Preface

This book began as a revision of *Elements of Computational Statistics*, published by Springer in 2002. That book covered computationally-intensive statistical methods from the perspective of statistical applications, rather than from the standpoint of statistical computing.

Most of the students in my courses in computational statistics were in a program that required multiple graduate courses in numerical analysis, and so in my course in computational statistics, I rarely covered topics in numerical linear algebra or numerical optimization, for example. Over the years, however, I included more discussion of numerical analysis in my computational statistics courses. Also over the years I have taught numerical methods courses with no or very little statistical content. I have also accumulated a number of corrections and small additions to the elements of computational statistics. The present book includes most of the topics from *Elements* and also incorporates this additional material. The emphasis is still on computationally-intensive *statistical* methods, but there is a substantial portion on the numerical methods supporting the statistical applications.

I have attempted to provide a broad coverage of the field of computational statistics. This obviously comes at the price of depth.

Part I, consisting of one rather long chapter, presents some of the most important concepts and facts over a wide range of topics in intermediate-level mathematics, probability and statistics, so that when I refer to these concepts in later parts of the book, the reader has a frame of reference.

Part I attempts to convey the attitude that *computational inference*, together with exact inference and asymptotic inference, is an important component of statistical methods.

Many statements in Part I are made without any supporting argument, but references and notes are given at the end of the chapter. Most readers and students in courses in statistical computing or computational statistics will be familiar with a substantial proportion of the material in Part I, but I do not recommend skipping the chapter. If readers are already familiar with the material, they should just read faster. The perspective in this chapter is that of the "big picture". As is often apparent in oral exams, many otherwise good students lack a basic understanding of what it is all about.

A danger is that the student or the instructor using the book as a text will too quickly gloss over Chapter 1 and miss some subtle points.

Part II addresses statistical computing, a topic dear to my heart. There are many details of the computations that can be ignored by most statisticians, but no matter at what level of detail a statistician needs to be familiar with the computational topics of Part II, there are two simple, higher-level facts that all statisticians should be aware of and which I state often in this book:

> *Computer numbers are not the same as real numbers, and the arithmetic operations on computer numbers are not exactly the same as those of ordinary arithmetic.*

and

> *The form of a mathematical expression and the way the expression should be evaluated in actual practice may be quite different.*

Regarding the first statement, some of the differences in real numbers and computer numbers are summarized in Table 2.1 on page 98.

A prime example of the second statement is the use of the normal equations in linear regression, $X^{\mathrm{T}}Xb = X^{\mathrm{T}}y$. It is quite appropriate to write and discuss these equations. We might consider the elements of $X^{\mathrm{T}}X$, and we might even write the least squares estimate of β as $b = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y$. That does not mean that we ever actually compute $X^{\mathrm{T}}X$ or $(X^{\mathrm{T}}X)^{-1}$, although we may compute functions of those matrices or even certain elements of them.

The most important areas of statistical computing (to me) are

- computer number systems
- algorithms and programming
- function approximation and numerical quadrature
- numerical linear algebra
- solution of nonlinear equations and optimization
- generation of random numbers.

These topics are the subjects of the individual chapters of Part II.

Part III in six relatively short chapters addresses methods and techniques of computational statistics. I think that any exploration of data should begin with graphics, and the first chapter in Part III, Chapter 8, presents a brief overview of some graphical methods, especially those concerned with multidimensional data. The more complicated the structure of the data and the higher the dimension, the more ingenuity is required for visualization of the data; it is, however, in just those situations that graphics is most important. The orientation of the discussion on graphics is that of computational statistics; the emphasis is on discovery, and the important issues that should be considered in making presentation graphics are not addressed.

Chapter 9 discusses methods of projecting higher-dimensional data into lower dimensions. The tools discussed in Chapter 9 will also be used in Part IV for clustering and classification, and, in general, for exploring structure in data. Chapter 10 covers some of the general issues in function *estimation*, building on the material in Chapter 4 on function *approximation*.

Chapter 11 is about Monte Carlo simulation and some of its uses in computational inference, including Monte Carlo tests, in which artificial data are generated according to a hypothesis. Chapters 12 and 13 discuss computational inference using resampling and partitioning of a given dataset. In these methods, a given dataset is used, but Monte Carlo sampling is employed repeatedly on the data. These methods include randomization tests, jackknife techniques, and bootstrap methods, in which data are generated from the empirical distribution of a given sample, that is, the sample is resampled.

Identification of interesting features, or "structure", in data is an important activity in computational statistics. In Part IV, I consider the problem of identification of structure and the general problem of estimation of probability densities. In simple cases, or as approximations in more realistic situations, structure may be described in terms of functional relationships among the variables in a dataset.

The most useful and complete description of a random data-generating process is the associated probability density, if it exists. Estimation of this special type of function is the topic of Chapters 14 and 15, building on general methods discussed in earlier chapters, especially Chapter 10. If the data follow a parametric distribution, or rather, if we are willing to assume that the data follow a parametric distribution, identification of the probability density is accomplished by estimation of the parameters. Nonparametric density estimation is considered in Chapter 15.

Features of interest in data include clusters of observations and relationships among variables that allow a reduction in the dimension of the data. I discuss methods for statistical learning in Chapter 16, building on some of the basic measures introduced in Chapter 9.

Higher-dimensional data have some surprising and counterintuitive properties, and I discuss some of the interesting characteristics of higher dimensions.

In Chapter 17, I discuss asymmetric relationships among variables. For such problems, the objective often is to estimate or predict a response for a given set of explanatory or predictive variables, or to identify the class to which an observation belongs. The approach is to use a given dataset to develop a model or a set of rules that can be applied to new data. Statistical modeling may be computationally intensive because of the number of possible forms considered or because of the recursive partitioning of the data used in selecting a model. In computational statistics, the emphasis is on *building* a model rather than just estimating the parameters in the model. Parametric estimation, of course, plays an important role in building models.

Many of the topics addressed in this book could easily be (and are) subjects for full-length books. My intent is to describe these methods in a general manner and to emphasize commonalities among them. Decompositions of matrices and of functions are examples of basic tools that are used in a variety of settings in computational statistics. Decompositions of matrices, which are introduced on page 28 of Chapter 1, play a major role in many computations in linear algebra and in statistical analysis of linear models. The decompositional approach to matrix computations has been chosen as one of the Top 10 developments in algorithms in the twentieth century. (See page 138.) The PDF decomposition of a function so that the function has a probability density as a factor, introduced on page 37 of Chapter 1, plays an important role in many statistical methods. We encounter this technique in Monte Carlo methods (pages 192 and 418), in function estimation (Chapters 10 and 15), and in projection pursuit (Chapter 16).

My goal has been to introduce a number of topics and devote some suitable proportion of pages to each. I have given a number of references for more in-depth study of most of the topics. The references are not necessarily chosen because they are the "best"; they're just the ones I'm most familiar with. A general reference for a slightly more detailed coverage of most of the topics in this book is the handbook edited by Wolfgang Härdle, Yuichi Mori, and me (Gentle, Härdle, and Mori, 2004).

The material in Chapters 2, 5, and 9 relies heavily on my book on *Matrix Algebra* (Gentle, 2007), and some of the material in Chapters 7 and 11 is based on parts of my book on *Random Number Generation* (Gentle, 2003).

Each chapter has a section called "notes and further reading". The content of these is somewhat eclectic. In some cases, I had fun writing the section, so I went on at some length; in other cases, my interest level was not adequate for generation of any substantial content.

A Little History

While I have generally tried to keep up with developments in computing, and I do not suffer gladly old folks who like to say "well, the way we used to do it was ...", occasionally while writing this book, I looked in *Statistical Computing* to see what Bill Kennedy and I said thirty years ago about the things I discuss in Part II. The biggest change in computing of course has resulted from the personal computer. "Computing" now means a lot more than it did thirty years ago, and there are a lot more people doing it. Advances in display devices has been a natural concurrence with the development of the PC, and this has changed statistical graphics in a quantum way.

While I think that the PC *sui generis* is the Big Thing, the overall advance in computational power is also important. There have been many evolutionary advances, basically on track with Moore's law (so long as we adjust the number of months appropriately). The net result of the evolutionary advance in speed has been enormous. Some people have suggested that statistical methods/approaches should undergo fundamental changes every time there is an increase of one order of magnitude in computational speed and/or storage. Since 1980, and roughly in accordance with Moore's law, there have been 4 such increases. I leave to others an explicit interpretation of the relevance of this fact to the field of statistics, but it is clear that the general increase in the speed of computations has allowed the development of useful computationally-intensive methods. These methods together constitute the field of computational statistics. Computational inference as an approximation is now generally as accepted as asymptotic inference (more readily by many people).

At a more technical level, standardization of hardware and software has yielded meaningful advances. In the 1970's over 75% of the computer market was dominated by the IBM 360/370 systems. Bill and I described the arithmetic implemented in this computer. It was in base 16 and did not do rounding. The double precision exponent had the same range as that of single precision. The IBM Fortran compilers (G and H) more-or-less conformed to the Fortran 66 standard (and they chose the one-trip for null DO-loops). Pointers and dynamic storage allocation were certainly not part of the standard. PL/I was a better language/compiler and IBM put almost as many 1970s dollars in pushing it as US DoD in 1990s dollars pushing Ada. And of course, there was JCL!

The first eight of the Top 10 algorithms were in place thirty years ago, and we described statistical applications of at least five of them. The two that were not in place in 1980 do not have much relevance to statistical applications. (OK, I know somebody will tell me soon how important these two algorithms are, and how they have just been used to solve some outstanding statistical problem.)

One of the Top 10 algorithms, dating to the 1940s, is the basis for MCMC methods, which began receiving attention by statisticians around 1990, and in the past twenty years has been one of the hottest areas in statistics.

I could go on, but I tire of telling "the way we used to do it". Let's learn what we need to do it the best way now.

Data

I do not make significant use of any "real world" datasets in the book. I often use "toy" datasets because I think that is the quickest way to get the essential characteristics of a method. I sometimes refer to the datasets that are available in R or S-Plus, and in some exercises, I point to websites for various real world datasets.

Many exercises require the student to generate artificial data. While such datasets may lack any apparent intrinsic interest, I believe that they are often the best for learning how a statistical method works. One of my firm beliefs is

If I understand something, I can simulate it.

Learning to simulate data with given characteristics means that one understands those characteristics. Applying statistical methods to simulated data may lack some of the perceived satisfaction of dealing with "real data", but it helps us better to understand those methods and the principles underlying them.

A Word About Notation

I try to be very consistent in notation. Most of the notation is "standard". Appendix C contains a list of notation, but a general summary here may be useful. Terms that represent mathematical objects, such as variables, functions, and parameters, are generally printed in an italic font. The exceptions are the standard names of functions, operators, and mathematical constants, such as sin, log, Γ (the gamma function), Φ (the normal CDF), E (the expectation operator), d (the differential operator), e (the base of the natural logarithm), and so on.

I tend to use Greek letters for parameters and English letters for almost everything else, but in some cases, I am not consistent in this distinction.

I do not distinguish vectors and scalars in the notation; thus, "x" may represent either a scalar or a vector, and x_i may represent either the i^{th} element of an array or the i^{th} vector in a set of vectors. I use uppercase letters for matrices and the corresponding lowercase letters with subscripts for elements of the matrices. I do not use boldface except for emphasis or for headings.

I generally use uppercase letters for random variables and the corresponding lowercase letters for realizations of the random variables. Sometimes I am not completely consistent in this usage, especially in the case of random samples and statistics.

I describe a number of methods or algorithms in this book. The descriptions are in a variety of formats. Occasionally they are just in the form of text; the algorithm is described in (clear?!) English text. Often they are presented in the form of pseudocode in the form of equations with simple for-loops, such as for the accumulation of a sum of corrected squares on page 116, or in pseudocode that looks more like Fortran or C. (Even if C-like statements are used, I almost always begin the indexing at the 1^{st} element; that is, at the *first* element, not the zeroth element. The exceptions are for cases in which the index also represents a power, such as in a polynomial; in such cases, the 0^{th} element is the first element. I call this "0 equals first" indexing.) Other times the algorithms are called "Algorithm x.x" and listed as a series of steps, as on page 218. There is a variation of the "Algorithm x.x" form. In one form the algorithm is given a name and its input is listed as input arguments, for example MergeSort, on page 122. This form is useful for recursive algorithms because it allows for an easy specification of the recursion. Pedagogic considerations (if not laziness!) led me to use a variety of formats for presentation of algorithms; the reader will likely encounter a variety of formats in literature in statistics and computing, and some previous exposure should help to make the format irrelevant.

Use in the Classroom

Most statistics students will only take one or two courses in the broad field of computational statistics. I have tried at least to introduce the major areas of the field, but this means, of course, that depth of coverage of most areas has been sacrificed.

The chapters and sections in the book vary considerably in their lengths, and this sometimes presents a problem for an instructor to allocate the coverage over the term. The number of pages is a better, but still not very accurate, measure of the time required to cover the material.

There are several different types of courses for which this book could be used, either as the primary text or as a supplement.

Statistical Computing Courses

Most programs in statistics in universities in the United States include a course called "statistical computing". There are two kinds of courses called "statistical computing". One kind is "packages and languages for data analysis". This book would not be of much use in such a course.

The other kind of course in statistical computing is "numerical methods in statistics". Part II of this book is designed for such a course in statistical computing. Selected chapters in Parts III and IV could also be used to illustrate and motivate the topics of those six chapters. Chapter 1 could be covered as necessary in a course in statistical computing, but that chapter should not be skipped over too lightly.

One of the best ways to learn and understand a computational method is to implement the method in a computer program. Executing the program provides immediate feedback on the correctness. Many of the exercises in Part II require the student to "write a program in Fortran or C". In some cases, the purpose is to identify design issues and how to handle special datasets, but in most cases the purpose is to ensure that the method is understood; hence, in most cases, instead of Fortran or C, a different language could be used, even a higher-level one such as R. Those exercises also help the student to develop a facility in programming. Programming is the best way to learn programming. (Read that again; yes, that's what I mean. It's like learning to type.)

Computational Statistics Courses

Another course often included in statistics programs is one on "computationally intensive statistical methods", that is, what I call "computational statistics". This type of course, which is often taught as a "special topics" course, varies widely. The intent generally is to give special attention to such statistical methods as the bootstrap or to such statistical applications as density estimation. These topics often come up in other courses in statistical theory and methods, but because of the emphasis in these courses, there is no systematic development of the computationally-intensive methods. Parts III and IV of this book are designed for courses in computational statistics. I have taught such a course for a number of years, and I find that the basic material of Chapter 1 bears repeating (although it is prerequisite for the course that I teach). Some smattering of Part II, especially random number generation in Chapter 7, may also be useful in such a course, depending on whether or not the students have a background in statistical computing (meaning "numerical methods in statistics").

Modern Applied Statistics Courses
The book, especially Parts III and IV, could also be used as a text in a course on "modern applied statistics". The emphasis would be on modeling and statistical learning; the approach would be exploration of data.

Exercises

The book contains a number of exercises that reinforce the concepts discussed in the text or, in some cases, extend those concepts. Appendix D provides solutions or comments for several of the exercises.

Some of the exercises are rather open-ended, asking the student to "explore". Some of the "explorations" are research questions.

One weakness of students (and lots of other people!) is the ability to write clearly. Writing is improved by practice and by criticism. Several of the exercises, especially the "exploration" ones, end with the statement: "Summarize your findings in a clearly-written report." Grading such exercises, including criticism of the writing, usually requires more time — so a good trick is to let students "grade" each others' work. Writing and editing are major activities in the work of statisticians (not just the academic ones!), and so what better time to learn and improve these activities than during the student years.

In most classes I teach in computational statistics, I give Exercise A.3 in Appendix A (page 656) as a term project. It is to replicate and extend a Monte Carlo study reported in some recent journal article. Each student picks an article to use. The statistical methods studied in the article must be ones that the student understands, but that is the only requirement as to the area of statistics addressed in the article. I have varied the way in which the project is carried out, but it usually involves more than one student working together. A simple way is for each student to referee another student's first version (due midway through the term) and to provide a report for the student author to use in a revision. Each student is both an author and a referee. In another variation, I have students be coauthors.

Prerequisites

It is not (reasonably) possible to itemize the background knowledge required for study of this book. I could claim that the book is self-contained, in the sense that it has brief introductions to almost all of the concepts, but that would not be fair. An intermediate-level background in mathematics and statistics is assumed.

The book also assumes some level of computer literacy, and the ability to "program" in some language such as R or Matlab. "Real" programming ability is highly desirable, and many of the exercises in Part II require real programming.

Acknowledgements

I thank John Kimmel of Springer for his encouragement and advice on this book and other books he has worked with me on. I also thank the many readers who have informed me of errors in other books and who have otherwise provided comments or suggestions for improving my exposition.

I thank my wife María, to whom this book is dedicated, for everything.

I used LaTeX 2_ε to write the book, and I used R to generate the graphics. I did all of the typing, programming, etc., myself, so all mistakes are mine. I would appreciate receiving notice of errors as well as suggestions for improvement.

Notes on this book, including errata, are available at
`http://mason.gmu.edu/~jgentle/cmstatbk/`

Fairfax County, Virginia

James E. Gentle
July 14, 2009