a fixed packet structure, consider all interleavings equally likely.

Notation.

(1) S_n – The set of states for a Markov process modeling the order of the deck, the set of permutations on n cards.

(2) Initial state X_0 – the identity permutation with probability 1

(3) π and σ – particular card orders

(4) probability distribution Q from the family of shuffles, the transition probabilities are $p_{\pi\sigma} = \mathbb{P}\left[X_{t+1} = \sigma \mid X_t = \pi\right] = Q(\pi^{-1} \circ \sigma)$.

(5) P – transition probability matrix

(6) t – epoch or time for the Markov chain

(7) k – arbitrary card or position in deck of n cards

(8) U – uniform distribution on a group, or more generally a state space

(9) Q be a probability distribution on a finite group G inducing an irreducible and aperiodic Markov chain with transition probabilities $Q(\pi^{-1} \circ \sigma)$ from π to σ

(10) τ_{top} be the first time that card n reaches the top of the deck

(11) If μ and ν are probability distributions on Ω, the **total variation distance** of μ from ν is $\|\mu - \nu\|_{TV}$

(12) V – the number of draws required until each ball has been drawn at least once

(13) set $d_P(n) = \|P^n - U\|_{TV}$

(14) τ_i – the time until card i is placed under the original bottom card

(15) V_i – the number of draws required until i distinct balls have been drawn at least once

(16) A_j – the set of configurations of the deck such that the bottom j original cards stay in their original relative order. Plainly $U(A_j) = \frac{1}{j!}$

(17) $k = k(n) = n \log n - c_n n$ where $c_n \to \infty$

(18) cut the deck randomly into two packets, one containing k cards and the other containing $n - k$ cards. Choose the number of cards cut, k, according to the binomial density

(19) Call the density a *riffle shuffle* and denote it by R. That is, $R(\pi)$ is the sum of probabilities of each cut and interleaving that gives the rearrangement of the deck corresponding to π

(20) a – number of packets of nonnegative sizes $m_1, m_2, \ldots, m_a$ with $m_1 + \cdots + m_a = n$ but some of the m_i may be zero

(21) R_a – induced density on the permutations leading to the cuts and shuffles is then called the a-shuffle with notation

18.10. Sources

This section is adapted from [**Man94**]. The exercise on the "geometric" algorithm is adapted from [**AD86**].

18.11. Algorithms and Scripts

Algorithm.

1 permutations(n) Generate permutation of $1:n$ in lexicographic order
2 **return** *List of all n! permutations in lexicographic order* `/* The R`
    `implementation here uses recursion, and an index shift with`
    `an increment by 1 of some prior entries, returning a`
    `matrix.  Any other algorithm may be used.`   `*/`
3 rowmatch(v, m) Find index of permutation v in list m
4 **return** index
  `/* In the R implementation here, this is the code bottle neck.`
  `*/`
5 TTRS(v) Find length of v
6 Generate random selection of position in 1 to length
7 Insert $v[1]$ at random position, shifting rest of v
8 **return** *Top-to-random-shuffle of v*

9 **riffle** v
10   Find length of v
11   Cut deck v into two stacks by binomial distribution
12   **for** $i \leftarrow 1$ **to** *deck size* **do**
13   G
14   **end**
15   enerate Bernoulli random variate with parameter proportional to
     current size of left stack
16   **if** *true* **then**
17   Drop one card from left stack
18   **end**
19   **else**
20   Drop one card from right stack
21   **end**
22   **return** *riffle-shuffle of v*

23 Initialize trials factor
  `/* At the end, each permutation should occur about this factor`
                                   `*/`
24 Initialize deck size d
25 Set the number of shuffles in each trial $1.5 \cdot d \log(d)$
  `// The factor 1.5 is experimentally adequate for complete`
    `shuffling`
26 Make list of all permutations
27 Initialize matrix to hold occurrences of each permutation
28 Set the number of trials to be trials factro time number of permutations
29 **for** $t \leftarrow 1$ **to** *number trials* **do**
30   **for** $s \leftarrow 1$ **to** *number of shuffles* **do**
31   Initialize the deck
32   Shuffle the deck with TTRS or **riffle**
33
34   D
35   etermine which permutation is the new shuffle
36   For that shuffle and permutation, increment the record of
         occurrences
37   **end**
38 **end**

Scripts.

R: R script for TTRS.

```r
permutations <- function(n){
# from  https://stackoverflow.com/questions/11095992/generating-all-
    distinct-permutations-of-a-list-in-r
    if(n==1){
        return(matrix(1))
    } else {
        sp <- permutations(n-1)
        p <- nrow(sp)
        A <- matrix(nrow=n*p,ncol=n)
        for(i in 1:n){
            A[(i-1)*p+1:p,] <- cbind(i,sp+(sp>=i))
        }
        return(A)
    }
}

row_match <- function(v, m) {
    # find row number of vector v in matrix m, slow spot in code
    which(apply(m, 1, function(x) return(all(x == v))))
}

ttrs <- function(v) {
    len <- length(v)

    r <- sample(1:len, 1)
    if (r == 1) {
        v <- v                     #Do nothing
    } else if (r == len) {
        v <-  c( v[2:len], v[1]) # Move to bottom
    } else {
        v <- c( v[2:r], v[1], v[(r+1):len] )
    }
    v
}

n <- 20 #trials factor
deckSize <- 5
maxNumTTRSShuffles <- floor(1.5 * deckSize * log(deckSize))
# factor of 1.5 derived from experimentation as being adequate
nperms <- factorial(deckSize)
perms <- permutations(deckSize)

record <- matrix(0, nperms, maxNumTTRSShuffles)

totalVar <- rep(0, maxNumTTRSShuffles)

trials = n * factorial(deckSize)
# after adequate shuffling, should have about n for each permutation

for (t in 1:trials) {
    v = 1:deckSize
    for (shuffle in 1:maxNumTTRSShuffles) {
        v <- ttrs(v)
        prow <- row <- row_match(v, perms)
        record[prow, shuffle] <- record[prow, shuffle] + 1
    }
}

totalVar <- (1/2) * colSums(abs(record/trials - 1/factorial(deckSize) ) )
```

.

R: R script for Riffle.

```r

permutations <- function(n){
```

```r
 3 # from   https://stackoverflow.com/questions/11095992/generating-all-
       distinct-permutations-of-a-list-in-r
 4     if(n==1){
 5         return(matrix(1))
 6     } else {
 7         sp <- permutations(n-1)
 8         p <- nrow(sp)
 9         A <- matrix(nrow=n*p,ncol=n)
10         for(i in 1:n){
11             A[(i-1)*p+1:p,] <- cbind(i,sp+(sp>=i))
12         }
13         return(A)
14     }
15 }
16
17 row_match <- function(v, m) {
18     # find row number of vector v in matrix m, slow spot in code
19     which(apply(m, 1, function(x) return(all(x == v))))
20 }
21
22 riffle <- function(v) {
23     len <- length(v)
24     vt <- numeric(len) #local temp vector to hold shuffle of v
25
26     ## cut into 2 stacks, size chosen with binomial distribution
27     rL <- rbinom(1, len, 1/2)
28     rR <- len - rL
29
30     if (rL == len | rR == len) {
31         vt <- v
32         return(vt)
33     }
34
35     L <- v[c(1:rL)] #left stack
36     R <- v[c(rL+1):len] #right stack
37
38     j <- 1
39     k <- 1 #card counters
40
41     for (i in (1:len)) {
42         lr <- rbinom(1, 1, rL/(rL + rR))
43         # true == 1 if uniform variate < p = rL/(rL+rR)
44         if (lr == 1) { # drop from left stack
45             ## cat("riffle: ", L, "drop from left at ", j, "\n" )
46             vt[i] <- L[j]
47             rL <- rL-1
48             j<- j+1 #update left stack
49         } else {   # drop from right stack
50             ## cat("riffle: ", R, "drop from right at ", k, "\n")
51             vt[i] <- R[k]
52             rR <- rR -1
53             k <- k + 1 #update right stack
54         }
55     }
56     vt
57 }
58
59 n <- 10 #trials factor
60 deckSize <- 5
61 maxNumRiffleShuffles <- floor(1.5 * deckSize * log(deckSize))
62 # factor of 1.5 derived from experimentation as being adequate
63 nperms <- factorial(deckSize)
64 perms <- permutations(deckSize)
65
66 record <- matrix(0, nperms, maxNumRiffleShuffles)
67
68 totalVar <- rep(0, maxNumRiffleShuffles)
69
70 trials = n * factorial(deckSize)
71 # after adequate shuffling, should have about n for each permutation
72
```

```
73  for (t in 1:trials) {
74      v = 1:deckSize
75      for (shuffle in 1:maxNumRiffleShuffles) {
76          v <- riffle(v)
77          prow <- row <- row_match(v, perms)
78          record[prow, shuffle] <- record[prow, shuffle] + 1
79      }
80  }
81
82  totalVar <- (1/2) * colSums(abs(record/trials - 1/factorial(deckSize) ) )
```

.

Bibliography

[AB72]   Robert Ash and Richard Bishop, *Monopoly as a Markov Process*, Mathematics
           Magazine **45** (1972), 26–29.

[AB15]   Max A. Alekseyev and Toby Berger, *Solving the tower of hanoi with random
           moves*, The Mathematics of Various Entertaining Subjects (2015).

[AD86]   David Aldous and Persi Diaconis, *Shuffling cards and stopping times*, The
           American Mathematical Monthly **93** (1986), no. 5, 333–348.

[AKS93]   S. C. Althoen, L. King, and K. Schilling, *How long is a game of snakes and
           ladders*, The Mathematical Gazette **77** (1993), no. 478, 71–76.

[AR97]   Steve Abbott and Matt Richey, *Take a Walk on the Boardwalk*, The College
           Mathematical Journal **28** (1997), no. 3, 162–171.

[BDSX06]   Stephen Boyd, Persi Diaconis, Jun Sun, and Lin Xiao, *Fastest mixing Markov
           chain on a path*, The American Mathematical Monthly **113** (2006), no. 1,
           70–74, Markov chains.

[BhCHJ18] P. H. Brill, Chi ho Cheung, Myron Hlynka, and Q. Jiang, *Reversibility check-
           ing for Markov chains*, Communications on Stochastic Analysis **12** (2018),
           no. 2, 129–135.

[BHP97]   Bruce Bukiet, Elliott Rusty Harold, and José Luis Palacios, *A markov chain
           approach to baseball*, Operations Research **45** (1997), no. 1, 14–23.

[BP20]   Steve Bacinski and Timothy Pennings, *How to win at tenzi*, The College
           Mathematics Journal **51** (2020), no. 4, 242–253.

[CG92]   George Casella and Edward I. Geoerge, *Explaining the gibbs sampler*, The
           American Statistician **46** (1992), no. 3, 167–174.

[CH17]   Michael A. Carchidi and Robert L. Higgins, *Covariances between transient
           states in finite absorbing Markov chains*, The College Mathematics Journal
           48 (2017), no. 1, 42–50.

[CK77]   Thomas M. Cover and Carroll W. Keilers, *An offensive earned-run average
           for baseball*, Operations Research **25** (1977), no. 5, 729–740.

[D'A10]   John P. D'Angelo, *Baseball and Markov chains: Power hitting and power
           series*, Notices of the American Mathematical Society **57** (2010), no. 4, 490–
           495.

[Dia09]   Persi Diaconis, *The Markov chain Monte Carlo revolution*, Notices of the American Mathematical Society **46** (2009), 179–205.

[Dol]   Regina Dolgoarshinnykh, *Criticality in epidemic models*, accessed April 17, 2020.

[DS81]   P. Diaconis and M. Shahshahani, *Generating a random permutation with random tranpositions*, Z. Warsch. Verw. Gebiete **57** (1981), no. 2, 159–179.

[DS84]   Peter G. Doyle and J. Laurie Snell, *Random walks and electric networks*, The Mathematical Association of America, 1984.

[DS98]   Persi Diaconis and L. Saloff, *What do we know about the metropolis algorithm*, J. Comput. System Sci. **57** (1998), 20–36.

[DSC98]   P. Diaconis and L. Saloff-Coste, *What do we know about the metropolis algorithm ?*, Journal of Computer and System Sciences **57** (1998), no. 1, 20 – 36.

[Dur09]   Richard Durrett, *Elementary probability for applications*, Cambridge University Press, 2009.

[Eps20]   Sacha Epskamp, *Isingsampler*, January 2020, accessed December 16, 2020.

[Fel73]   William Feller, *An introduction to probability theory and its applications, volume i*, third ed., vol. I, John Wiley and Sons, 1973, QA 273 F3712.

[GS97]   Charles M. Grinstead and J. Laurie Snell, *Introduction to probability*, American Mathematical Society, 1997.

[Hay19]   Brian Hayes, *Glauber's dynamics*, http://bit-player.org/2019/glaubers-dynamics, January 2019.

[Hay21]   ———, *Three months in monte carlo*, http://bit-player.org/2021/three-months-in-monte-carlo, July 2021.

[HS09]   Karl Heinz Hoffmann and Peter Salamon, *Bounding the lumping error in Markov chain dynamics*, Applied Mathematics Letters **22** (2009), no. 9, 1471–1475.

[Jac10]   Christine Jacob, *Branching processes: their role in epidemiology.*, Int J Environ Res Public Health **7** (2010), no. 3, 1186–1204.

[JLL⁺21]   Yu Huang Jiang, Tong Liu, Zhiya Lou, Jeffrey S. Rosenthal, Shanshan Shangguan, Fei Wang, and Zixuan Wu, *The coupling/minorization/drift approach to Markov chain convergence rates*, Notices of the American Mathematical Society **68** (2021), no. 4, 471–478.

[KS60]   John G. Kemeny and J. Laurie Snell, *Finite Markov chains*, D. Van Nostrand Company, Inc., 1960.

[KST74]   J. G. Kemeny, J. L. Snell, and G. L. Thompson, *Introduction to finite mathematics*, Prentice-Hall, 1974.

[KT75]   S. Karlin and H. Taylor, *A first course in stochastic processes, second edition*, Academic Press, 1975.

[KT81]   ———, *A second course in stochastic processes*, Academic Press, 1981.

[Lar]   Guy Larangeira, *Three simple applications of Markov chains and gibbs sampling*, accessed April 7, 2020.

[Les05]   Emmanuel Lesigne, *Heads or tails: An introduction to limit theorems in probability*, Student Mathematical Library, vol. 28, American Mathematical Society, 2005.

[LPW09]   David A. Levin, Yuval Peres, and Elizabeth L. Wilmer, *Markov chains and mixing times*, American Mathematical Society, 2009.

[Man94]   Brad Mann, *How many times should you shuffle a deck of cards*, UMAP Journal **15** (1994), no. 4, 303–332.

[MM19]   David McCune and Lori McCune, *Counting your chickens with Markov chains*, Mathematics Magazine **92** (2019), no. 3, 163–172.

[Osb03]   Jason A. Osborne, *Markov chains for the risk board game revisited*, Mathematics Magazine **76** (2003), no. 2, 129–135.

[OWD20]   Jason A. Osbourne, Meoldy Wen, and Neil Dey, *Mlbdecider: A shiny app for baseball*, Notices of the American Mathematical Society **67** (2020), no. 9, 1389–1395.

[Pan]   Mark D. Pankin, *Markov chain models:theoretical background*, Accessed December 19, 2019.

[PW15]   Pamela Pierce and Robert Wooster, *Conquer the world with Markov chains*, Math Horizons **22** (2015), no. 4, 18–21.

[Ric10]   Matthew Richey, *The evolution of Markov chain monte carlo methods*, American Mathematical Monthly **117** (2010), no. 5, 383–414.

[Ros97]   Sheldon M. Ross, *Introduction to probability models*, 6th ed., Elsevier, 1997.

[Ros06]   _____, *Introduction to probability models*, 9th ed., Academic Press, 2006.

[Sch18]   Tanya Schlusser, *Mcmc and the ising model*, July 2018, accesed December 16, 2020.

[Spe17]   Giorgio Alfredo Spedicato, *Discrete time markov chains with r*, The R Journal **9** (2017), no. 2, 84–104.

[SR13]   Constantinos I. Siettos and Lucia Russo, *Mathematical modeling of infectious disease dynamics*, Virulence **4** (2013), no. 4, 295–304.

[SS93]   L. E. Sadovskii and A. L. Sadovskii, *Mathematics and sports*, Mathematical World, vol. 3, ch. 3, American Mathematical Society, 1993.

[Sti05]   David Stirzaker, *Stochastic processes and models*, Oxford University Press, 2005.

[Str16]   Daniel W. Stroock, *The new palgrave dictionary of economics*, ch. Markov Processes, pp. 1–6, Palgrave Macmillan UK, London, 2016.

[Sve68]   A. A. Svesnikov, *Problems in probability theory, mathematical statistics and theory of random functions*, Dover, 1968.

[Tan97]   Bariş Tan, *Markov chains and the risk board game*, Mathematics Magazine **70** (1997), no. 5, 349–357.

[TK98]   H. M. Taylor and Samuel Karlin, *An introduction to stochastic modeling*, third ed., Academic Press, 1998.

[Wil]   Darren Wilkinson, *A simple gibbs sampler*, accessed April 7, 2020.

Index