

# Introduction to Probability for Computing

MOR HARCHOL-BALTER

*Cambridge University Press*

*Illustrated by Elin Zhou*



“Based on 20 years of teaching Computer Science and Operations Research at Carnegie Mellon University, Professor Harchol-Balter provides a unique presentation of probability and statistics that is both highly engaging and also strongly motivated by real-world computing applications that students will encounter in industry. This book is approachable and fun for undergraduate students, while also covering advanced concepts relevant to graduate students.”

*Eytan Modiano, Massachusetts Institute of Technology*

“This book provides a fantastic introduction to probability for computer scientists and computing professionals, addressing concepts and techniques crucial to the design and analysis of randomized algorithms, to performance well-designed simulations, to statistical inference and machine learning, and more. Also contains many great exercises and examples. Highly recommend!”

*Avrim Blum, Toyota Technological Institute at Chicago*

“Mor Harchol-Balter’s new book does a beautiful job of introducing students to probability! The book is full of great computer science-relevant examples, wonderful intuition, simple and clear explanations, and mathematical rigor. I love the question-answer style she uses, and could see using this book for students ranging from undergraduate students with zero prior exposure to probability all the way to graduate students (or researchers of any kind) who need to brush up and significantly deepen (and/or broaden) their knowledge of probability.”

*Anna Karlin, University of Washington*

“Probability is at the heart of modeling, design, and analysis of computer systems and networks. This book by a pioneer in the area is a beautiful introduction to the topic for undergraduate students. The material in the book introduces theoretical topics rigorously, but also motivates each topic with practical applications. This textbook is an excellent resource for budding computer scientists who are interested in probability.”

*R. Srikant, University of Illinois at Urbana-Champaign*

“I know probability, and have taught it to undergrads and grads at MIT, UC Berkeley, and Carnegie Mellon University. Yet this book has taught me some wonderfully interesting important material that I did not know. Mor is a great thinker, lecturer, and writer. I would love to have learned from this book as a student – and to have taught from it as an instructor!”

*Manuel Blum, U.C. Berkeley and Carnegie Mellon University*

*To the students at CMU's  
School of Computer Science  
whose curiosity and drive  
inspire me every day  
to keep writing.*



# Contents

**Preface** *page* xvi

**Acknowledgments** xxiii

## **Part I Fundamentals and Probability on Events**

**1 Before We Start ... Some Mathematical Basics** 2

1.1 Review of Simple Series 2

1.2 Review of Double Integrals and Sums 4

1.3 Fundamental Theorem of Calculus 7

1.4 Review of Taylor Series and Other Limits 8

1.5 A Little Combinatorics 11

1.6 Review of Asymptotic Notation 15

1.6.1 Big-O and Little-o 15

1.6.2 Big-Omega and Little-omega 17

1.6.3 Big-Theta 18

1.7 Exercises 18

**2 Probability on Events** 21

2.1 Sample Space and Events 21

2.2 Probability Defined on Events 22

2.3 Conditional Probabilities on Events 24

2.4 Independent Events 27

2.5 Law of Total Probability 30

2.6 Bayes' Law 32

2.7 Exercises 34

## **Part II Discrete Random Variables**

**3 Common Discrete Random Variables** 44

3.1 Random Variables 44

3.2 Common Discrete Random Variables 45

3.2.1 The Bernoulli( $p$ ) Random Variable 46

3.2.2 The Binomial( $n, p$ ) Random Variable 47

3.2.3 The Geometric( $p$ ) Random Variable 48

3.2.4	The Poisson( $\lambda$ ) Random Variable	49
3.3	Multiple Random Variables and Joint Probabilities	50
3.4	Exercises	54
<b>4</b>	<b>Expectation</b>	<b>58</b>
4.1	Expectation of a Discrete Random Variable	58
4.2	Linearity of Expectation	63
4.3	Conditional Expectation	67
4.4	Computing Expectations via Conditioning	72
4.5	Simpson's Paradox	74
4.6	Exercises	76
<b>5</b>	<b>Variance, Higher Moments, and Random Sums</b>	<b>83</b>
5.1	Higher Moments	83
5.2	Variance	85
5.3	Alternative Definitions of Variance	86
5.4	Properties of Variance	88
5.5	Summary Table for Discrete Distributions	91
5.6	Covariance	91
5.7	Central Moments	92
5.8	Sum of a Random Number of Random Variables	93
5.9	Tails	97
	5.9.1 Simple Tail Bounds	98
	5.9.2 Stochastic Dominance	99
5.10	Jensen's Inequality	102
5.11	Inspection Paradox	104
5.12	Exercises	107
<b>6</b>	<b>z-Transforms</b>	<b>116</b>
6.1	Motivating Examples	116
6.2	The Transform as an Onion	117
6.3	Creating the Transform: Onion Building	118
6.4	Getting Moments: Onion Peeling	120
6.5	Linearity of Transforms	121
6.6	Conditioning	123
6.7	Using z-Transforms to Solve Recurrence Relations	124
6.8	Exercises	128
<b>Part III Continuous Random Variables</b>		
<b>7</b>	<b>Continuous Random Variables: Single Distribution</b>	<b>134</b>
7.1	Probability Density Functions	134
7.2	Common Continuous Distributions	137
7.3	Expectation, Variance, and Higher Moments	141

7.4	Computing Probabilities by Conditioning on a R.V.	143
7.5	Conditional Expectation and the Conditional Density	146
7.6	Exercises	150
<b>8</b>	<b>Continuous Random Variables: Joint Distributions</b>	<b>153</b>
8.1	Joint Densities	153
8.2	Probability Involving Multiple Random Variables	156
8.3	Pop Quiz	160
8.4	Conditional Expectation for Multiple Random Variables	161
8.5	Linearity and Other Properties	163
8.6	Exercises	163
<b>9</b>	<b>Normal Distribution</b>	<b>170</b>
9.1	Definition	170
9.2	Linear Transformation Property	172
9.3	The Cumulative Distribution Function	173
9.4	Central Limit Theorem	176
9.5	Exercises	178
<b>10</b>	<b>Heavy Tails: The Distributions of Computing</b>	<b>181</b>
10.1	Tales of Tails	181
10.2	Increasing versus Decreasing Failure Rate	183
10.3	UNIX Process Lifetime Measurements	186
10.4	Properties of the Pareto Distribution	187
10.5	The Bounded-Pareto Distribution	189
10.6	Heavy Tails	189
10.7	The Benefits of Active Process Migration	190
10.8	From the 1990s to the 2020s	191
10.9	Pareto Distributions Are Everywhere	192
10.10	Summary Table for Continuous Distributions	194
10.11	Exercises	194
<b>11</b>	<b>Laplace Transforms</b>	<b>198</b>
11.1	Motivating Example	198
11.2	The Transform as an Onion	198
11.3	Creating the Transform: Onion Building	200
11.4	Getting Moments: Onion Peeling	201
11.5	Linearity of Transforms	203
11.6	Conditioning	203
11.7	Combining Laplace and z-Transforms	204
11.8	One Final Result on Transforms	205
11.9	Exercises	206



---

## Part IV Computer Systems Modeling and Simulation

<b>12</b>	<b>The Poisson Process</b>	210
12.1	Review of the Exponential Distribution	210
12.2	Relating the Exponential Distribution to the Geometric	211
12.3	More Properties of the Exponential	213
12.4	The Celebrated Poisson Process	216
12.5	Number of Poisson Arrivals during a Random Time	219
12.6	Merging Independent Poisson Processes	220
12.7	Poisson Splitting	221
12.8	Uniformity	224
12.9	Exercises	225
<b>13</b>	<b>Generating Random Variables for Simulation</b>	229
13.1	Inverse Transform Method	229
13.1.1	The Continuous Case	230
13.1.2	The Discrete Case	231
13.2	Accept–Reject Method	232
13.2.1	Discrete Case	233
13.2.2	Continuous Case	234
13.2.3	A Harder Problem	238
13.3	Readings	238
13.4	Exercises	238
<b>14</b>	<b>Event-Driven Simulation</b>	240
14.1	Some Queueing Definitions	240
14.2	How to Run a Simulation	242
14.3	How to Get Performance Metrics from Your Simulation	244
14.4	More Complex Examples	247
14.5	Exercises	249

## Part V Statistical Inference

<b>15</b>	<b>Estimators for Mean and Variance</b>	255
15.1	Point Estimation	255
15.2	Sample Mean	256
15.3	Desirable Properties of a Point Estimator	256
15.4	An Estimator for Variance	259
15.4.1	Estimating the Variance when the Mean is Known	259
15.4.2	Estimating the Variance when the Mean is Unknown	259
15.5	Estimators Based on the Sample Mean	261
15.6	Exercises	263
15.7	Acknowledgment	264

<b>16</b>	<b>Classical Statistical Inference</b>	265
16.1	Towards More General Estimators	265
16.2	Maximum Likelihood Estimation	267
16.3	More Examples of ML Estimators	270
16.4	Log Likelihood	271
16.5	MLE with Data Modeled by Continuous Random Variables	273
16.6	When Estimating More than One Parameter	276
16.7	Linear Regression	277
16.8	Exercises	283
16.9	Acknowledgment	284
<b>17</b>	<b>Bayesian Statistical Inference</b>	285
17.1	A Motivating Example	285
17.2	The MAP Estimator	287
17.3	More Examples of MAP Estimators	290
17.4	Minimum Mean Square Error Estimator	294
17.5	Measuring Accuracy in Bayesian Estimators	299
17.6	Exercises	301
17.7	Acknowledgment	304
<b>Part VI Tail Bounds and Applications</b>		
<b>18</b>	<b>Tail Bounds</b>	306
18.1	Markov's Inequality	307
18.2	Chebyshev's Inequality	308
18.3	Chernoff Bound	309
18.4	Chernoff Bound for Poisson Tail	311
18.5	Chernoff Bound for Binomial	312
18.6	Comparing the Different Bounds and Approximations	313
18.7	Proof of Chernoff Bound for Binomial: Theorem 18.4	315
18.8	A (Sometimes) Stronger Chernoff Bound for Binomial	316
18.9	Other Tail Bounds	318
18.10	Appendix: Proof of Lemma 18.5	319
18.11	Exercises	320
<b>19</b>	<b>Applications of Tail Bounds: Confidence Intervals and Balls and Bins</b>	327
19.1	Interval Estimation	327
19.2	Exact Confidence Intervals	328
19.2.1	Using Chernoff Bounds to Get Exact Confidence Intervals	328
19.2.2	Using Chebyshev Bounds to Get Exact Confidence Intervals	331
19.2.3	Using Tail Bounds to Get Exact Confidence Intervals in General Settings	332
19.3	Approximate Confidence Intervals	334

---

19.4	Balls and Bins	337
19.5	Remarks on Balls and Bins	341
19.6	Exercises	341
<b>20</b>	<b>Hashing Algorithms</b>	<b>346</b>
20.1	What is Hashing?	346
20.2	Simple Uniform Hashing Assumption	348
20.3	Bucket Hashing with Separate Chaining	349
20.4	Linear Probing and Open Addressing	352
20.5	Cryptographic Signature Hashing	355
20.6	Remarks	360
20.7	Exercises	360

## Part VII Randomized Algorithms

<b>21</b>	<b>Las Vegas Randomized Algorithms</b>	<b>364</b>
21.1	Randomized versus Deterministic Algorithms	364
21.2	Las Vegas versus Monte Carlo	366
21.3	Review of Deterministic Quicksort	367
21.4	Randomized Quicksort	368
21.5	Randomized Selection and Median-Finding	370
21.6	Exercises	373
<b>22</b>	<b>Monte Carlo Randomized Algorithms</b>	<b>383</b>
22.1	Randomized Matrix-Multiplication Checking	383
22.2	Randomized Polynomial Checking	387
22.3	Randomized Min-Cut	389
22.4	Related Readings	394
22.5	Exercises	394
<b>23</b>	<b>Primality Testing</b>	<b>403</b>
23.1	Naive Algorithms	403
23.2	Fermat's Little Theorem	404
23.3	Fermat Primality Test	408
23.4	Miller–Rabin Primality Test	410
	23.4.1 A New Witness of Compositeness	410
	23.4.2 Logic Behind the Miller–Rabin Test	411
	23.4.3 Miller–Rabin Primality Test	413
23.5	Readings	415
23.6	Appendix: Proof of Theorem 23.9	415
23.7	Exercises	417

## Part VIII Discrete-Time Markov Chains

<b>24</b>	<b>Discrete-Time Markov Chains: Finite-State</b>	420
24.1	Our First Discrete-Time Markov Chain	420
24.2	Formal Definition of a DTMC	421
24.3	Examples of Finite-State DTMCs	422
24.3.1	Repair Facility Problem	422
24.3.2	Umbrella Problem	423
24.3.3	Program Analysis Problem	424
24.4	Powers of $\mathbf{P}$ : $n$ -Step Transition Probabilities	425
24.5	Limiting Probabilities	426
24.6	Stationary Equations	428
24.7	The Stationary Distribution Equals the Limiting Distribution	429
24.8	Examples of Solving Stationary Equations	432
24.9	Exercises	433
<b>25</b>	<b>Ergodicity for Finite-State Discrete-Time Markov Chains</b>	438
25.1	Some Examples on Whether the Limiting Distribution Exists	439
25.2	Aperiodicity	441
25.3	Irreducibility	442
25.4	Aperiodicity plus Irreducibility Implies Limiting Distribution	443
25.5	Mean Time Between Visits to a State	448
25.6	Long-Run Time Averages	450
25.6.1	Strong Law of Large Numbers	452
25.6.2	A Bit of Renewal Theory	454
25.6.3	Equality of the Time Average and Ensemble Average	455
25.7	Summary of Results for Ergodic Finite-State DTMCs	456
25.8	What If My DTMC Is Irreducible but Periodic?	456
25.9	When the DTMC Is Not Irreducible	457
25.10	An Application: PageRank	458
25.10.1	Problems with Real Web Graphs	461
25.10.2	Google's Solution to Dead Ends and Spider Traps	462
25.10.3	Evaluation of the PageRank Algorithm and Practical Considerations	463
25.11	From Stationary Equations to Time-Reversibility Equations	464
25.12	Exercises	469
<b>26</b>	<b>Discrete-Time Markov Chains: Infinite-State</b>	479
26.1	Stationary = Limiting	479
26.2	Solving Stationary Equations in Infinite-State DTMCs	480
26.3	A Harder Example of Solving Stationary Equations in Infinite-State DTMCs	483
26.4	Ergodicity Questions	484
26.5	Recurrent versus Transient: Will the Fish Return to Shore?	487
26.6	Infinite Random Walk Example	490
26.7	Back to the Three Chains and the Ergodicity Question	492

---

26.7.1	Figure 26.8(a) is Recurrent	492
26.7.2	Figure 26.8(b) is Transient	492
26.7.3	Figure 26.8(c) is Recurrent	494
26.8	Why Recurrence Is Not Enough	494
26.9	Ergodicity for Infinite-State Chains	496
26.10	Exercises	498
<b>27</b>	<b>A Little Bit of Queueing Theory</b>	<b>510</b>
27.1	What Is Queueing Theory?	510
27.2	A Single-Server Queue	511
27.3	Kendall Notation	513
27.4	Common Performance Metrics	514
27.4.1	Immediate Observations about the Single-Server Queue	515
27.5	Another Metric: Throughput	516
27.5.1	Throughput for $M/G/k$	517
27.5.2	Throughput for Network of Queues with Probabilistic Routing	518
27.5.3	Throughput for Network of Queues with Deterministic Routing	519
27.5.4	Throughput for Finite Buffer	520
27.6	Utilization	520
27.7	Introduction to Little's Law	521
27.8	Intuitions for Little's Law	522
27.9	Statement of Little's Law	524
27.10	Proof of Little's Law	525
27.11	Important Corollaries of Little's Law	527
27.12	Exercises	531
	<b>References</b>	<b>539</b>
	<b>Index</b>	<b>544</b>

# Preface

Probability theory has become indispensable in computer science. It is at the core of machine learning and statistics, where one often needs to make decisions under stochastic uncertainty. It is also integral to computer science theory, where most algorithms today are randomized algorithms, involve random coin flips. It is a central part of performance modeling in computer networks and systems, where probability is used to predict delays, schedule jobs and resources, and provision capacity.

## Why This Book?

This book gives an introduction to probability as it is used in computer science theory and practice, drawing on applications and current research developments as motivation and context. This is not a typical counting and combinatorics book, but rather it is a book centered on distributions and how to work with them.

Every topic is driven by what computer science students need to know. For example, the book covers distributions that come up in computer science, such as heavy-tailed distributions. There is a large emphasis on variability and higher moments, which are very important in empirical computing distributions. Computer systems modeling and simulation are also discussed, as well as statistical inference for estimating parameters of distributions. Much attention is devoted to tail bounds, such as Chernoff bounds. Chernoff bounds are used for confidence intervals and also play a big role in the analysis of randomized algorithms, which themselves comprise a large part of the book. Finally, the book covers Markov chains, as well as a bit of queueing theory, both with an emphasis on their use in computer systems analysis.

## Intended Audience

The material is presented at the advanced undergraduate level. The book is based on an undergraduate class, Probability and Computing (PnC), which I have been teaching at Carnegie Mellon University (CMU) for almost 20 years. While PnC is primarily taken by undergraduates, several Masters and PhD students choose to take the class. Thus we imagine that instructors can use the book for different levels of classes, perhaps spanning multiple semesters.

## Question/Answer Writing Style

The book uses a style of writing aimed at engaging the reader to be active, rather than passive. Instead of large blocks of text, we have short “Questions” and “Answers.” In working through the book, you should cover up the answers, and write down your own answer to each question, before looking at the given answer. The goal is “thinking” rather than “reading,” where each chapter is intended to feel like a conversation.

## Exercises

The exercises in this book are an integral part of learning the material. They also introduce many of the computer science and statistics applications. Very few of the exercises are rote. Every problem has important insights, and the insights often build on each other. Exercises are (very roughly) organized from easier to harder. Several of the exercises in the book were contributed by students in the class!

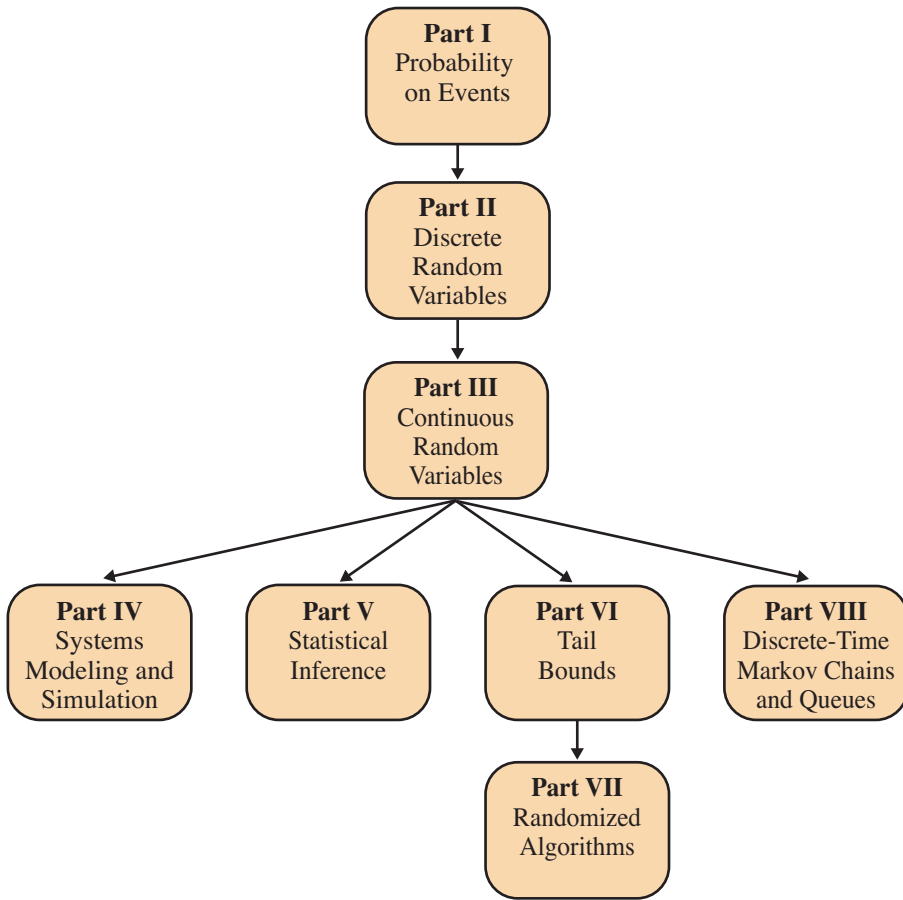
To aid in teaching, solutions to a large subset of the exercises are available *for instructors only* at [www.cambridge.org/harchol-balter](http://www.cambridge.org/harchol-balter). Instructors who need solutions to the remaining exercises can request these from the author. The solutions are for the personal use of the instructor only. They should not be distributed or posted online, so that future generations can continue to enjoy the exercises.

## Organization of the Material

The book consists of eight parts. Parts I, II, and III provide an introduction to basic probability. Part IV provides an introduction to computer systems modeling and simulation. Part V provides an introduction to statistical inference. Parts VI and VII comprise a course in randomized algorithms, starting with tail bound inequalities and then applying these to analyze a long list of randomized algorithms. Part VIII provides an introduction to stochastic processes as they’re used in computing.

Before we describe the parts in more detail, it is worth looking at the *dependency structure* for the book, given in Figure P1. Aside from Parts I, II, and III, most of the parts can be taught in any order.

In particular, it is possible to imagine at least *four different courses* being taught from this book, depending on the parts that an instructor might choose to teach. Figure P2 depicts different courses that one might teach. All the courses start with Parts I, II, and III, but then continue with Simulation, or Statistics, or Randomized Algorithms, or Stochastic Processes, depending on the particular course.



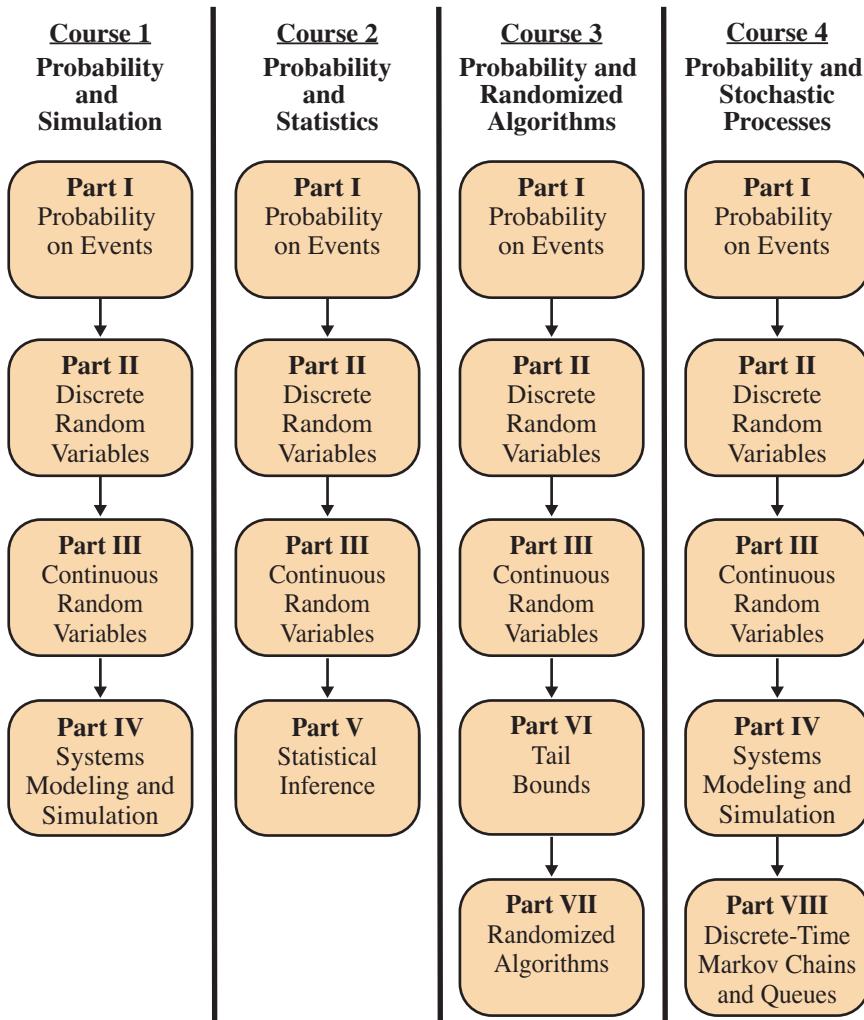
**Figure P1** The dependency structure between the parts of this book. Most parts are independent of other parts and can be taught in any order.

## Description of Each Part

**Part I: Foundations and Probability on Events:** Part I starts by reviewing the prerequisites for the book. These include series, calculus, elementary combinatorics, and asymptotic notation. Exercises and examples are provided to help in reviewing the prerequisites. The main focus of Part I is on defining probability on events, including conditioning on events, independence of events, the Law of Total Probability, and Bayes' Law. Some examples of applications covered in Part I are: faulty computer networks, Bayesian reasoning for healthcare testing, modeling vaccine efficacy, the birthday paradox, Monty Hall problems, and modeling packet corruption in the Internet.

**Part II: Discrete Random Variables:** Part II introduces the most common discrete random variables (Bernoulli, Binomial, Geometric, and Poisson), and then





**Figure P2** Four different courses that one can teach out of this book.

continues with the standard material on random variables, such as linearity of expectation, conditioning, conditional probability mass functions, joint distributions, and marginal distributions. Some more advanced material is also included, such as: variance and higher moments of random variables; moment-generating functions (specifically z-transforms) and their use in solving recurrence relations; Jensen’s inequality; sums of a random number of random variables; tail orderings, and simple tail inequalities. Both Simpson’s paradox and the inspection paradox are covered. Some examples of applications covered in Part II are: noisy reading from a flash storage, the binary symmetric channel, approximating a Binomial distribution by a Poisson, the classical marriage algorithm, modeling the time until a disk fails, the coupon collector problem, properties of

random graphs, time until  $k$  consecutive failures, computer virus propagation, epidemic growth modeling, hypothesis testing in data analysis, stopping times, total variation distance, and polygon triangulation.

**Part III: Continuous Random Variables:** Part III repeats the material in Part II, but this time with continuous random variables. We introduce the Uniform, Exponential, and Normal distributions, as well as the Central Limit Theorem. In addition, we introduce the Pareto heavy-tailed distribution, which is most relevant for empirical computing workloads, and discuss its relevance to today's data center workloads. We cover failure rate functions and the heavy-tail property and their relevance to computing workloads. We again cover moment-generating functions, but this time via Laplace transforms, which are more commonly used with continuous random variables. Some applications covered in Part II are: classifying jobs in a supercomputing center, learning the bias of a coin, dart throwing, distributions whose parameters are random variables, relating laptop quality to lifetime, modeling disk delays, modeling web file sizes, modeling compute usage, modeling IP flow durations, and Internet node degrees.

**Part IV: Computer Systems Modeling and Simulation:** Part IV covers the basics of what is needed to run simulations of computer systems. We start by defining and analyzing the Poisson process, which is the most commonly used model for the arrival process of jobs into computer systems. We then study how to generate random variables for simulation, using the inverse transform method and the accept-reject method. Finally, we discuss how one would program a simple event-driven or trace-driven simulator. Some applications that we cover include: Malware detection of infected hosts, population modeling, reliability theory, generating a Normal random variable, generating Pareto and Bounded Pareto random variables, generating a Poisson random variable, simulation of heavy-tailed distributions, simulation of high-variance distributions, simulation of jointly distributed random variables, simulation of queues, and simulation of networks of queues.

**Part V: Statistical Inference:** Part V switches gears to statistics, particularly statistical inference, where one is trying to estimate some parameters of an experiment. We start with the most traditional estimators, the sample mean and sample variance. We also cover desirable properties of estimators, including zero bias, low mean squared error, and consistency. We next cover maximum likelihood estimation and linear regression. We complete this part with a discussion of maximum a posterior (MAP) estimators and minimum mean square error (MMSE) estimators. Some applications that we cover include: estimating voting probabilities, deducing the original signal in a noisy environment, estimating true job sizes from user estimates, estimation in interaction graphs, and estimation in networks with error correcting codes.

**Part VI: Tail Bounds and Applications:** Part VI starts with a discussion of tail bounds and concentration inequalities (Markov, Chebyshev, Chernoff), for which we provide full derivations. We provide several immediate applications for these tail bounds, including a variety of classic balls-and-bins applications. The balls and bins framework has immediate application to dispatching tasks to servers in a server farm, as well as immediate application to hashing algorithms, which we also study extensively. We cover applications of tail bounds to defining confidence intervals in statistical estimation, and well as bias estimation, polling schemes, crowd sourcing, and other common settings from computing and statistics.

**Part VII: Randomized Algorithms:** Part VII introduces a wide range of randomized algorithms. The randomized algorithms include Las Vegas algorithms, such as randomized algorithms for sorting and median finding, as well as Monte Carlo randomized algorithms such as MinCut, MaxCut, matrix multiplication checking, polynomial multiplication, and primality testing. The exercises in this part are particularly relevant because they introduce many additional randomized algorithms such as randomized dominating set, approximate median finding, independent set, AND/OR tree evaluation, knockout tournaments, addition of  $n$ -bit numbers, randomized string exchange, path-finding in graphs, and more. We use the tail bounds that we derived earlier in Part VI to analyze the runtimes and accuracy of our randomized algorithms.

**Part VIII: Markov Chains with a Side of Queueing Theory:** Part VIII provides an introduction to stochastic processes as they come up in computer science. Here we delve deeply into discrete-time Markov chains (both finite and infinite). We discuss not only how to solve for limiting distributions, but also when they exist and why. Ergodicity, positive-recurrence and null-recurrence, passage times, and renewal theory are all covered. We also cover time averages versus ensemble averages and the impact of these different types of averages on running simulations. Queueing theory is integral to Part VIII. We define the performance metrics that computer scientists care about: throughput, response time, and load. We cover Little's Law, stability, busy periods, and capacity provisioning. A huge number of applications are covered in Part VIII, including, for example, the classic PageRank algorithm for ranking web pages, modeling of epidemic spread, modeling of caches, modeling processors with failures, Brownian motion, estimating the spread of malware, reliability theory applications, population modeling, server farm and data center modeling, admission control, and capacity provisioning.



# Acknowledgments

Most textbooks begin with a class, and this book is no exception. I created the Probability and Computing (called “PnC” for short) class 20 years ago, with the aim of teaching computer science undergraduates the probability that they need to know to be great computer scientists. Since then I have had a few opportunities to co-teach PnC with different colleagues, and each such opportunity has led to my own learning. I would like to thank my fantastic co-instructors: John Lafferty, Klaus Sutner, Rashmi Vinayak, Ryan O’Donnell, Victor Adamchik, and Weina Wang. I’m particularly grateful to Weina, who collaborated with me on three of the chapters of the book and who is a kindred spirit in Socratic teaching. The book has also benefited greatly from many spirited TAs and students in the class, who proposed fun exercises for the book, many referencing CMU or Pittsburgh.

I would also like to thank my illustrator, Elin Zhou, who painstakingly created every image and figure in the book, while simultaneously managing her undergraduate classes at CMU. I chose Elin as my illustrator because her artwork embodies the spirit of fun and inclusiveness that permeates the PnC class. One of the themes of PnC is chocolate, which is tossed out throughout the class to students who answer questions. This chocolate would not be possible if it weren’t for our class sponsor, Citadel, who even paid to have chocolate mailed directly to student homes throughout the pandemic, while classes were online.

I have been fortunate to have several excellent editors at Cambridge University Press: Julie Lancashire, Ilaria Tassistro, and Rachel Norridge. Thanks to their recommendations, the statistics chapters were added, redundant material was removed, and the style and layout of the book improved immensely. My copy editor, Gary Smith, was also fantastic to work with and meticulous!

On a personal note, I want to thank my family. In particular, I’m grateful to my son, Danny Balter, for always telling me that I’m good at explaining things. I’m also grateful to my mom, Irit Harchol, who is one of my best friends, and who takes the time to talk with me every day as I walk to and from work. Thanks to my inlaws, Ann and Richard Young, who are my cheering squad. Finally, I have infinite love and gratitude for my husband, Ary Young, for always making me their top priority and for never leaving my side, even if it means sleeping on my sofa as I sit here typing away.

