

# CONTENTS

---

Preface	xv
Acknowledgments	xix
<b>1 Review</b>	<b>1</b>
1.1 Mathematical notation	1
1.2 Taylor's theorem and mathematical limit theory	2
1.3 Statistical notation and probability distributions	4
1.4 Likelihood inference	6
1.5 Bayesian inference	11
1.6 Statistical limit theory	13
1.7 Markov chains	14
1.8 Computing	17
<b>PART I OPTIMIZATION</b>	
<b>2 Optimization and Solving Nonlinear Equations</b>	<b>3</b>
2.1 Univariate problems	5
2.1.1 Newton's method	9
2.1.1.1 Convergence order	12
2.1.2 Fisher scoring	13
2.1.3 Secant method	13
	<b>vii</b>

2.1.4	Fixed-point iteration	15
2.1.4.1	Scaling	15
2.2	Multivariate problems	17
2.2.1	Newton's method and Fisher scoring	17
2.2.1.1	Iteratively reweighted least squares	18
2.2.2	Newton-like methods	21
2.2.2.1	Ascent algorithms	21
2.2.2.2	Discrete Newton and fixed-point methods	23
2.2.2.3	Quasi-Newton methods	24
2.2.3	Gauss–Newton method	26
2.2.4	Nelder–Mead algorithm	27
2.2.5	Nonlinear Gauss–Seidel iteration	36
	Problems	36
<b>3</b>	<b>Combinatorial Optimization</b>	<b>43</b>
3.1	Hard problems and NP-completeness	44
3.1.1	Examples	46
3.1.2	The need for heuristics	49
3.2	Local search	50
3.3	Simulated annealing	53
3.3.1	Practical issues	55
3.3.1.1	Neighborhoods and proposals	55
3.3.1.2	Cooling schedule and convergence	56
3.3.2	Enhancements	59
3.4	Genetic algorithms	60
3.4.1	Definitions and the canonical algorithm	61
3.4.1.1	Basic definitions	61
3.4.1.2	Selection mechanisms and genetic operators	62
3.4.1.3	Allele alphabets and genotypic representation	63
3.4.1.4	Initialization, termination, and parameter values	65
3.4.2	Variations	65
3.4.2.1	Fitness	65
3.4.2.2	Selection mechanisms and updating generations	66
3.4.2.3	Genetic operators and permutation chromosomes	67
3.4.3	Initialization and parameter values	70
3.4.4	Convergence	70
3.5	Tabu algorithms	71
3.5.1	Basic definitions	72
3.5.2	The tabu list	72
3.5.3	Aspiration criteria	74
3.5.4	Diversification	75
3.5.5	Intensification	76
3.5.6	A comprehensive tabu algorithm	77
	Problems	78

<b>4</b>	<b>EM Optimization Methods</b>	<b>83</b>
4.1	Missing data, marginalization, and notation	84
4.2	The EM algorithm	84
4.2.1	Convergence	89
4.2.2	Usage in exponential families	92
4.2.3	Variance estimation	93
4.2.3.1	Louis's method	93
4.2.3.2	SEM algorithm	95
4.2.3.3	Bootstrapping	97
4.2.3.4	Empirical information	97
4.2.3.5	Numerical differentiation	98
4.3	EM Variants	98
4.3.1	Improving the E step	98
4.3.1.1	Monte Carlo EM	98
4.3.2	Improving the M step	99
4.3.2.1	ECM algorithm	99
4.3.2.2	EM gradient algorithm	103
4.3.3	Acceleration methods	104
4.3.3.1	Aitken acceleration	104
4.3.3.2	Quasi-Newton acceleration	106
	Problems	108
<b>PART II INTEGRATION AND SIMULATION</b>		
<b>5</b>	<b>Numerical Integration</b>	<b>117</b>
5.1	Newton–Côtes quadrature	118
5.1.1	Riemann rule	119
5.1.2	Trapezoidal rule	123
5.1.3	Simpson's rule	125
5.1.4	General $k$ th-degree rule	127
5.2	Romberg integration	127
5.3	Gaussian quadrature	131
5.3.1	Orthogonal polynomials	132
5.3.2	The Gaussian quadrature rule	132
5.4	Frequently encountered problems	135
5.4.1	Range of integration	135
5.4.2	Integrands with singularities or other extreme behavior	135
5.4.3	Multiple integrals	136
5.4.4	Adaptive quadrature	136
5.4.5	Software for exact integration	136
	Problems	137
<b>6</b>	<b>Simulation and Monte Carlo Integration</b>	<b>139</b>
6.1	Introduction to the Monte Carlo method	140

6.2	Exact Simulation	141
6.2.1	Generating from standard parametric families	141
6.2.2	Inverse cumulative distribution function	143
6.2.3	Rejection sampling	143
6.2.3.1	Squeezed rejection sampling	146
6.2.3.2	Adaptive rejection sampling	148
6.3	Approximate Simulation	152
6.3.1	The sampling importance resampling algorithm	152
6.3.1.1	Adaptive importance, bridge, and path sampling	156
6.3.2	Sequential Monte Carlo	157
6.3.2.1	Sequential importance sampling for Markov processes	158
6.3.2.2	General sequential importance sampling	160
6.3.2.3	Weight degeneracy, rejuvenation and effective sample size	161
6.3.2.4	Sequential importance sampling for hidden Markov models	165
6.3.2.5	Particle filters	169
6.4	Variance reduction techniques	170
6.4.1	Importance sampling	170
6.4.2	Antithetic sampling	176
6.4.3	Control variates	179
6.4.4	Rao–Blackwellization	183
	Problems	185
<b>7</b>	<b>Markov Chain Monte Carlo</b>	<b>191</b>
7.1	Metropolis–Hastings algorithm	192
7.1.1	Independence chains	194
7.1.2	Random walk chains	198
7.2	Gibbs sampling	199
7.2.1	Basic Gibbs sampler	200
7.2.2	Properties of the Gibbs sampler	206
7.2.3	Update ordering	207
7.2.4	Blocking	207
7.2.5	Hybrid Gibbs sampling	208
7.2.6	Griddy–Gibbs sampler	209
7.3	Implementation	210
7.3.1	Ensuring good mixing and convergence	210
7.3.1.1	Simple graphical diagnostics	211
7.3.1.2	Burn-in and run length	212
7.3.1.3	Choice of proposal	214
7.3.1.4	Reparameterization	214
7.3.1.5	Comparing chains: Effective sample size	215
7.3.1.6	Number of chains	216

7.3.2	Practical implementation advice	217
7.3.3	Using the results	218
	Problems	222
<b>8</b>	<b>Advanced Topics in MCMC</b>	<b>229</b>
8.1	Adaptive MCMC	229
8.1.1	Adaptive Random Walk Metropolis-within-Gibbs algorithm	230
8.1.2	General Adaptive Metropolis-Within-Gibbs	233
8.1.3	Adaptive Metropolis algorithm	240
8.2	Reversible Jump MCMC	243
8.2.1	RJMCMC for variable selection in regression	247
8.3	Auxiliary variable methods	250
8.3.1	Simulated tempering	251
8.3.2	Slice sampler	252
8.4	Other Metropolis-Hastings Algorithms	254
8.4.1	Hit-and-run algorithm	254
8.4.2	Multiple-try Metropolis-Hastings algorithm	255
8.4.3	Langevin Metropolis-Hastings algorithm	257
8.5	Perfect sampling	258
8.5.1	Coupling from the past	258
8.5.1.1	Stochastic monotonicity and sandwiching	261
8.6	Markov chain maximum likelihood	262
8.7	Example: MCMC for Markov random fields	263
8.7.1	Gibbs sampling for Markov random fields	264
8.7.2	Auxiliary variable methods for Markov random fields	269
8.7.3	Perfect sampling for Markov random fields	272
	Problems	274
	<b>PART III APPROXIMATING DISTRIBUTIONS</b>	
<b>9</b>	<b>Bootstrapping</b>	<b>281</b>
9.1	The bootstrap principle	281
9.2	Basic methods	283
9.2.1	Nonparametric bootstrap	283
9.2.2	Parametric bootstrap	284
9.2.3	Bootstrapping regression	284
9.2.4	Bootstrap bias correction	286
9.3	Bootstrap inference	286
9.3.1	Percentile method	286
9.3.1.1	Justification for the percentile method	288
9.3.2	Pivoting	289
9.3.2.1	Accelerated bias-corrected percentile method, $BC_a$	289
9.3.2.2	The bootstrap $t$	291

9.3.2.3	Empirical variance stabilization	292
9.3.2.4	Nested bootstrap and pre pivoting	293
9.3.3	Hypothesis testing	296
9.4	Reducing Monte Carlo error	297
9.4.1	Balanced bootstrap	297
9.4.2	Antithetic bootstrap	297
9.5	Bootstrapping dependent data	298
9.5.1	Model-based approach	299
9.5.2	Block bootstrap	299
9.5.2.1	Non-moving block bootstrap	299
9.5.2.2	Moving block bootstrap	300
9.5.2.3	Blocks of blocks bootstrapping	301
9.5.2.4	Centering and studentizing	305
9.5.2.5	Block size	306
9.6	Bootstrap performance	310
9.6.1	Independent data case	310
9.6.2	Dependent data case	311
9.7	Other uses of the bootstrap	312
9.8	Permutation tests	313
	Problems	314

## PART IV DENSITY ESTIMATION AND SMOOTHING

<b>10</b>	<b>Nonparametric Density Estimation</b>	<b>321</b>
10.1	Measures of performance	322
10.2	Kernel density estimation	324
10.2.1	Choice of bandwidth	325
10.2.1.1	Cross-validation	329
10.2.1.2	Plug-in methods	331
10.2.1.3	Maximal smoothing principle	334
10.2.2	Choice of kernel	336
10.2.2.1	Epanechnikov kernel	336
10.2.2.2	Canonical kernels and rescalings	337
10.3	Nonkernel methods	338
10.3.1	Logspline	338
10.4	Multivariate methods	341
10.4.1	The nature of the problem	341
10.4.2	Multivariate kernel estimators	343
10.4.3	Adaptive kernels and nearest neighbors	346
10.4.3.1	Nearest neighbor approaches	346
10.4.3.2	Variable-kernel approaches and transformations	348
10.4.4	Exploratory projection pursuit	351
	Problems	356
<b>11</b>	<b>Bivariate Smoothing</b>	<b>361</b>

11.1	Predictor–response data	362
11.2	Linear smoothers	364
11.2.1	Constant-span running mean	364
11.2.1.1	Effect of span	366
11.2.1.2	Span selection for linear smoothers	367
11.2.2	Running lines and running polynomials	371
11.2.3	Kernel smoothers	373
11.2.4	Local regression smoothing	374
11.2.5	Spline smoothing	375
11.2.5.1	Choice of penalty	376
11.3	Comparison of linear smoothers	376
11.4	Nonlinear smoothers	378
11.4.1	Loess	378
11.4.2	Supersmoother	381
11.5	Confidence bands	385
11.6	General bivariate data	388
	Problems	389
<b>12</b>	<b>Multivariate Smoothing</b>	<b>393</b>
12.1	Predictor–response data	393
12.1.1	Additive models	394
12.1.2	Generalized additive models	397
12.1.3	Other methods related to additive models	401
12.1.3.1	Projection pursuit regression	401
12.1.3.2	Neural networks	404
12.1.3.3	Alternating conditional expectations	404
12.1.3.4	Additivity and variance stabilization	405
12.1.4	Tree-based methods	406
12.1.4.1	Recursive partitioning regression trees	408
12.1.4.2	Tree pruning	410
12.1.4.3	Classification trees	413
12.1.4.4	Other issues for tree-based methods	414
12.2	General multivariate data	415
12.2.1	Principal curves	415
12.2.1.1	Definition and motivation	415
12.2.1.2	Estimation	417
12.2.1.3	Span selection	418
	Problems	419
	<b>Data Acknowledgments</b>	<b>423</b>
	References	425
	Index	453





## PREFACE

---

This book covers most topics needed to develop a broad and thorough working knowledge of modern computational statistics. We seek to develop a practical understanding of how and why existing methods work, enabling readers to use modern statistical methods effectively. Since many new methods are built from components of existing techniques, our ultimate goal is to provide scientists with the tools they need to contribute new ideas to the field.

A growing challenge in science is that there is so much of it. While the pursuit of important new methods and the honing of existing approaches is a worthy goal, there is also a need to organize and distill the teeming jungle of ideas. We attempt to do that here. Our choice of topics reflects our view of what constitutes the core of the evolving field of computational statistics, and what will be interesting and useful for our readers.

Our use of the adjective *modern* in the first sentence of this preface is potentially troublesome: there is no way that this book can cover all the latest, greatest techniques. We have not even tried. We have instead aimed to provide a reasonably up-to-date survey of a broad portion of the field, while leaving room for diversions and esoterica.

The foundations of optimization and numerical integration are covered in this book. We include these venerable topics because (i) they are cornerstones of frequentist and Bayesian inference; (ii) routine application of available software often fails for hard problems; and (iii) the methods themselves are often secondary com-

ponents of other statistical computing algorithms. Some topics we have omitted represent important areas of past and present research in the field, but their priority here is lowered by the availability of high-quality software. For example, the generation of pseudo-random numbers is a classic topic, but one that we prefer to address by giving students reliable software. Finally, some topics (e.g., principal curves and tabu search) are included simply because they are interesting and provide very different perspectives on familiar problems. Perhaps a future researcher may draw ideas from such topics to design a creative and effective new algorithm.

In this second edition, we have both updated and broadened our coverage, and we now provide computer code. For example, we have added new MCMC topics to reflect continued activity in that popular area. A notable increase in breadth is our inclusion of more methods relevant for problems where statistical dependency is important, such as block bootstrapping and sequential importance sampling. This second edition provides extensive new support in **R**. Specifically, code for the examples in this book is available from the book website [www.stat.colostate.edu/computationalstatistics](http://www.stat.colostate.edu/computationalstatistics).

Our target audience includes graduate students in statistics and related fields, statisticians, and quantitative empirical scientists in other fields. We hope such readers may use the book when applying standard methods and developing new methods.

The level of mathematics expected of the reader does not extend much beyond Taylor series and linear algebra. Breadth of mathematical training is more helpful than depth. Essential review is provided in Chapter 1. More advanced readers will find greater mathematical detail in the wide variety of high-quality books available on specific topics, many of which are referenced in the text. Other readers caring less about analytical details may prefer to focus on our descriptions of algorithms and examples.

The expected level of statistics is equivalent to that obtained by a graduate student in his or her first year of study of the theory of statistics and probability. An understanding of maximum likelihood methods, Bayesian methods, elementary asymptotic theory, Markov chains, and linear models is most important. Many of these topics are reviewed in Chapter 1.

With respect to computer programming, we find that good students can learn as they go. However, a working knowledge of a suitable language allows implementation of the ideas covered in this book to progress much more quickly. We have chosen to forgo any language-specific examples, algorithms, or coding in the text. For those wishing to learn a language while they study this book, we recommend that you choose a high-level, interactive package that permits the flexible design of graphical displays and includes supporting statistics and probability functions, such as **R** and MATLAB.<sup>1</sup> These are the sort of languages often used by researchers during the development of new statistical computing techniques, and they are suitable for implementing all the methods we describe, except in some cases for problems of vast scope or complexity. We use **R** and recommend it. Although lower-level languages

<sup>1</sup>**R** is available for free from [www.r-project.org](http://www.r-project.org). Information about MATLAB can be found at [www.mathworks.com](http://www.mathworks.com).

such as C++ could also be used, they are more appropriate for professional grade implementation of algorithms after researchers have refined the methodology.

The book is organized into four major parts: optimization (Chapters 2, 3, and 4), integration and simulation (Chapters 5, 6, 7, and 8), bootstrapping (Chapter 9) and density estimation and smoothing (Chapters 10, 11 and 12). The chapters are written to stand independently, so a course can be built by selecting the topics one wishes to teach. For a one-semester course, our selection typically weights most heavily topics from Chapters 2, 3, 6, 7, 9, 10, and 11. With a leisurely pace or more thorough coverage, a shorter list of topics could still easily fill a semester course. There is sufficient material here to provide a thorough one-year course of study, notwithstanding any supplemental topics one might wish to teach.

A variety of homework problems are included at the end of each chapter. Some are straightforward, while others require the student to develop a thorough understanding of the model/method being used, to carefully (and perhaps cleverly) code a suitable technique, and to devote considerable attention to the interpretation of results. A few exercises invite open-ended exploration of methods and ideas. We are sometimes asked for solutions to the exercises, but we prefer to sequester them to preserve the challenge for future students and readers.

The datasets discussed in the examples and exercises are available from the book website, [www.stat.colostate.edu/computationalstatistics](http://www.stat.colostate.edu/computationalstatistics). The **R** code is also provided there. Finally, the website includes an errata. Responsibility for all errors lies with us.