Computer Science and Applied Mathematics

PROBABILITY, STATISTICS, AND QUEUEING THEORY

WITH COMPUTER SCIENCE APPLICATIONS

Arnold O. Allen

PROBABILITY, STATISTICS, AND QUEUEING THEORY

With Computer Science Applications

ARNOLD O. ALLEN
IBM Systems Science Institute
Los Angeles, California

ACADEMIC PRESS
New York San Francisco London 1978
A Subsidiary of Harcourt Brace Jovanovich, Publishers

ACADEMIC PRESS, INC.
111 Fifth Avenue, New York, New York 10003

United Kingdom Edition published by
ACADEMIC PRESS, INC. (LONDON) LTD.
24/28 Oval Road, London NW1

Library of Congress Cataloging in Publication Data

Allen, Arnold O
Probability, statistics, and queueing theory.

(Computer science and applied mathematics series)
Includes bibliographical references.
1. Probabilities. 2. Queueing theory. 3. Mathematical statistics. 4. Mathematics–Data processing.
I. Title.
QA273.A46 519.2 77-80778
ISBN 0-12-051050-2

PRINTED IN THE UNITED STATES OF AMERICA

Computer Science and Applied Mathematics

A SERIES OF MONOGRAPHS AND TEXTBOOKS

Editor

Werner Rheinboldt

University of Maryland

HANS P. KÜNZI, H. G. TZSCHACH, and C. A. ZEHNDER. Numerical Methods of Mathematical Optimization: With ALGOL and FORTRAN Programs, Corrected and Augmented Edition

AZRIEL ROSENFELD. Picture Processing by Computer

JAMES ORTEGA AND WERNER RHEINBOLDT. Iterative Solution of Nonlinear Equations in Several Variables

AZARIA PAZ. Introduction to Probabilistic Automata

DAVID YOUNG. Iterative Solution of Large Linear Systems

ANN YASUHARA. Recursive Function Theory and Logic

JAMES M. ORTEGA. Numerical Analysis: A Second Course

G. W. STEWART. Introduction to Matrix Computations

CHIN-LIANG CHANG AND RICHARD CHAR-TUNG LEE. Symbolic Logic and Mechanical Theorem Proving

C. C. GOTLIEB AND A. BORODIN. Social Issues in Computing

ERWIN ENGELER. Introduction to the Theory of Computation

F. W. J. OLVER. Asymptotics and Special Functions

DIONYSIOS C. TSICHRITZIS AND PHILIP A. BERNSTEIN. Operating Systems

ROBERT R. KORFHAGE. Discrete Computational Structures

PHILIP J. DAVIS AND PHILIP RABINOWITZ. Methods of Numerical Integration

A. T. BERZTISS. Data Structures: Theory and Practice, Second Edition

N. CHRISTOPHIDES. Graph Theory: An Algorithmic Approach

ALBERT NIJENHUIS AND HERBERT S. WILF. Combinatorial Algorithms

AZRIEL ROSENFELD AND AVINASH C. KAK. Digital Picture Processing

SAKTI P. GHOSH. Data Base Organization for Data Management

DIONYSIOS C. TSICHRITZIS AND FREDERICK H. LOCHOVSKY. Data Base Management Systems

WILLIAM F. AMES. Numerical Methods for Partial Differential Equations, Second Edition

ARNOLD O. ALLEN. Probability, Statistics, and Queueing Theory: With Computer Science Applications

In preparation

ALBERT NIJENHUIS AND HERBERT S. WILF. Combinatorial Algorithms. Second edition.

AZRIEL ROSENFELD. Picture Languages, Formal Models for Picture Recognition

ISAAC FRIED. Numerical Solution of Differential Equations

This is a volume in

COMPUTER SCIENCE AND APPLIED MATHEMATICS

A Series of Monographs and Textbooks

Editor: WERNER RHEINBOLDT

Faith is belief without evidence in what is told by one who speaks without knowledge of things without parallel.
Ambrose Bierce

PREFACE

The genesis of this book is my experience in teaching the use of statistics and queueing theory for the design and analysis of data communication systems at the Los Angeles IBM Systems Science Institute. After 18 hours of instruction, spread over a three-week period, with emphasis on how to apply the theory to computer science problems, students with good mathematical training were able to use statistics and queueing theory effectively in a case study for a data communication system. Even students whose mathematical education consisted of high school algebra were able to appreciate how these disciplines were applied. The results were due to the fact that the applications of the theory were demonstrated by straightforward examples that could easily be understood and which illustrated important concepts.

This book contains a great deal more material than could be presented to students at the IBM Systems Science Institute. The book is designed as a junior–senior level textbook on applied probability and statistics with computer science applications. It may also be used as a self-study book for the practicing computer science (data processing) professional. The assumed mathematical level of the reader is the traditional one year of analytical geometry and calculus. However, readers with only a college algebra background should be able to follow much of the development and most of the practical examples; such readers should skip over most of the proofs.

I have attempted to state each theorem carefully and explicitly so the student will know exactly when the theorem applies. I have omitted many of the proofs. With a few exceptions, I have given the proof of a theorem only when the following conditions apply: (a) the proof is straightforward, (b) reading through the proof will improve the student's understanding, and (c) the proof is not long.

The emphasis in this book is on how the theorems and theory can be used to solve practical computer science problems. However, the book and a course based on the book should be useful for students who are interested not in computer science itself, but in using a computer to solve problems in other fields such as engineering, physics, operations research, and management science.

A great deal of computation is needed for many of the examples in this book because of the nature of the subject matter. In fact the use of a computer is almost mandatory for the study of some of the queueing theory models. I believe Kenneth Iverson's APL is the ideal choice of a programming language for making probability, statistics, or queueing theory calculations. Short APL programs are presented in Appendix B to aid the student in making the required calculations. In writing these APL programs I have attempted to write as directly as possible from the equations given in the text, avoiding "one liners"; I have not sought efficiency at the expense of clarity. Every APL program referred to in the text can be found in Appendix B.

The excellent series of books by Donald E. Knuth [1–3] has influenced the writing of this book. I have adopted Knuth's technique of presenting complex procedures in an algorithmic way, that is, as a step by step process. His practice of rewarding the first finder of any error with $1 for the first edition and $2 for the second will also be adopted. I have also followed his system of rating the exercises to encourage students to do at least the simpler ones. I believe the exercises are a valuable learning aid.

Following Knuth, each exercise is given a rating number from 00 to 40. The rating numbers can be interpreted as follows: 00—a very easy problem that can be answered at a glance if the text has been read and understood; 10—a simple exercise which can be done in a minute or so; 20—an exercise of moderate difficulty requiring 18 or 20 minutes of work to complete; 30—a problem of some difficulty requiring two or more hours of work; 40—a lengthy, difficult problem suitable for a term project. (All entries with numbers higher than 30 are "virtual.")

We precede the rating number by HM for "higher mathematics" if the problem is of some mathematical sophistication requiring an understanding of calculus such as the evaluation of proper or improper integrals or summing an infinite series. The prefix C is used if the problem requires extensive computation which would be laborious without computer aid such as an APL terminal or some such facility. T is used to indicate an exercise whose solution is basically tedious, even though the result may be important or exciting; that is, the required procedure is too complex to program for computer solution without more frustration than carrying it out manually.

The reader is assumed to have a basic knowledge of computer hardware and software. Thus he or she should be familiar with such concepts as the central processing unit (CPU), main storage, channels, registers, direct access storage devices such as disks and drums, and programming languages such as FORTRAN, COBOL, PL/I, and APL. The reader should have coded and tested several programs, and generally be familiar with the methods whereby computers are used to solve problems.

I have attempted to use the same principles in planning and writing this book that are used in designing and writing good computer programs, namely, the rules of structured programming as advocated by Edsger W. Dijkstra [4] and others. That is, in planning this book, I first decided what my overall objectives were. I then decided what selections from probability and statistics were necessary to achieve these goals. For each of these selections, in turn, I decided what subobjectives and what examples were needed to achieve the objectives assigned to that part of the book. I thus proceeded in a top-down fashion to a detailed outline of the book. Modifications to the original plan were found to be necessary, of course. However, before each modification was made, its impact upon the whole book was evaluated; necessary changes to the outline and to already written sections were made. I hope the final product has the hierarchical structure that was intended.

Cited References

1. D. E. Knuth, *The Art of Computer Programming, Vol. 1, Fundamental Algorithms*, 2nd ed. Addison Wesley, Reading, Massachusetts, 1973.
2. D. E. Knuth, *The Art of Computer Programming, Vol. 2, Seminumerical Algorithms*. Addison Wesley, Reading, Massachusetts, 1969.
3. D. E. Knuth, *The Art of Computer Programming, Vol. 3, Sorting and Searching*. Addison Wesley, Reading, Massachusetts, 1973.
4. E. W. Dijkstra, Notes on Structured Programming, in *Structured Programming* by O. J. Dahl, E. W. Dijkstra, and C. A. R. Hoare. Academic Press, New York, 1972.

ACKNOWLEDGMENTS

Much of the material in this book has been class tested at the Los Angeles IBM Systems Science Institute, first in a ten-day class "Performance Analysis of Communication Systems" and then in its successor "Performance Evaluation and Capacity Planning." I am grateful for the constructive suggestions of the students; some of their humorous comments have been included in the book as "Student Sayings."

It is my pleasure to acknowledge the help of several individuals. Professor David Cantor of UCLA and Gerald K. McAuliffe of IBM provided helpful advice. My colleague John Cunneen at the IBM Systems Science Institute provided encouragement and constructive suggestions. Mr. Lucian Bifano of IBM Los Angeles provided access to an IBM 5100 computer when I needed it most. Mr. John Hesse, also of IBM Los Angeles, provided a great deal of help in using the IBM 5100 and some useful programs for editing and listing the APL programs in Appendix C. Ms. Vi Ma of the IBM Research Library in San Jose, California, was most helpful in obtaining copies of important references. My manager Mr. J. Perry Free was most supportive in allowing me vacation time to finish the book. My wife Betty and my son John were valuable proof readers. Finally, I want to thank Elaine Barth for the outstanding way she converted my inscrutable scrawling into finished typing.

I hope the completed work shows the joy I felt in writing it. (Only the wisdom of the publisher kept me from calling the book "The Joy of Statistics.") I would appreciate hearing from readers. My address is: Dr. Arnold O. Allen, IBM Systems Science Institute, 3550 Wilshire Boulevard, Los Angeles, California 90010.

CONTENTS

Chapter Six Queueing Theory Models of Computer Systems

PART THREE STATISTICAL INFERENCE

Chapter Seven Estimation

Chapter Eight Hypothesis Testing

The theory of probabilities is at bottom nothing but common sense reduced to calculus. Pierre Simon de Laplace

Chapter One

INTRODUCTION

This chapter is a preview of what the book is all about. As the title suggests, it is concerned with the application of probability, statistics, and queueing theory to computer science problems. It was written for the computer science (data processing) specialist or for one preparing for a career in this field. However, it should be of interest to many who are not computer science oriented, but who merely use a computer occasionally in their daily activities.

The book is divided into three parts: Probability, Queueing Theory, and Statistical Inference.

There are three chapters in Part I. In the first of these, Chapter 2, we take up the basic concept of probability and how to deal with it. In most areas of computer science we deal not with deterministic phenomena, but rather with probabilistic phenomena. The time it takes to write and check out a computer program, the time it takes to run a program (measured from the time it is submitted to a batch system or invoked via an on-line system), the time it takes to retrieve information from a storage device, as well as the number of jobs awaiting execution on a computer system, are all examples of probabilistic or random variables. This means that we cannot predict, in advance, what these values are. However, by using the concepts of probability, probability distributions, and random variables, we can make probability estimates (give the fraction of the time) that the values will fall into certain ranges, exceed certain limits, etc. These subjects are all covered in Chapter 2. We also discuss parameters of random variables (such as the response time, X, at an on-line terminal); these parameters include the mean or average value, the standard deviation (a measure of the deviation of values from the mean), as well as higher moments. We show how to use these parameters to make probability calculations. In the final part of Chapter 2 we discuss some powerful tools to use in dealing with random variables; these include conditional expectation, transform methods, and inequalities. Transform methods are important in studying random variables. We define and illustrate the use of the moment generating function, the generating function or z-transform, and the Laplace–Stieltjes transform. However, we do not make extensive use of transforms in this book, for to do so would take us too far from our primary goals. We do not want to "raise the Laplacian curtain" in the words of the eminent British statistician David Kendall.

In Chapter 3 we study the probability distributions most commonly used in applied probability, particularly for computer science applications. We give examples of the use of all of these except those used only in statistical inference, the subject of Part III of the book. A summary of the properties of the random variables discussed in Chapter 3 is given in Tables 1 and 2 of Appendix A.

In Chapter 4 the important concept of a stochastic process is defined, discussed, and illustrated with a number of examples. The chapter was written primarily as a support chapter for Part II, queueing theory. We examine the Poisson process because it is extremely important for queueing theory; similarly for the birth-and-death process. We finish the chapter by discussing Markov processes and chains—subjects that are important not only for queueing theory but also for other widely used models in computer science.

Part II of the book is the area that is likely to be most foreign to the reader, the discipline of queueing theory. Queueing theory is an applied branch of probability theory, itself, but some expressions and symbols are used differently in queueing theory than they are in other areas of probability and in statistics.

Figure 1.1, which also appears as Fig. 5.1.1 in Chapter 5, shows the elements of a queueing system. There is a "customer" population; a customer may be an inquiry to be processed by an on-line inquiry system, a job to be processed by a batch computer system, a message to be transmitted over a communication line, a request for service by a computer input/output channel, etc. Customers arrive in accordance with an "arrival process" of

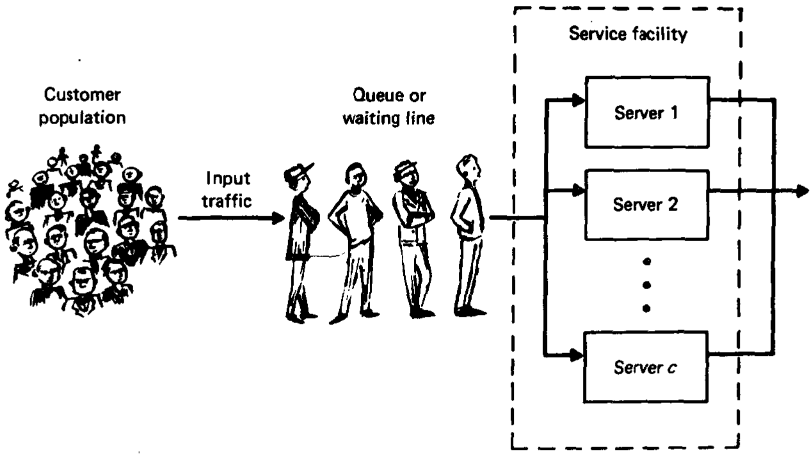

Fig. 1.1 Elements of a queueing system.

some type (a "Poisson arrival process" is one of the most common). Customers are provided service by a service facility which has one or more servers, each capable of providing service to a customer. Thus a server could be a program which processes an inquiry, a batch computer system, a communication line, a computer channel, a central processing unit, etc. If all the servers in the service facility are busy when a customer arrives at the queueing system, that customer must queue for service; that is, the customer must join a queue (waiting line) until a server is available. In Chapter 5 we study the standard (one might say "canonical") queueing systems and see how they can be applied in studying computer science problems. We have gathered most of the queueing theory formulas from Chapters 5 and 6 and put them in Appendix C. You will find this appendix to be very useful as a reference section once you have mastered Chapters 5 and 6. The APL programs to calculate the values given by the formulas in Appendix C have been collected in Appendix B. In Chapter 6 we discuss more sophisticated queueing system models that have been developed to study computer systems, particularly on-line computer systems where customers are typically represented by remote computer terminals. As in Chapter 5, a number of examples of "real world" use of the models are presented.

The subject matter of Part III, statistical inference, is rather standard statistical fare but we have attempted to give it a computer science orientation. Statistical inference could perhaps be defined as "the science of drawing conclusions about a population on the basis of a random sample from that population." For example, we may want to estimate the mean arrival rate of inquiries to an on-line inquiry system. We may also want to deduce what type of arrival process is involved. We can approach these tasks on the basis of a sample of the arrival times of inquiries during n randomly selected time periods. The first task is one of estimation. We want to estimate the mean arrival rate on the basis of the observed arrival rate during n time intervals. This is the subject of Chapter 7. In this chapter we learn not only how to make estimates but also how to make probability judgements concerning the accuracy of the estimates.

One of the important topics of Chapter 8 (on hypothesis testing) is goodness-of-fit tests, the kind of test needed in the second task of the previous paragraph. In that particular case we might want to test the hypothesis that the arrival process has a Poisson pattern (recall we said earlier that this is one of the most popular kinds of arrival patterns because of its desirable mathematical properties). We discuss the two most widely used goodness-of-fit tests, the chi-square and the Kolmogorov–Smirnov. We give several examples of the application of these tests. Chapter 8 also contains a number of other hypothesis tests of great utility. These tests are summarized in Table 8.5.1 at the end of the chapter. You will probably find this to be a useful reference, too, when you have mastered Chapter 8.

This completes the summary of the book. We hope you will find the study of this book entertaining as well as educational. We have tried to avoid being too solemn. For example, we chose whimsical names for mythical companies in our examples. We have attempted to make the examples as practical as possible within the constraints of a reasonably short description. We welcome your comments, suggestions, and observations.